WILEY FINANCE

MODELING STRUCTURED FINANCE Cash Flows with Microsoft® Excel® +CD

A Step-by-Step Guide

CD-ROM includes modeling exercises and a final version of the model discussed in the text

Keith A. Allman

Modeling Structured Finance Cash Flows with Microsoft® Excel®

A Step-by-Step Guide

KEITH A. ALLMAN

John Wiley & Sons, Inc.

Modeling Structured Finance Cash Flows with Microsoft® Excel®

Founded in 1807, John Wiley & Sons is the oldest independent publishing company in the United States. With offices in North America, Europe, Australia and Asia, Wiley is globally committed to developing and marketing print and electronic products and services for our customers' professional and personal knowledge and understanding.

The Wiley Finance series contains books written specifically for finance and investment professionals as well as sophisticated individual investors and their financial advisors. Book topics range from portfolio management to e-commerce, risk management, financial engineering, valuation and financial instrument analysis, as well as much more.

For a list of available titles, please visit our Web site at www.WileyFinance.com.

Modeling Structured Finance Cash Flows with Microsoft® Excel®

A Step-by-Step Guide

KEITH A. ALLMAN

John Wiley & Sons, Inc.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Wiley Bicentennial Logo: Richard J. Pacifico.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books. For more information about Wiley products, visit our Web site at www.wiley.com.

Library of Congress Cataloging-in-Publication Data:

Allman, Keith A., 1977-
Modeling structured finance cash flows with Microsoft Excel : a step-by-step guide / Keith A. Allman.
p. cm.—(Wiley finance series)
Includes bibliographical references and index.
ISBN 978-0-470-04290-8 (paper/cd-rom)
1. Cash management—Mathematical models. 2. Cash flow—Mathematical models. 3. Microsoft Excel (Computer file) 4. Corporations—Finance—Mathematical models. I. Title.
HG4028.C45A454 2007
658.15′50285554--dc22
2006025757

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

Contents

CHAPTER 2
Asset Cash Flow Generation 23

CHAPTER 3
Prepayments 43

CHAPTER 4
Delinquency, Default, and Loss Analysis 59

Preface

During my first analytics position after graduate school, I asked a vice president at our company what the best way was to learn how his group modeled transactions. He answered with a grin: “Trial by fire.” From that point on, I could not have counted the gray hairs that I developed trying to figure out the most precise and efficient method of modeling a transaction. I am pleased to say those days are behind me and it no longer takes me hours to construct a powerful, accurate model. Nevertheless I am dismayed when I speak with finance peers who convey their desire to learn better financial modeling and are intimidated by the task or simply at a loss for where to begin. At those moments, I often think how I came to acquire the knowledge and skills necessary to model a diverse array of financial transactions.

I recalled hours spent poring over “how-to” books about Excel that were filled with hundreds of functions and formulas and left me feeling like I didn’t have any idea where to start modeling a transaction. The how-to books provide excellent basics of application operation yet they do not offer any context for applying those skills. My next thought was graduate school, where many courses such as Statistics, Economics, Corporate Finance, Capital Markets, and Decision Making utilize Excel for assignments and examinations. Unfortunately, for everyday application, the graduate school classes provide context, but typically on very specialized subjects that still left me with no framework to build a financial model. The next step I took was to purchase more advanced books with the words “Financial Modeling” in the title. With these, I found the topics highly theoretical or applicable to extremely focused fields that do not translate into a practical model oriented towards cash flow analysis.

I realized that most of my knowledge, expertise, and fluidity in financial modeling came from working in analytics groups. There I focused on interpreting structures from documents and benefited by learning from others about how to convert the deal structure into a working model. Between the insurance and banking industries, I’ve seen and built numerous models—from the very basic that are little more than a balance sheet with formulas to incredibly complex models involving stochastic simulations. With every model on which I have worked, I have tried to take away what I have felt to be the best attributes and incorporate those features into my current modeling.

As my experience with financial models continues to grow, I definitely feel that I am at a point where I have worked with enough models to distinguish trends, common practices, and characteristics of exceptional financial modeling. My personal experience has been with cash-flow-based models seen in most fixed income, structured, asset-based, or project finance transactions. To avoid trial by fire, this book teaches the framework and specifics of cash-flow-based modeling using structured finance as a context. If examples are followed from beginning to end, the result will be a fully operating cash flow model that the reader built step by step.

Aside from being able to create a model from the ground up, understanding how each component is built and interacts will aid a reader who needs to work with other peoples' models. I often find working with another person's model more difficult than building a new one from scratch. It takes time to discern the core components and functionality of the model. However, most well-thought-out models have similar basic elements that can be understood and manipulated. This book intends to cover each of those elements and provide the reader with enough depth to proficiently work with existing models.

Looking back at the moment when I had that trial-by-fire response, I certainly do not feel that has to be the standard that anyone should have to rely on. Regardless if the reader is a new finance professional who wants to learn how to build a model, a seasoned professional who works with others' models, a structured finance professional looking for analyses specific to the field, or simply anyone interested in understanding financial modeling better, I feel that passing on my experience in the form of a book with practical examples can help make the learning process easier and more efficient.

Keith A. Allman

New York, New York
December 2006

Acknowledgments

My career in finance began at MBIA, Inc., a leading financial guarantor and provider of specialized financial services. There three individuals provided an excellent introduction to financial modeling, namely Henry Wilson, William Devane, and Melissa Brice-Johnson. In particular, I would like to thank Henry for giving me the opportunity to work on a variety of transactions and William for showing me many fundamental techniques. After leaving MBIA, I wrote the first three chapters of this book as part of a proposal to John Wiley & Sons, where I would like to thank my peer editors Maria Costa for her in-depth review as well as Lionel Beehner for his editorial suggestions. Further editorial suggestions were made by Omar Haneef and Matthew Niedermaier as the book developed, both whom I would like to thank especially for their work on the text and Model Builder exercises. Also, this book could not possibly have been brought to market without the amazing support of William Preinitz, who read through, approved, and was a driving force in receiving Citigroup's compliance approval. Lastly, I am very grateful for Siobhan Devine, whose patience and encouragement kept me centered throughout everything.

Also at Wiley, I would like to thank Bill Falloon for working with me from taking the proposal to a signed contract, Emilie Herman for her consistent involvement in every aspect, Laura Walsh and her team for the cover and marketing work, and Mary Daniello and her team for copyediting such a detail-oriented book.

K. A. A.

About the Author

Keith Allman is currently a vice president in the Global Special Situations Group at Citigroup, where he focuses on emerging market analysis. He has created, audited, and used hundreds of cash flow models for mortgages, autos, equipment leases, credit cards, project finance, and multiple esoterics. Prior to his current role, he worked in the Structured Finance group at Citigroup modeling transactions for their conduits. Mr. Allman began his career in finance at MBIA, Inc., a leading financial guarantor, where he was a senior analyst in its quantitative analytics group. Outside of corporate work, Mr. Allman has written computer curriculum and provides instruction for low-income individuals through Streetwise Partners. His education includes a master's degree in international affairs with a concentration in finance and banking from Columbia University and bachelor degrees in political science and psychology from UCLA.

Modeling Structured Finance Cash Flows with Microsoft® Excel®

Introduction

The basic idea behind any financial model is to bring order and understanding to the numerous variables and complex information that financial transactions present. Learning to build one from a blank spreadsheet is often a daunting task to newcomers because of the sheer amount of information and nearly infinite methods of manipulating data. This book seeks to bring a systematic, well-explained method to constructing a particularly popular and adaptable type of model—the cash flow model. Through the use of thorough explanation, graphical examples, and the simultaneous application of learned methods featured in the *Model Builder* exercises, anyone with a background in finance and basic spreadsheet understanding can develop and understand a fully functioning financial model.

The most significant aspect of the model that will be created is that it is constructed within a real-world context focusing on the structured finance industry. Many other financial modeling books explain either application functions or specific theoretical concepts. These books are good for learning a program or understanding an academic topic, yet they are difficult to translate into a functioning financial model. By combining specific application instruction with theory, this book teaches skills that can be applied instantly to professional level modeling.

While the book focuses on structured finance analysis, the model created here can be adapted for use in other fields. A fundamental question is whether a cash flow model is the appropriate choice for the transaction under consideration. With cheap memory, powerful processors, and constant evolutions in financial analytics, a multitude of models are available ranging from real-time market value models to code-intensive Monte Carlo simulations. The cash flow model is primarily used for transactions that involve assets generating cash flow, which is applied against a set of liabilities. These transaction types are often encountered in structured, asset-based, and project finance and typically include the following asset classes:

- Automobile loans and leases
- Residential mortgages
- Commercial mortgages
- Equipment leases
- Credit card receivables
- Insurance/annuity arbitrage
- Emerging market remittances
- CBO/CLO/CDO
- Small business loans
- Timeshares

- Infrastructure (toll road, airport, etc.)
- Resources (oil, timber, etc.)

Naturally this list is not exhaustive. It covers a majority of asset classes that use a cash flow based model. It is possible to merge types of models such as using a Monte Carlo model to determine defaults and then running the results through a cash flow model. The key to deciding on whether a cash flow model is necessary depends on the desired result.

A cash flow model takes in asset assumptions, runs the generated cash through a series of liability assumptions, and determines where and how much cash was allocated over time. This type of modeling is used from many different perspectives, with many different results in mind.

One of the most common uses is an issuer that needs to fund the generation of assets. A company such as Toyota, which has a finance division, may want to fund leases for their own vehicles. Toyota needs to raise money to provide the leases. It could do so in the capital markets by asking a bank to loan funds against the leases by either having a bank directly issue money or sell debt in the term market. Toyota's cash flow analysis would have to focus on how much cash its leases would generate over time to determine the amount of debt that can be issued. A Toyota analyst would want to build a cash flow model to project the expected cash being generated by the leases over time and how that cash would be allocated in a structured financing. The purpose of his or her analysis would be to understand the cash flow well enough to make sure they are receiving as much money as possible for the lowest cost.

The bank would do a similar analysis in more detail. It would want to know how typical Toyota leases perform over time in terms of delinquency, default, and prepayment. No bank would want to issue a billion dollars only to find that the assets will pay back anything less. Also, transactions typically have to be structured to a certain credit rating level set by the three primary credit rating agencies (Standard & Poor's, Moody's, and Fitch). To do this, a transaction has to withstand a certain amount of stress as dictated by the rating agency. The only way to do this is to build a dynamic model and stress it according to rating agency standards. In the bank's cash flow model, it would want to see how much cash the assets generate under stressful situations and whether that is enough to cover the financing costs imposed either by the market or by the bank itself.

In addition, a surety provider might provide insurance on the issuance. It would be extremely analytical in its decision because, if an interest or principal payment is missed on the financing, it would have to pay an insurance claim. A surety would use a cash flow model to ensure that, when variables are stressed, the interest and principal of the debt they insured is paid.

Finally, there are many other related parties that need to know what the issuers and the banks are doing. Credit rating agencies need to model the transactions to make sure that they support certain ratings that the bank and issuer desire. An auditor may want to make sure all the data in a prospectus is correct by modeling the deal on his or her own. A law firm may want to know if a certain legal structure works in practice. All of these parties need to build a cash flow model to complete such analyses.

THE THREE BASIC ELEMENTS OF A CASH FLOW MODEL

The cash flow model presented in this book can be parsed into three basic elements: inputs, cash flow structure, and outputs. A useful way to think about the three basic elements of financial modeling is to compare them to the elements of cooking. When preparing food, a chef has three basic elements: ingredients, method of preparation, and finished result. The ingredients all have different characteristics such as taste, smell, and texture. The chef then takes certain quantities of ingredients, mixes them in a particular way, and cooks them at a certain temperature for a set amount of time. The appearance, scent, and flavor of the finished food are entirely dependent on the ingredients and cooking process. Any alteration results in different qualities.

Likewise, in financial modeling, there are a number of inputs to start with, a cash flow structure that manipulates the inputs, and a final set of outputs that is reflective of both the selected inputs and structure. The simple pattern that should be realized from this comparison is that the first two elements are interconnected and integral in producing the defining characteristics of the third element.

Inputs

The general idea of an *input* is that it is any piece of data related to the transaction being modeled, factual or assumed, that is necessary to produce accurate results. Inputs can range from simple interest rate assumptions to more difficult concepts such as loss timing and severity. This book takes a model builder through the following inputs:

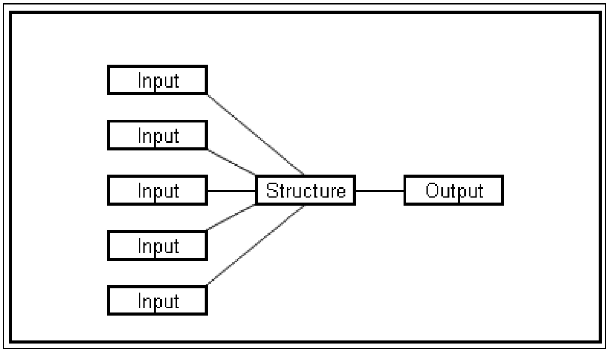


FIGURE I.1 Multiple inputs are passed through a structure to generate results.

1. Basic global inputs such as dates and timing.
2. Common asset inputs that cause and affect cash flow generation.
3. Common liability inputs that include interest rate vectors, fees, and basic liability structures.

Some of these inputs, such as prepayment and loss curves require additional explanation. Two chapters in the book explain the information and methodologies required to properly extrapolate and predict prepayment and loss curves. If questions arise regarding these two very important components, it may be worthwhile to jump to Chapters 3 and 4.

Cash Flow Structure

While inputs are the most familiar part to a model operator who is constantly changing them for different scenarios, the true heart of a cash flow model resides in the actual *cash flow structure*. This structure is created by using formulas, functions, and function/formula combinations to manipulate the inputs in a way that mirrors the transaction's agreed upon structure. The exact structure is dictated by the deal documents, but the cash flow section has a conventional flow.

Usually contained on one sheet, the cash flow structure "moves" cash in one direction for each period. Moving cash refers to the idea that as cash comes into a transaction from the asset inputs it is typically used in a very specific order that is determined by a term sheet, trust agreement, or indenture. The cash moves from one use to the next until the end of the priority of payments that designates where leftover cash, if any remains, should be distributed. Once this process is complete for an individual period, the process begins again for the next period until all cash is exhausted or the final term of the transaction is realized.

Some basic examples of uses of cash include:

- Transaction fees
- Taxes
- Senior interest and principal
- Subordinated interest and principal
- Equity payments due
- Reimbursements

Also, an interesting nuance in many transactions is that frequently the cash flow structure changes with differing assumptions. Using a basic mortgage-backed securities transaction as an example, often times the priority of payments will change if defaults increase to a certain level, breaching a preestablished limit—known as a *trigger*—set in the deal documents. The typical order may be to pay senior interest and principal then subordinated interest and principal. But if defaults breach the trigger, then all remaining cash may be directed to senior payments and the subordinated payments cutoff. This will be explained in much more detail in later cash flow structure chapters. The idea to understand is that the model being created often has a dynamic cash flow structure.

Outputs

The final element, *outputs*, is equally important to the first two because it is what most likely is seen by people unfamiliar to the model. Most of us do not have the time to pick through the minutiae of a model, but we want to read about the assumptions and results quickly and detailed enough to make a decision. Even printing out the cash flows period by period is ineffective because a majority of managers want to look at a single piece of paper in a format that they are used to and garner enough information from the single sheet to make a decision. It would be highly inefficient to develop a sophisticated financial model if it is overlooked because the results it presents are not clear and easy to read.

THE PROCESS OF BUILDING A CASH FLOW MODEL

Although the primary purpose of this book is to guide a reader through the mechanics of constructing a cash flow model, there are some steps that should be taken before and after the model is created in Excel. In particular for readers new to financial modeling, it is important to go through each of these steps to save time. As one becomes more fluent in financial modeling, the steps can be combined, such as building both the basic and advanced framework at the same time rather than in two separate steps. However, it should be noted that a flaw, which even seasoned financial modelers make, is skipping the plan-and-design and testing steps. A major design problem encountered halfway through building a new model may have been prevented by investing even a minimal amount of time planning. Even worse is not realizing that the model has a problem before using it for final results.

Plan and Design

The first step, planning and design, is what good financial modelers and computer programmers spend most of their time doing. This is best accomplished by writing or drawing out the necessary inputs, the expected flow of cash, and the type of results that are necessary. Each sheet should be thought of with memory, space, cell, and function limitations in mind. For instance, it would be extremely frustrating to build a model only to realize that the inputs require more than 65,556 rows (Excel's row constraint). Since this book uses a preplanned model as the basis for discussion, not much time is spent on this topic; but the importance of planning a new model should not be understated.

Obtain All Necessary Information

The more information available to determine inputs and base assumptions, the more accurate the model. Inputs are typically determined from historical data and studies or current market information. Many databases track common inputs such as interest rates. Asset-specific data such as historical losses, prepayments, and recovery rates are normally provided by the asset owners. If there is a lack of adequate historical data, proxy information or simulations can be used to estimate the input. For the cash flow structure, every financial transaction will be unique in terms of the documentation available, but important documents that detail how cash moves in the deal may include credit memos, term sheets, indentures, and trust documents.

Construct Basic Framework

The next step, which is the focus of this book, is to create the basic framework for the model by entering input data fields and cash flow formulas. The inputs and formulas should be entered in a logical order so that they build off of each other and leave off where the next one begins. The most fundamental concepts are covered in the basic framework such as dates and timing, asset amortization, transaction fees, liability principal and interest, reserve accounts, action buttons, and tests. As each concept is covered, the inputs and cash flow sheets will grow to where all of the basic information is entered and more advanced concepts outlined by the exact deal structure can be created. The basic framework is what should be saved as a template for future models.

Develop Advanced Structure

Once a basic structure is in place, the next step is to add advanced concepts that are unique to the transaction. Such concepts are typically ones that affect the cash flow structure and change how cash flows through the model. These concepts are mostly defined as triggers that are either in the process of being negotiated as the model is being built or are explicitly outlined in the deal documentation. The reason advanced structures are left to be added later is because they are typically unique to a transaction and if their remnants are carried over to future models there could be errors in the cash flow.

Validate Assumptions

There goes a saying in the financial modeling world, “A model is only as good as the assumptions that are put into it.” There is general consensus for this statement because an incredibly sophisticated model can be built—but if the assumptions are wrong, the results will be wrong. Once the framework and advanced structure of the model is complete, each assumption should be verified for accuracy and justified. Historical studies that produce prepayment, default, and interest rates should be reviewed for accuracy of data and methodology.

Test Model

The final step is to test the model. This goes beyond the tests that were set up in the basic framework. Those tests are intended to assess the rationality and logic of individual calculations. This last step is to test the rationality and logic of the model as a whole. Questions should be posed such as, ‘‘If the asset rate is increased how does that affect the liability amortization?’’, ‘‘If defaults increase how does that affect the liabilities?’’, and so on. The best way to answer these questions is to run extreme examples. Setting interest rates to 30 percent or 0 percent should produce vastly different effects in the model. In addition, if there is data available to benchmark the model against, those scenarios should be run.[1] Finally, if there is a back of the envelope calculation method possible, the model should be tested to see if the results are within a reasonable range.

HOW THIS BOOK IS DESIGNED

Since the primary purpose of this book is model construction, the first two steps mentioned in this introduction—plan and design and obtaining all necessary information—are assumed to be completed. The task at hand is to construct a basic framework for the model with a blank Excel workbook. The basic framework will consist of the most fundamental pieces of data and structure that are required for many further more detailed concepts to be developed.

The process of creating this framework is concept based rather than by each element in order. This means that instead of starting with inputs, then moving on to cash flow structure, and finishing up with outputs, the progression of the book covers concepts such as assets and work through the various inputs and structural building techniques. This methodology is preferred because the model builder will understand that each input interacts with the cash flow structure to a better degree rather than typing in a page of inputs and then having to go back and figure out what each one does. Also, to ease understanding, the concepts are introduced starting with the most fundamental and gradually becoming more complex.

Copying formulas into cells produces a financial model; but it is unlikely that such a process allows for understanding. Explaining everything from the overall process to the specific formulas is necessary. That is why each section in this text will begin with an explanation of the purpose of the section, followed by a Model Builder exercise where the theory is turned into practice.

The Model Builder exercises should be completed in Excel after reading the section’s text. If something is unclear in the Model Builder exercises, the CD-ROM that comes with this book contains each Model Builder exercise saved as a separate Excel file in individual chapter folders (e.g., *MB1-1.xls* in the Ch01 folder). The final

[1] Looking at prior public transactions and obtaining reports from the credit rating agencies are excellent sources of data to benchmark models.

working Project Model Builder—*Project_Model_Builder_Complete_Model.xls*—is also on the CD-ROM so that the reader can see how each section comes together at the end. There are also data files used in certain Model Builder exercises as well as supplementary files for reference. These are located in the Additional Files subfolders provided for some chapters. The About the CD-ROM section at the end of this book provides information on system requirements and the like.

Additionally, a typical problem in teaching any application is that students are often at different levels and become quickly uninterested when material is too basic or very intimidated when the material is too advanced. This book is focused at the intermediate Excel user. It can, however, be used by a beginner with the help of the Toolboxes found at the end of each section. The Toolboxes explain the Microsoft Excel features and functionality used in this book and can refresh your knowledge of this computer application or skipped over depending on your skill level.

One final note regards the upcoming release of Microsoft Excel 2007. The primary value of this book lies in the integration of finance theory with Excel's formula's and functions. Since the formulas and functions have not changed in Excel 2007, the model created from this book will go unchanged and will work perfectly. However, there are a few technical and instructional details that an Excel 2007 user may notice while going through the book. These details have been addressed in an appendix at the end of this book.

CHAPTER 1

Dates and Timing

The importance of dates and timing could not resonate more strongly than in the field of finance. From simple present value equations to more complex concepts such as yields and duration, time is an essential variable. Not surprisingly, dates and timing are also extremely important to cash flow modeling. Both dictate the core format of the model and permeate throughout many formulas and analyses.

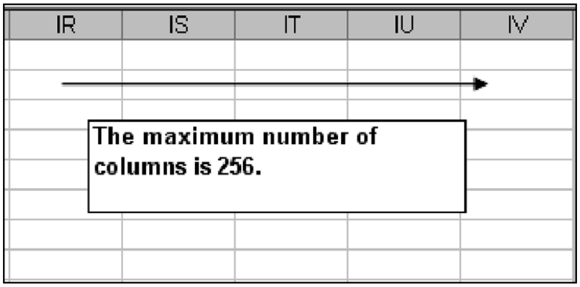
For Excel-based modeling, dates and timing can be separated into their own individual categories. Dates are often in calendar format and widely used to initiate or terminate preplanned events such as rate step-ups, final maturity, and the like. Timing is typically represented in numerical format or a vector of numbers to control payment frequency and most analytics involving periodicity.

TIME PROGRESSION

Before jumping right to specific methods and examples, it is worth a few minutes to think about the structure of the transaction and whether to have time "progress" horizontally or vertically. The key to this decision is understanding Excel's constraints within the context of the necessary payment frequency and overall length of the analysis. The designers of Excel limited the number of rows to 65,556 and columns to 256. If a transaction required modeling quarterly over a period of 25 years, it would not matter whether the payments move horizontally or vertically because only 100 (4 payments per year * 25 years = 100) columns or rows would be needed. However, if for some reason the frequency had to be switched to monthly, the payments would have to move vertically because 300 columns (12 payments per year * 25 years = 300) would exceed Excel's column constraint (see Figure 1.1).[1]

While it is generally recommended to have time progressing in the same direction through all sections of the model, a sheet will be required to have time progressing in

[1] Some modelers attempt to overcome the constraints by using additional sheets to continue the same section. While a viable option, it unnecessarily complicates the model flow and allows more opportunity for error.



The maximum number of columns is 256.

FIGURE 1.1 Notice that there are only 256 columns on a sheet, ending with column IV.

an alternate direction in some instances. There are a number of methods to transpose data in Excel, but many can be problematic when it comes time to refresh data or time consuming to implement. The best method is to use Excel's OFFSET function, which is described in Chapter 2's Toolbox.

DATES AND TIMING ON THE INPUTS SHEET

Regardless of the type of transaction, the dates and timing section of the Inputs sheet at bare minimum consist of two dates and two timing inputs. The first of the two minimum dates is the *closing date*. This date is the day that the deal closes and funds have been issued. Many timing factors initiate off this date such as fees and interest charges, asset lives, and so on. The second date that is close in time to the closing date is the *first payment date*. This is the date that the first payments of fees, interest, and principal are due. It is important to distinguish between these two dates as many transactions have first payment dates that are irregularly spaced between closing. This means that the initial period could have a partial month's payment requirement, which can only be calculated if both the closing date and the first payment date are inputted.

The two absolute necessities for timing are actually one concept, *payment frequency*. Payment frequency is an integer or fraction representing the time difference between payment dates. There are many possibilities for representing how often a transaction pays, but conventionally time is "represented" in years, which can be parsed into 2 for semiannual pay, 4 for quarterly pay, and 12 for monthly pay. However, it is extremely important to understand that time can sometimes be "measured" more accurately on a daily basis for the purpose of calculating payment amounts. This difference occurs because in finance the same months do not always have the same number of days in them depending on the day-count system used.

DAY-COUNT SYSTEMS: 30/360 VERSUS ACTUAL/360 VERSUS ACTUAL/365

Day-count systems evolved due to the nuances of the Gregorian calendar (i.e., different numbers of days in certain months and leap years). The *30/360-day-count system* simplifies the annual calendar by assuming there are 12 months with 30 days each, for a total of 360 days. Payment frequency is then every 30 days per month, 90 days per quarter, 180 days per semiannual, and so on. The 30/360 system is the most common in the United States because it is used for most municipal, corporate, and agency bonds, mortgage-backed securities, and many other types of notes and certificates.

Working with the 30/360-day-count system is made very easy in Excel, thanks to the DAYS360 function. This function requires three inputs: a start date, an end date, and a method.

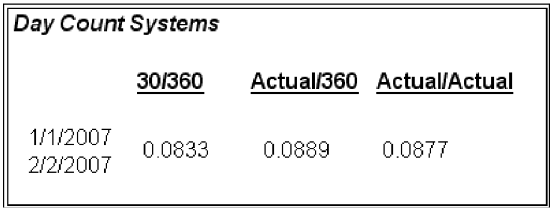
= DAYS360(start date, end date, method)

The start date is the first date to begin counting the days in between, the end date is the date to count the days up to, and finally the method is FALSE if the day count system desired is the U.S./National Association of Securities Dealers (NASD) system or TRUE if the system is European. The subtle difference between the two systems is that the U.S. system's starting date becomes equal to the 30th of the start month if the starting date is actually the 31st of that month. For the ending date, if it lands on the 31st of a month and the starting date is before the 30th of a month, the ending date become the 1st of the next month. However, if the start date is the 31st of a month, the ending date becomes equal to the 30th of the same ending date month. The European system is much simpler. If the starting or ending date lands on the 31st, the date just becomes the 30th of that month. The more common of the two is the U.S. system, which is the default method if a TRUE or FALSE is omitted.

Another popular day-count system is *actual/360*, which is primarily used for money market securities and U.S. Treasury bills. This system calculates an interest period as the actual number of days between two dates. It is important to note that to represent the day difference as a fraction of a year, the denominator uses 360 days, rather than 365. Since the natural format for dates in Excel are serial numbers, beginning with 1 for January 1, 1900, the actual difference between two dates can be calculated by subtracting the beginning date from the end date.

The final day-count system used is *actual/365* or *actual/actual*. Created to be as realistic to a year as possible, the calculation is virtually identical to actual/360, but instead of a year having 360 days it has 365.

An example of the differing date-count systems and their respective fractions of a year can be seen in Figure 1.2. While a few hundredths of a percent does not seem like much of a difference, this can be tens of thousands of dollars on deals in the hundred millions or billions of dollars.



Day Count Systems

	30/360	Actual/360	Actual/Actual
1/1/2007 2/2/2007	0.0833	0.0889	0.0877

FIGURE 1.2 A different fraction of time exists for the period January 1, 2007 to February 1, 2007 depending on the day-count system used.

MODEL BUILDER 1.1: INPUTS SHEET—DATES AND TIMING

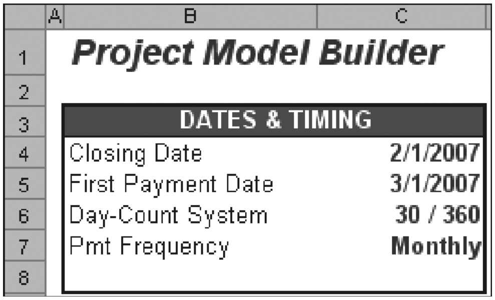
As this book suggested, the reader can develop a cash flow model from an entirely blank workbook. This Model Builder section begins with the assumption that the reader is staring at Sheet1 of a blank workbook in Excel. This first Model Builder exercise introduces basic labels and the fundamental date and timing inputs.

Keep in mind two minor details while trying to follow the Model Builder section. The first detail is formatting. The most important formatting concept to remember is to format a cell with the correct data format. Many cells have different data formats such as dates, numerical values, text, and the like. If a number is appearing instead of a date, it could mean that the format for that cell is set to number and should be changed to date. Be mindful of cell data formats.

Another detail is the color scheme that has been used. Not only does it make the model more professional looking, but grouping related sections together by color makes it easier to read. There are no specific instructions on formatting, but it is very useful to have a well-formatted model.

If there is any question as to what a completed section or formula should look like, each Model Builder section (along with a completed Project Model) can be found on the CD-ROM that comes with this book. Otherwise the first Model Builder begins with:

1. When Microsoft Excel is first opened most systems' default layout is to begin with three worksheets labeled Sheet1, Sheet2, and Sheet3. The first Model Builder exercise requires two worksheets. The worksheet tabs should be renamed from Sheet1 and Sheet2 to **Inputs** and **Hidden**, respectively.
2. Instead of beginning on column A, this column should be adjusted to a width of 1.0. The reason for this is readability, since some formatting can be cut off by the edge of the screen. With column A adjusted, move to cell B1 and enter, **Project Model Builder**. Make sure to change the font to blue and bold. In most financial models assumptions that are variables to be altered are denoted in bold blue font. This cell should be named **ProjName**. If naming cells is unclear, jump ahead to the Toolbox section at the end of this chapter.



Project Model Builder

DATES & TIMING

Closing Date	2/1/2007
First Payment Date	3/1/2007
Day-Count System	30 / 360
Pmt Frequency	Monthly

FIGURE 1.3 Basic inputs for dates and timing.

3. Moving under the transaction name in cell B3, input the title **Dates & Timing**. Now starting in cell B4 and continuing through to cell B7, enter the following text into the cells respectively: **Closing Date**, **First Payment Date**, **Day-Count System**, and **Pmt Frequency**. Highlight cells B3:C8 and format the area with a border. At this point, aside from stylistic preference, the upper corner of the Inputs sheet should look like Figure 1.3.
4. Before filling in starting values for the dates and timing inputs, switch over to the Hidden sheet to create data validation lists. The first one that will be needed is the day-count system. Input **Day-Count System** into cell A5 as a header. The list will start in cell A6 with the first type of day-count system that can be activated, cell A7 with the second, and so on. For this model input **30 / 360** into cell A6, **Actual / 360** into cell A7, and **Actual / 365** into cell A8. Highlight A6:A8 and name the range **lstDayCountSys**.
5. Similarly, it is inefficient to constantly enter what type of payment frequency there is. So, in cell A10, enter the heading **Payment Frequency**. Starting in cell A11 and continuing down a cell for each label, enter the following: **Monthly**, **Quarterly**, **Semiannual**, **Annual**. Highlight the cells that were just entered (A11:A14) and name them **lstPaymentFreq**.
6. Switching back to the Inputs sheet, cells C4 and C5 contain date-formatted values, for now enter **02/01/07** for the closing date and **3/01/07** for the First Payment Date. Name these cells **ClosingDate** and **FirstPayDate**, respectively.
7. The two named ranges that were created from the Hidden sheet are used for the Column C values corresponding to cells B6 and B7. In C6 use the data validation function to create a list using = lstDayCountSys as the source. Repeat for cell C7 using = lstPaymentFreq as the source. Name these two cells **DayCountSys** and **PmtFreq**, respectively. Similar to naming ranges and cells, if creating data validation lists is new jump to the Toolbox section at the end of this chapter.
8. Finally, switch back to the Hidden sheet. A formula cell needs to be created to represent the payment frequency numerically, depending on which payment frequency was selected. It is important to keep the payment frequency in monthly

terms because there are many functions in Excel that can adjust dates on a monthly basis (e.g., EDATE). In C11 input the following formula:

```
= IF(PmtFreq="Monthly",1,IF(PmtFreq="Quarterly",3,
  IF(PmtFreq="Semi-Annual",6,12)))
```

The last step for this section is to name C11 **PmtFreqAdd**. For a complete example of Model Builder 1.1, review the Excel file named *MB1-1.xls* on the CD-ROM.

DATES AND TIMING ON THE CASH FLOW SHEET

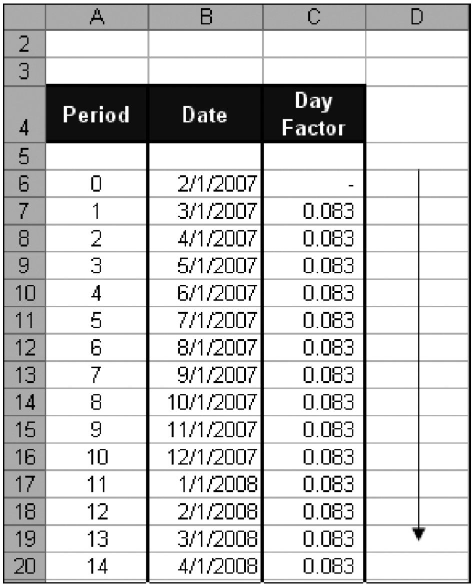
As a first introduction to the cash flow sheet, a general overview is helpful. The cash flow sheet is the section that will ‘‘move’’ the cash created by the asset amortization engine through the transaction’s priority of payments according to the user defined inputs and assumptions. Dates and timing are typically the first three columns or rows seen on this sheet because the entire concept of cash flow modeling relies on time.

While many finance professionals are familiar with a balance sheet type model, where time progresses horizontally across rows, cash flow modeling is more flexible if time progresses vertically, up and down columns. The examples in this book are based off a vertical progression of time (see Figure 1.4).

The first of the three columns, usually the A column, should be for tracking the *period*. The period is a numeric representation of time as it progresses. There is usually a period 0 that represents the closing date, with period 1 being the first payment date. Thereafter, the period simply grows in an ordinal fashion, increasing by one. Even if the model is switched to quarterly or semiannual frequency, the periods should not change. For example, in a monthly model, period 1 will be a month and period 2 will be the next month. When switched to a quarterly model, period 1 is a quarter and period 2 is the next quarter.

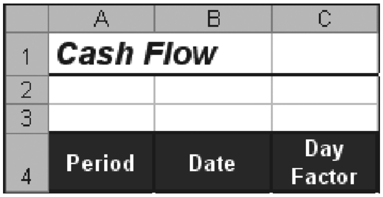
The second column is where the dates corresponding to the period are tracked. As mentioned, period 0 is the closing date and period 1 is the first payment date. Dates after that are determined by the desired frequency of payment. That is, if it is a monthly model, the next date after the first payment date is the next month; if it is a quarterly model, the next date is three months from the first payment date, and so on.

The third column is termed the day factor and is a precise calculation of the number of days between each period, represented as a fraction of the year. The calculation is quite simple: the number of days between the last period date and the current period date over the number of days in the year. This factor will provide a more precise breakdown of annual rates if an Actual/360 or Actual/365 day count system is selected. Refer to Figure 1.4, columns A, B, and C as an example.



	A	B	C	D
2				
3				
4	Period	Date	Day Factor	
5				
6	0	2/1/2007	-	
7	1	3/1/2007	0.083	
8	2	4/1/2007	0.083	
9	3	5/1/2007	0.083	
10	4	6/1/2007	0.083	
11	5	7/1/2007	0.083	
12	6	8/1/2007	0.083	
13	7	9/1/2007	0.083	
14	8	10/1/2007	0.083	
15	9	11/1/2007	0.083	
16	10	12/1/2007	0.083	
17	11	1/1/2008	0.083	
18	12	2/1/2008	0.083	
19	13	3/1/2008	0.083	
20	14	4/1/2008	0.083	

FIGURE 1.4 Time will progress vertically in this model.



	A	B	C
1	*Cash Flow*		
2			
3			
4	Period	Date	Day Factor

FIGURE 1.5 The beginning of the dates and timing section on the Cash Flow sheet.

MODEL BUILDER 1.2: CASH FLOW SHEET—DATES AND TIMING

1. The Cash Flow sheet is the third sheet in the model. Rename the third sheet from Sheet3 to **Cash Flow**.
2. In the Cash Flow sheet, Row 4 will be dedicated to headers. Input **Period** in cell A4, **Date** in cell B4, and **Day Factor** in cell C4.
3. The first column to work on is where the periods are tracked. In cell A6 input 0, followed by inputting 1 in the cell directly below, A7. Highlight both

of these cells, left click and hold down the small square on the bottom right cube of the highlighted cells, and drag the numbers down. As the cells are being dragged a box should appear with an increasing value. By dragging down, Excel automatically inputs increasing numbers in the cells. The increase is based on the difference between the two numbers that were originally highlighted. Since the model will only go out 30 years and the smallest periodicity option is a month, there is only a need for 360 periods. Stop dragging when the number reaches 360; all of the numbers between 0 and 360 should have populated into the cell range A6:A366.

4. The next column to work on is the Date column. Because it would be inefficient to constantly have to change this column every time the dates are changed for a transaction, the cells in this column are a formula. Looking back to the Inputs sheet, the first date that was created was the closing date, which is the initial date the transaction begins. This date always corresponds to the start of period 0. Similarly, the second important date on the Inputs sheet, the first payment date, always corresponds to period 1. The formula for the correct date must differentiate between periods 1 and 2. The use of an IF statement is the simplest method to make sure that the date for period 0 is always the closing date and that the date for period 1 is the first payment date.[2] The first part of the formula should look like this:

   =IF(A6=0,ClosingDate,IF(A6=1,FirstPayDate

   The rest of the periods grow by the increments that are set by the payment frequency cell (PMTFreq) on the Inputs sheet. The payment frequency cell is mainly set up for a model user and will only offer word values.

   Recall that on the Hidden sheet a cell named PMTFreqAdd was created that returns numeric values depending on the word value in PMTFreq. With a numeric value for the frequency between payments, the EDATE function can be used to grow the dates after period 1. The completed formula should look as follows:

   =IF(A6=0,ClosingDate,IF(A6=1,FirstPayDate,EDATE(B5,PmtFreqAdd)))

   The completed formula will return the Closing Date for period 0, the First Pay Date for period 1, and will then return a date that is PmtFreqAdd number of months from the First Pay Date. Copy this formula down to the final period.

5. The final column affecting dates and timing on the Cash Flow sheet is the day factor, which quantifies the time difference between periods as a fraction of the

[2] In general a high-quality financial model uses one consistent formula for every column in a section that involves timing. Often a model builder uses a different formula for period 0 versus the remaining periods because the remaining period calculations are based on period 0. The use of IF statements allows for consistent formulas that make a model easier to audit, faster to build, and less prone to errors.

year. As mentioned earlier, depending on the type of day count system that is selected there can be varying amounts of time between dates.

6. Recall that a cell specifically for selecting the day count was created on the Inputs sheet (DayCountSys). This cell has three options: 30/360, Actual/360, and Actual/365. The functionality needed in the model is to have all of the calculations based on the day count system switch when the cell DayCountSys is changed on the Inputs sheet.
7. The actual formula for C6 is composed of many different parts. The first part is due to the fact that period 0 will always have a zero value between dates, since there is no date prior to 0. An IF statement for period 0 takes care of this problem:

```
=IF(A6=0,0
```

8. The next part of the formula is written to calculate the time difference for the 30/360 system. As explained earlier, the DAYS360 function is perfect for this task:

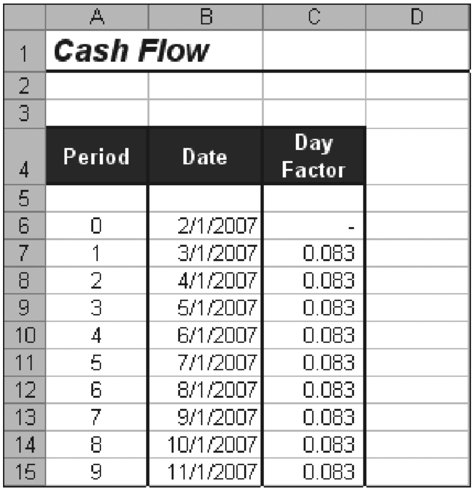
```
=IF(A6=0,0,IF(DayCountSys="30 / 360",DAYS360(B5,B6)/360,
```

   First, notice that in order to handle the possible changing value of DayCountSys, a second IF statement was inserted. Next, the DAYS360 function is used if 30/360 is selected. This function takes the date from the previous period (B5) and calculates the difference in days between the current period (B6). Because the return value for the DAYS360 function is in days and all of the calculations relying on the Day Factor are annual rates, the return value needs to be divided by 360 to produce a fraction value (assuming a 360 day year as the 30/360 system does).
9. The next part of the formula is when Actual/360 is used. Another IF statement is required to identify when DayCountSys is set to Actual/360. If the statement is true the Actual/360 calculation is very simple; it is the difference between the current date and the date the period before.

```
= IF(A6=0,0,IF(DayCountSys="30 / 360",DAYS360(B5,B6)/360,
  IF(DayCountSys="Actual / 360",(B6−B5)/360,
```

   Remember that the return value will also be in days and needs to be divided by 360 for Actual/360.
10. The final step is when Actual/365 is selected. Because there are only three possible values for DayCountSys and if the first two IF statements are false, the setting must be Actual/365. This means that the calculation for Actual/365, which is virtually the same as Actual/360 except that the divisor is 365, can be set as the final FALSE value for the last IF statement. The complete formula should read:

```
= IF(A6=0,0, IF(DayCountSys="30 / 360", DAYS360(B5,B6)/360,
  IF(DayCountSys="Actual / 360",(B6−B5)/360,(B6−B5)/365)))
```



	A	B	C	D
1	Cash Flow			
2				
3				
4	Period	Date	Day Factor	
5				
6	0	2/1/2007	-	
7	1	3/1/2007	0.083	
8	2	4/1/2007	0.083	
9	3	5/1/2007	0.083	
10	4	6/1/2007	0.083	
11	5	7/1/2007	0.083	
12	6	8/1/2007	0.083	
13	7	9/1/2007	0.083	
14	8	10/1/2007	0.083	
15	9	11/1/2007	0.083	

FIGURE 1.6 The completed dates and timing section on the Cash Flow sheet.

Finally, copy this formula down to the final period. See Excel file *MB1-2.xls* in the Ch01 folder on the CD-ROM for complete Excel examples.

TOOLBOX

Naming Cells and Ranges

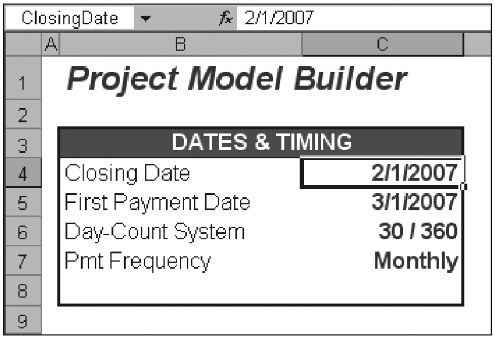
Naming cells and ranges is a technique that saves time, improves clarity, and reduces errors when working with financial models. At the most basic level, naming a cell or range changes how the cell or range is referred to in formulas and functions. For example, if cell A1 were named Input1, and cell A2 had the value 5 in it, a formula in cell A3 that added cells A1 and A2 would appear as:

=Input1+A2

Also, a named cell or range changes any reference to that cell or range from a relative reference to an absolute reference. This change reduces errors when named cells are used in formulas that are copied down over multiple cells. Normally the referenced cell would have to be "locked" using dollar signs (F4 shortcut), but a named range does not change as it is copied down.

Naming a cell or range can be accomplished in one of two ways. The formal method is:

1. Select a cell or a range of cells.



ClosingDate ▾ fx 2/1/2007

Project Model Builder

DATES & TIMING	
Closing Date	2/1/2007
First Payment Date	3/1/2007
Day-Count System	30 / 360
Pmt Frequency	Monthly

FIGURE 1.7 The active cell is C4 which shows the value in the Formula Bar and the cell name in the Name Box.

2. From the menu bar, click Insert, Name, and Define.
3. Enter the name of the cell or range in the field titled Names in Workbook. Then click Add or OK. Add would be used if multiple names were to be added from the Define Name dialogue box. Also note the Refers to textbox at the bottom of the Define Name dialogue box. This is where the reference is established and edited if changes need to be made.

The simpler solution to naming a range is to select the cell or range of cells on the active sheet and enter the desired name directly in the Name Box and press Enter. The Name Box is in the upper left corner of the Excel window, to the left of the Formula Bar. If a cell or range is already named, the name appears in the Name Box when the cell or range is selected. See the named cell ClosingDate (cell C4) in Figure 1.7.

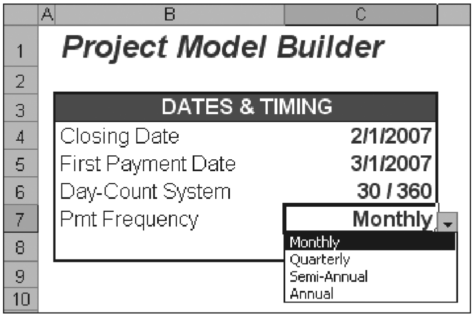
It is a good habit to name all inputs and important ranges because they help the model builder and any operators understand formulas faster. This is particularly important if the model uses Visual Basic for Applications to a high degree. The named cells and ranges can be understood and accessed faster than with the conventional Column/Row reference.

Data Validation Lists

Using data validation lists for inputs that can have a value from a set list saves time and reduces errors for model operators. Data validation lists essentially take a list of values in the workbook and offer them for selection in a combo box format for a specific cell.

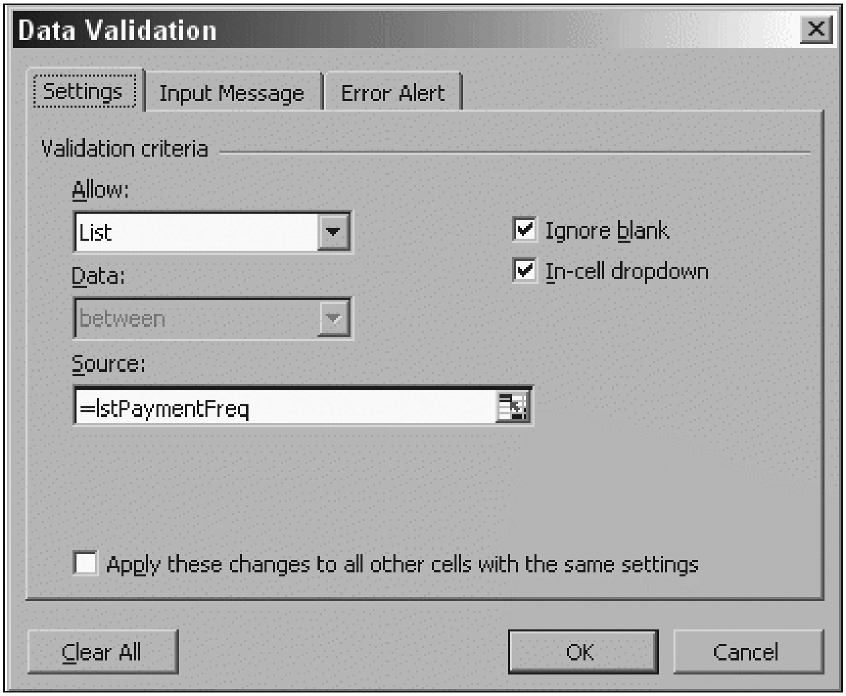
Creating a data validation list is very easy since it is a prebuilt Excel operation:

1. The first step is creating the list that will be the possible values. In the case of Figure 1.8 the list would be a range of four cells with the first cell



	A	B	C
1		*Project Model Builder*	
2			
3		DATES & TIMING	
4		Closing Date	2/1/2007
5		First Payment Date	3/1/2007
6		Day-Count System	30 / 360
7		Pmt Frequency	Monthly
8			
9			
10			

FIGURE 1.8 An example of a data validation list.



Data Validation

Validation criteria

Allow: List

Data: between

Source: =lstPaymentFreq

☑ Ignore blank

☑ In-cell dropdown

□ Apply these changes to all other cells with the same settings

FIGURE 1.9 Create a data validation list in the Data Validation dialogue box.

having the label Monthly, the second cell Quarterly, and so on. For organizational purposes, this list should be stored on a separate sheet. (In the Model Builder, this sheet is known as Hidden because it will be hidden from view later.)

2. Once the range is created it has to be named. A good convention for lists is to begin the name with *lst* so it can be identified when searching through a list of names.
3. The last step is to select the cell where the list will be active (cell C7 in Figure 1.8). Once the cell is selected, in the menu bar click Data and then Validation. The Data Validation dialogue box appears. Click the Allow arrow and select List. The Source textbox appears. This is where the range name that references the list should be entered as a formula as shown in Figure 1.9.

EDATE

An extremely useful function to increase the date in an orderly manner over time is *EDATE*. EDATE has two inputs: a date and a numeric value.

=EDATE(starting date, numeric value)

The function treats the numeric value in terms of number of months. EDATE returns the date that is the number of months away from the numeric value. For example, if the cell containing the date had a value of 07/15/05 and the numeric value was 1, EDATE would return 08/15/05. Similarly, if the 1 had been a 2 it would return 09/15/05. EDATE also works with negatives and would return 06/15/05 if −1 were used as the numeric cell reference.

For EDATE to work, the Analysis ToolPak must be installed—otherwise the workbook will be full of NAME errors. To install the Analysis ToolPak, in the menu bar click Tools and Add-Ins. In the Add-Ins dialogue box, select the Analysis ToolPak checkbox and click OK.

CHAPTER 2

Asset Cash Flow Generation

The first calculations to do are related to cash flowing into the structure, which is known as *asset cash flow generation*. Depending on the type of transaction, there can be wide variability as to how cash is generated. For most structured finance transactions, the asset cash flow generation is usually in the form of principal and interest from the assets. Some transactions, particularly those involving infrastructure, rely on consultant studies or simulations that project future cash flow. Regardless of the type of transaction, a cash flow model must have some type of assumption for the periodic inflow of cash.

With a structured framework, there are many intricate and distinct parts to creating the cash flows that assets produce such as interest calculations, default and prepayment assumptions, recoveries, and the like. Explaining in detail how each of these parts work is fundamental to accurate modeling and also beneficial to building a financial modeler's skill. Explaining the most basic concepts first, this chapter then focuses on creating a notional asset cash flow—that is, cash flows without giving affect to prepayments and defaults. Once this is complete, each detail will be introduced one at a time to create the actual asset cash flow.

Prior to explaining exactly how to model structured asset cash flows, the answers to two questions need to be taken into account: (1) How will the assets exist over time? (2) And what and how much data on the assets are there? Depending on the answers to those two questions two methodologies can be employed: loan level generation and representative line generation. Both methods produce a set of cash flows that can be used for modeling purposes, but each method used on the same assets can produce remarkably different results.

LOAN LEVEL VERSUS REPRESENTATIVE LINE AMORTIZATION

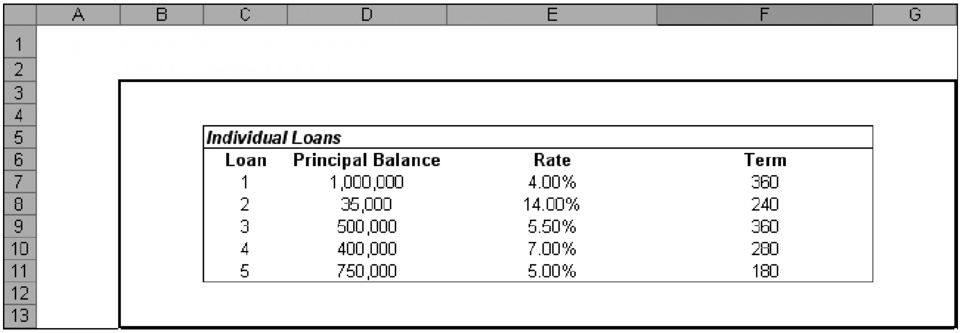
The answers to those questions asked in the previous section—about how assets exist over time and the amount and quality of data—determine which methodology to use for generating cash flow. So, how do assets exist over time? This concept can be a little cryptic to people new to structured finance. At the most basic level, the assets in a structured finance transaction can either remain the same in a definitive pool or they can revolve. A definitive pool means that the assets that the transaction starts with are the only assets available to the transaction. If there were 10,000 mortgages in a pool, those specific mortgages would remain in the transaction through the deal's life. As the mortgages pay off, refinance, or default, no new mortgages are added. This is usually seen in buy and hold transactions or deals where investors buy into a specific pool.

With a definitive pool, the most preferred method of asset amortization is a *loan level analysis* where each individual asset is amortized based on the asset's individual characteristics. Each one of the 10,000 mortgages would have its own amortization schedule that would be aggregated to form one transaction amortization schedule. That aggregated amortization schedule would provide the cash flow generation for the transaction.

Whether a loan level analysis is possible depends on the answer to the second question: What type of information is available for the assets? Detailed data for each asset is required to complete a loan level analysis. At minimum, information for each loan's term, balances, and interest rate would be required. This is why it is extremely important for companies to track extensive data on their assets and be able to access and transmit such data.

If no such data exists, the other option is to use a *representative line analysis*. A representative line can be thought of as an aggregation of similar loan's attributes as a single loan or line of information. For instance, if there were only five similar loans in a pool and each loan's specific balance, term, and rate was unknown, a loan level analysis could not be completed. However, if information on the aggregate pool of the five loans was available, such as the sum of all the balances, the weighted average terms, and the weighted average rate, an amortization schedule using those figures could be created. In such a case enough information exists to create and amortize a fictitious loan that is representative of the five actual loans in the pool.

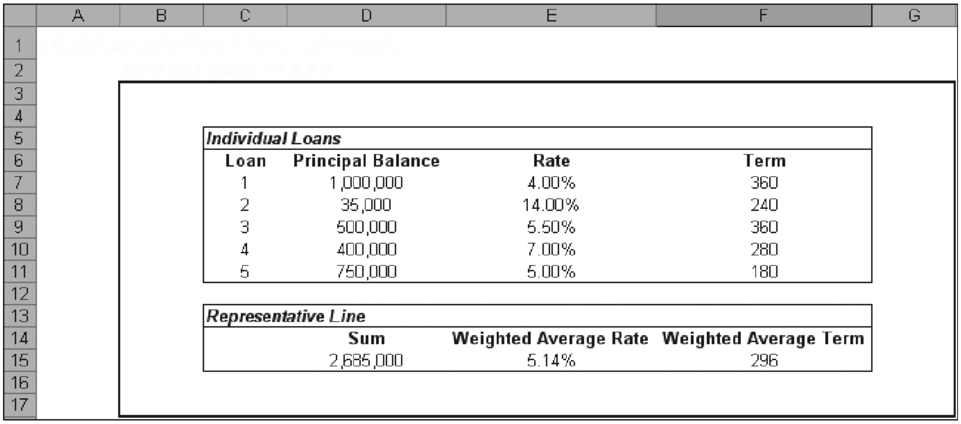
Many people find representative lines a difficult concept to grasp, so a basic example of taking a number of loans and making one representative line is the best way to get a feel for the concept. Assume the following loans shown in Figure 2.1.



Individual Loans

Loan	Principal Balance	Rate	Term
1	1,000,000	4.00%	360
2	35,000	14.00%	240
3	500,000	5.50%	360
4	400,000	7.00%	280
5	750,000	5.00%	180

FIGURE 2.1 Each individual loan has a principal balance, rate, and term.



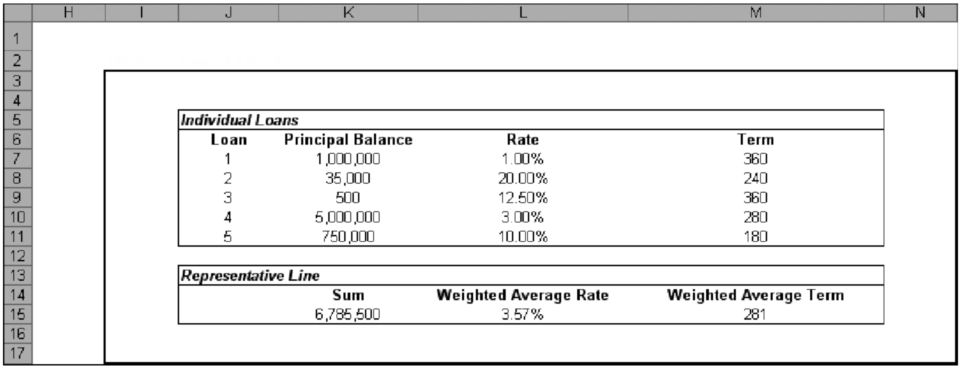
Individual Loans

Loan	Principal Balance	Rate	Term
1	1,000,000	4.00%	360
2	35,000	14.00%	240
3	500,000	5.50%	360
4	400,000	7.00%	280
5	750,000	5.00%	180

Representative Line

Sum	Weighted Average Rate	Weighted Average Term
2,685,000	5.14%	296

FIGURE 2.2 A representative line can be created from the individual loans by summing the principal balances and calculating the weighted average rate and term.



Individual Loans

Loan	Principal Balance	Rate	Term
1	1,000,000	1.00%	360
2	35,000	20.00%	240
3	500	12.50%	360
4	5,000,000	3.00%	280
5	750,000	10.00%	180

Representative Line

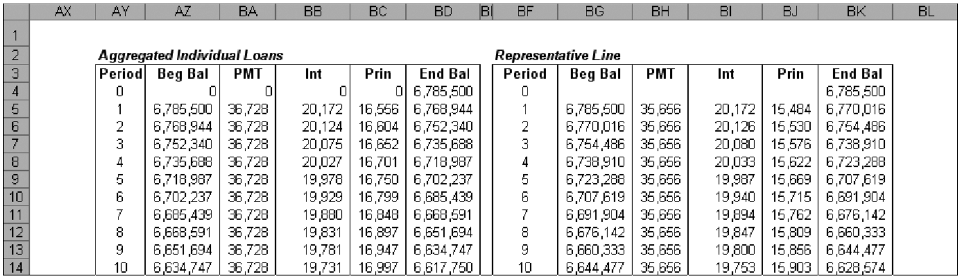
Sum	Weighted Average Rate	Weighted Average Term
6,785,500	3.57%	281

FIGURE 2.3 The five loans presented here have diverse principal balances, rates, and terms.

By summing the balances and calculating the weighted average rate and term, a single representative line could be made. This would create a single line of information that provides a relatively good summary of the pool of loans as shown in Figure 2.2.

This single line is much easier to create cash flows from then having to create five separate cash flows for each loan in the pool. However, subjective analysis is required when making representative lines because the more diverse a pool of assets is, the more distorted the cash flows are when making a single representative line. Imagine having the data tape shown in Figure 2.3.

In this example, the rates and terms between assets are very different, which means that the aggregate interest and periodic cash flows are distinctly different than



Aggregated Individual Loans

Period	Beg Bal	PMT	Int	Prin	End Bal
0	0	0	0	0	6,785,500
1	6,785,500	36,728	20,172	16,556	6,768,944
2	6,768,944	36,728	20,124	16,604	6,752,340
3	6,752,340	36,728	20,075	16,652	6,735,688
4	6,735,688	36,728	20,027	16,701	6,718,987
5	6,718,987	36,728	19,978	16,750	6,702,237
6	6,702,237	36,728	19,929	16,799	6,685,439
7	6,685,439	36,728	19,880	16,848	6,668,591
8	6,668,591	36,728	19,831	16,897	6,651,694
9	6,651,694	36,728	19,781	16,947	6,634,747
10	6,634,747	36,728	19,731	16,997	6,617,750

Representative Line

Period	Beg Bal	PMT	Int	Prin	End Bal
0					6,785,500
1	6,785,500	35,656	20,172	15,484	6,770,016
2	6,770,016	35,656	20,126	15,530	6,754,486
3	6,754,486	35,656	20,080	15,576	6,738,910
4	6,738,910	35,656	20,033	15,622	6,723,288
5	6,723,288	35,656	19,987	15,669	6,707,619
6	6,707,619	35,656	19,940	15,715	6,691,904
7	6,691,904	35,656	19,894	15,762	6,676,142
8	6,676,142	35,656	19,847	15,809	6,660,333
9	6,660,333	35,656	19,800	15,856	6,644,477
10	6,644,477	35,656	19,753	15,903	6,628,574

FIGURE 2.4 Notice the slight differences in payment, interest, and principal when using a representative line versus aggregating individual loan schedules.

the cash flows generated by a representative line. Figure 2.4 presents the periodic cash flows of each method.

In the example, the two cash flows are different on a periodic basis and in aggregate. The periodic differences can be problematic for the upcoming liability structure in the transaction. There could be periods of liquidity shortfall or triggers tripped when using one of the methodologies versus the other. Also, the aggregate interest sum of the individual loans comes to $2,926,259, while the interest sum of the representative line is $3,233,736. The balance difference is small in this example, but if the principal balance were $1 billion then the difference between the two methods could be in the tens of millions! For a detailed example of representative lines in Excel, see Rep_Lines.xls in Ch 02's Additional Files folder on the CD-ROM.

While it appears that loan level asset generation is ideal if the information is available, a representative line methodology is more appropriate when assets are revolving. A revolving structure is the other form in which assets can be financed. This means that assets can be added to a pool as the transaction continues throughout time. The deal might start with 10,000 mortgages; but some of those mortgages might pay off, refinance, or default over time and others can be added. There is no guarantee that the assets started with will be the assets in the transaction any time in the future. The key to this state of existence for assets is that there are *eligibility criteria* for the assets in the pool.

Eligibility criteria are a fixed set of rules that dictate what assets can be added to a revolving structure and are essential to modeling such a transaction. Typically transactions with revolving assets are modeled assuming the pool of assets is filled to the deal size limit with the absolute worst set of assets the eligibility criteria allows. For example, if the minimum asset yield the eligibility criteria allows is 7 percent, then a representative line created for this transaction should use 7 percent as the asset yield. The reason this is important is that there is no guarantee, other than the eligibility criteria, on the composition of the asset pool. It is also important because stress testing often involves creating a worst-case scenario and, in the case of a revolving transaction, the worst possible pool are the adverse limits of the eligibility

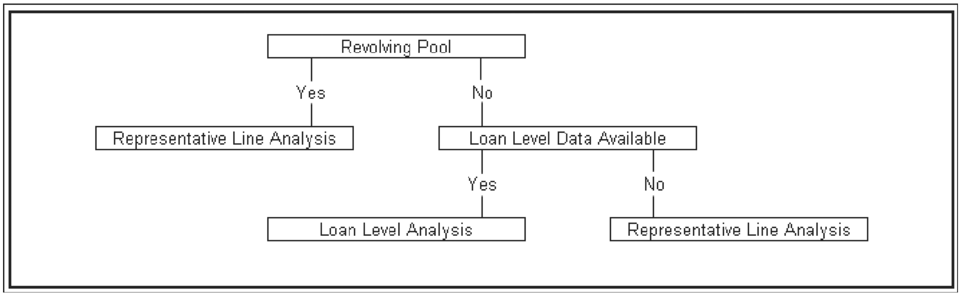

Revolving Pool
Yes
No
Representative Line Analysis
Loan Level Data Available
Yes
No
Loan Level Analysis
Representative Line Analysis

FIGURE 2.5 The type of analysis used to create cash flows depends on a number of factors.

criteria. See Figure 2.5 for a decision tree that summarizes whether loan level or representative line asset generation should be used.

HOW ASSET GENERATION IS DEMONSTRATED IN MODEL BUILDER

The easiest asset amortization to learn is a single loan or representative line created within Excel sheet formulas. This is a useful level at which to start because, as a more detailed asset generation tool is created, the basics can all be found in a single loan. This is why Project Model Builder that you started to create in Chapter 1 uses a single asset or single representative line. To create a more robust model that can handle multiple representative lines or tens of thousands of loans some type of computer code is required. Visual Basic for Applications (VBA) is perfect for such a task and is used frequently in more advanced models. A highly developed asset generation tool has the flexibility to accept any type of loan or amortize loans in groups based on many different criteria.

ASSET GENERATION ON THE INPUTS SHEET

Whether a loan level or representative line methodology is used, the asset generation should be easily manipulated from the Inputs sheet. In a loan level type approach there would need to be a sensitivity selector that changes calculation variables or switches between cash flows calculated and imported from another model. For Project Model Builder, a single representative line methodology will be used and controlled from the Inputs sheet.

The single representative line input for Project Model Builder is based on a pool of assets that pay principal and interest and therefore needs to include, at minimum, the following information: the original balance, the current balance, the interest rate, the original term, and the remaining term. Additional information is required, particularly if the assets are generating yield off of a floating rate. Since floating rate assets are certainly a possibility in many structured transactions, this text will explain and implement the option of floating rate asset amortization in Project Model Builder.

Fixed Rate Amortization Inputs

A fixed rate amortization with a level payment is very straightforward. The periodic cash flows are created by first calculating the level payment using the PMT function, next calculating the interest for the period depending on the period's beginning balance, then subtracting the interest from the payment to get the periodic principal, and finally subtracting the periodic principal from the beginning balance to get the end balance. Doing this process for each period until the balance is 0 produces a series of principal and interest cash flows.

Floating Rate Amortization Inputs

While most deals are modeled with a fixed rate, there are a number of deals that have floating rate assets, which are more complex to model correctly. A floating rate asset can have a number of additional attributes that could alter the cash flow:

- Rate index
- Rate margin
- Lifetime rate cap
- Lifetime rate floor
- Periodic rate cap
- Periodic rate floor
- Rate reset frequency
- First reset date

The forward projecting assumption for the underlying index primarily drives the interest calculation. This assumption is typically a vector such as a forward curve or rating agency stressed curve. The margin is added to the periodic rate. A *lifetime cap* is a hedge that prevents the index rate from exceeding a defined level over the lifetime of the deal. Conversely, a *lifetime floor* is a hedge that prevents the index rate from decreasing below a defined level over the lifetime of the deal. Closely related is a *periodic rate cap* and *floor*. Instead of preventing the index rate from going beyond a defined level over the lifetime of the deal, a periodic cap prevents breaching certain levels during any given period. Lastly, the *rate reset frequency* is how often the rate resets and changes.

A more general concept to consider when working with a floating rate amortization is how the cash flow is affected as the rate changes. One of two things can happen when the rate changes: the payment changes or the term of the loan changes. Many assets have payments that change as interest rates change. This is the source of payment shock that is often witnessed with variable rate products. Other assets are set up to keep a fixed payment, so when the rate changes the same payment is made, but the term is affected. If rates increase beyond expectations then less principal will be paid each period and the term will have to be extended. Likewise, if rates decrease below expectations, more principal will be paid each period and the term will have to be decreased.

Another point to consider when working with floating rate assets is that the index the assets are based off of becomes yet another attribute that could cause the need for multiple representative lines. Some pools may have floating rate assets based off of different indexes, which would require the use of a separate representative line for each index.

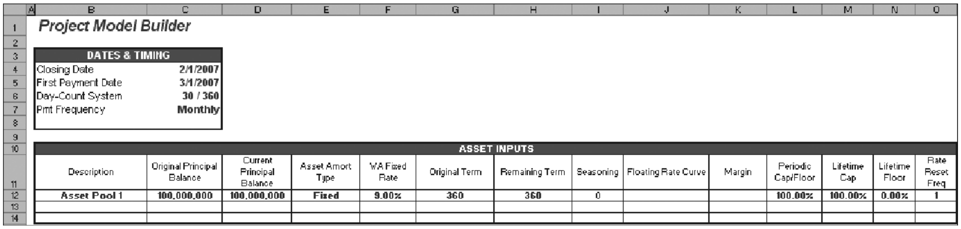
Organizing the different indexes is the final point to take into account. Each index is a projected vector of rates that is as long as the number of periods. In the example model, there is a possibility of 360 periods, so a single projected interest rate vector such as one-month LIBOR (London Interbank Offered Rate), will take up 360 cells. Storing the indexes on the Inputs sheet would be inefficient since there will be other rate vectors encountered later in the model (e.g., default rates, prepayment rates, etc.) all taking up a large amount of cells. For this reason, a separate sheet for the vectors is created.

MODEL BUILDER 2.1: INPUTS SHEET ASSET ASSUMPTIONS AND THE VECTORS SHEET

This section starts at the Inputs sheet, where the relevant data for a representative line is entered and stored. The inputs used for asset generation create the notional amortization schedule. After this is complete, the asset generation is further developed to account for prepayment, default, and recovery assumptions. Figure 2.6 glimpses at the Inputs assumptions that are created.

To create the inputs necessary for a notional amortization, perform the following steps:

1. In cell B10 create a label for this section of the Inputs sheet by entering **ASSET INPUTS**.



Project Model Builder

DATES & TIMING	
Closing Date	2/1/2007
First Payment Date	3/1/2007
Day-Count System	30 / 360
Pmt Frequency	Monthly

ASSET INPUTS

Description	Original Principal Balance	Current Principal Balance	Asset Amort Type	WA Fixed Rate	Original Term	Remaining Term	Seasoning	Floating Rate Curve	Margin	Periodic Cap/Floor	Lifetime Cap	Lifetime Floor	Rate Reset Freq
Asset Pool 1	100,000,000	100,000,000	Fixed	9.00%	360	360	0			100.00%	100.00%	0.00%	1

FIGURE 2.6 The section of the Inputs sheet that controls the asset assumptions.

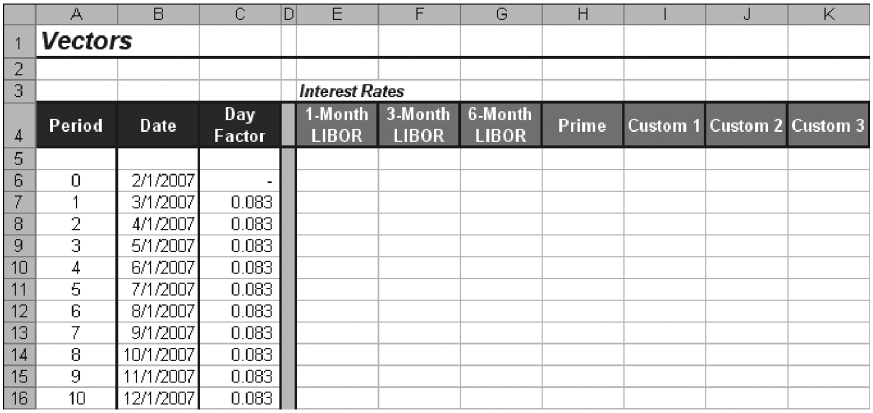
2. Range B11:O11 contains the labels for the variables that will be entered below on row 12. Enter the following text into each labeled cell:
   B11: **Description**
   C11: **Original Principal Balance**
   D11: **Current Principal Balance**
   E11: **Asset Amort Type**
   F11: **WA Fixed Rate**
   G11: **Original Term**
   H11: **Remaining Term**
   I11: **Seasoning**
   J11: **Floating Rate Curve**
   K11: **Margin**
   L11: **Periodic Cap/Floor**
   M11: **Lifetime Cap**
   N11: **Lifetime Floor**
   O11: **Rate Reset Freq**
3. It is much easier to work with values for the assumptions, even if they will be changed later. Beginning with cell B12, enter the label **Asset Pool 1**. This variable is self-explanatory; it is a description of the asset representative pool. For this cell and all of the following, make sure that the font format is blue and bold because they are variables that may change. Also, name this cell **AssetDes1**.
4. In cell C12, enter the value **100,000,000**. This is the original principal balance of the pool. It is important to know the original principal balance because the payments are calculated off of it. Name this cell **AssetOrgBal1**.
5. In cell D12, enter the value **100,000,000**. This is the current principal balance of the pool and can be different from the original balance because some loans in the pool may have been partially amortized. For simplicity, it is easier to assume that all of the assets are new and the current balance is the same as the original balance. Name this cell **AssetCurBal1**.
6. Cell E12 contains the amortization type (whether it is fixed interest, floating interest, or custom) and has a value from a data validation list. To do this, go back to the Hidden sheet and enter **Asset Amortization Type** in cell A16, **Fixed** in cell A17, **Floating** in cell A18, and **Custom** in cell A19. While still on the Hidden sheet, name the range A17:A19 **lstIntType**. Go back to the Inputs sheet and create a data validation list for cell E12 using **lstIntType** as the name reference. Cell E12 will be a reference for cells that depend on the type of amortization. Finally, name cell E12 **AssetIntType1**.
7. If the asset pool is going to be amortized on a fixed rate basis, an assumption is needed for that fixed rate. In cell F12 enter **9.00%** for now as a placeholder. Name this cell **AssetFxdRate1**.
8. Cell G12 contains the original term assumption of the pool. Remember that the maximum number of periods the model has been designed for is 360, so this value needs to be less than or equal to that. For now, assume that the original term is 360 and enter **360** in cell G12. Name this cell **OrgTerm1**.

9. Cell H12 is for the remaining term of the pool. This value can be equal to the original term if all of the assets are new but is typically less than the original term. As a starting variable assume that the remaining term is 360 and enter **360** in cell H12. Name this cell **RemTerm1**.
10. The next cell, seasoning (I12), is actually a calculation derived from the original term and the remaining term. The formula that should be entered is:

    = OrgTerm1−RemTerm1

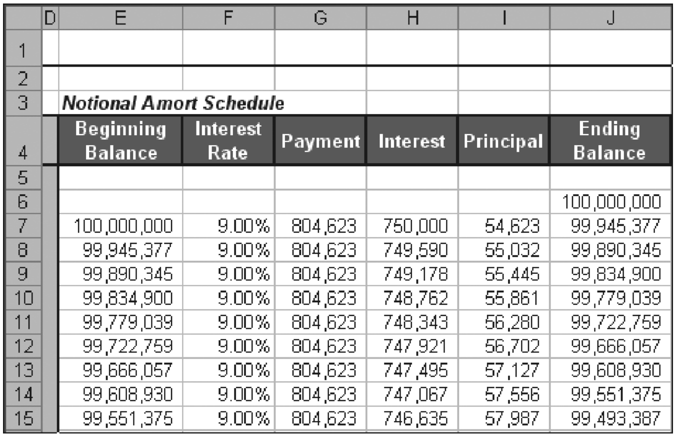
    While extremely easy, the seasoning should not be underestimated in terms of importance. Prepayment and loss amounts will heavily rely on the seasoning to determine accurate projections. Name this cell **Age1**.
11. Cell J12 contains the name of the curve (synonymous with *vector*) that will be used to calculate interest if the asset is based on a floating rate. This value will be a validation list similar to those done before, but with a twist. The names of the rates make up the range that is being referenced for the list, but they will not be contained on the Hidden sheet. Instead, the names of the curves will be created on a new sheet named *Vectors*.

    To create the Vectors sheet, insert a new worksheet and name the sheet **Vectors**. The vectors will be over time, so time labels are needed. In cell A4 on the Vectors sheet, reference cell A4 of the Cash Flow sheet by entering **=** and then clicking on cell A4 of the Cash Flow sheet. Copy the reference over the range A4:C366. The formatting is most likely incorrect, so copy and paste special the formatting for those three columns from the Cash Flow sheet. Also, there will probably be zero values for cells A5, B5, and C5, which can be deleted since they are unnecessary.
12. In cell E4 enter the first vector name which is **1-month LIBOR**, in cell F4 enter **3-month LIBOR**, in cell G4 enter **6-month LIBOR**, in cell H4 enter **Prime**, in cell I4 enter **Custom 1**, in cell J4 enter **Custom 2**, in cell K4 enter **Custom 3**. Now name the range E4:K4 **lstInterestRates**. The area should look like Figure 2.7.
13. Back on the Inputs sheet, create the data validation list using **lstInterestRates** as the named range reference. Name cell J12 **AssetFltIndx1**. The full utility of creating the data validation list in this manner will become clear later, when the formulas that need to select between the floating rates are explained.
14. The amount that is earned in addition to the floating rate, the margin, is stored in cell K12. For now assume this amount is 1.50 percent. Name this cell **AssetMarg1**.
15. The Periodic Rate Cap/Floor is the amount that the rate can go up or down each period and will be stored in cell L12. For now assume that there is essentially no Cap/Floor by entering **100%**. (It would be extraordinarily rare to have a rate increase or decrease by 100 percent in one month.) Name this cell **AssetPdCapFl1**.



	A	B	C	D	E	F	G	H	I	J	K
1	*Vectors*										
2											
3					*Interest Rates*						
4	Period	Date	Day Factor		1-Month LIBOR	3-Month LIBOR	6-Month LIBOR	Prime	Custom 1	Custom 2	Custom 3
5											
6	0	2/1/2007	-								
7	1	3/1/2007	0.083								
8	2	4/1/2007	0.083								
9	3	5/1/2007	0.083								
10	4	6/1/2007	0.083								
11	5	7/1/2007	0.083								
12	6	8/1/2007	0.083								
13	7	9/1/2007	0.083								
14	8	10/1/2007	0.083								
15	9	11/1/2007	0.083								
16	10	12/1/2007	0.083								

FIGURE 2.7 The new Vectors sheet should include basic timing and rate information at this point.



	D	E	F	G	H	I	J
1							
2							
3		*Notional Amort Schedule*					
4		Beginning Balance	Interest Rate	Payment	Interest	Principal	Ending Balance
5							
6							100,000,000
7		100,000,000	9.00%	804,623	750,000	54,623	99,945,377
8		99,945,377	9.00%	804,623	749,590	55,032	99,890,345
9		99,890,345	9.00%	804,623	749,178	55,445	99,834,900
10		99,834,900	9.00%	804,623	748,762	55,861	99,779,039
11		99,779,039	9.00%	804,623	748,343	56,280	99,722,759
12		99,722,759	9.00%	804,623	747,921	56,702	99,666,057
13		99,666,057	9.00%	804,623	747,495	57,127	99,608,930
14		99,608,930	9.00%	804,623	747,067	57,556	99,551,375
15		99,551,375	9.00%	804,623	746,635	57,987	99,493,387

FIGURE 2.8 The Notional Amortization schedule is created on the Cash Flow sheet.

16. Cell M12 contains the Lifetime Cap, which is the absolute maximum that the interest rate can increase to. Just as the Periodic Rate Cap/Floor enter **100%** for now so it is basically deactivated. Name this cell**AssetLifeCap1**.
17. Cell N12 contains the Lifetime Floor, which is the absolute maximum that the interest rate can decrease to. This will be deactivated now by entering **0%**. Name this cell **AssetLifeFloor1**.
18. The last cell of this section, cell O12, contains a numeric value that represents the frequency that the rate changes or resets. Depending on the type of rate, there

can be any frequency of rate reset. If the rate resets every period, then this value is one. If it resets every three periods, then it is three, etc. For now assume that the rate can reset every period by entering 1. Name this cell **AssetRateReset1**.

ASSET GENERATION ON THE CASH FLOW SHEET

As the name suggests, the Cash Flow sheet is where the calculations for the asset cash flow generation take place. When using a representative line methodology, a notional schedule of amortization needs to be created. The schedule is called *notional* because it is a hypothetical amortization that does not take into account prepayments, defaults, and recoveries. Later, prepayments, defaults, and recoveries will use the notional schedule to calculate the actual amortization schedule.

The notional asset amortization schedule uses six columns, one for each of the following: Beginning Balance, Payment, Periodic Interest Rate, Interest, Principal, and Ending Balance. Most models, including Project Model Builder, assume a level (mortgage style) payment system, making the calculations very straightforward. The only confusing part can be interest, particularly if a floating rate is desired; otherwise the periodic rate is fixed every period.

MODEL BUILDER 2.2: NOTIONAL ASSET AMORTIZATION ON THE CASH FLOW SHEET

1. As in prior Model Builder sections, the first step is to create labels for the calculations. On the Cash Flow sheet, in cell E3, enter **Notional Amort Schedule**. This will be a title for the next six columns that make up the schedule.
2. In cell E4 enter **Beginning Balance**, in cell F4 enter **Interest Rate**, in cell G4 enter **Payment**, in cell H4 enter **Interest**, in cell I4 enter **Principal**, and finally, in cell J4, enter **Ending Balance**.
3. It may seem a bit counterintuitive, but the next step is to start creating the calculations for the ending balance in column J. This is due to the fact that period 0 does not have a beginning balance. The purpose of having a period 0 is to track when the deal closed and funded. Period 0 has no beginning, it is merely a reference point to begin the deal and any values that correspond to period 0 are considered to be at the end of the period. As will be seen later, period 1 will begin where period 0 leaves off. When working with other models it is important to see whether calculations are based off of *end of period* (EOP) or *beginning of period* (BOP). Often times the difference can have a noticeable impact on results.
4. Since period 0 is an exception, an IF statement that inputs the initial asset balance is needed in column J. Otherwise the ending balance is just the beginning

balance minus the principal that has been paid in that period. In cell J6, enter the following formula and copy it down to cell J366:

= IF(A6=0,AssetCurBal1,E6–I6)

This formula reads: If the period is 0 then the value of cell J6 is the asset current balance off the Inputs sheet, otherwise it is the asset balance at the beginning of the period, minus the principal paid during that period.

5. The next calculation to complete is the beginning balance. This is extraordinarily easy because the beginning balance of any period is always the ending balance of the period prior. In cell E7 enter:

   = J6

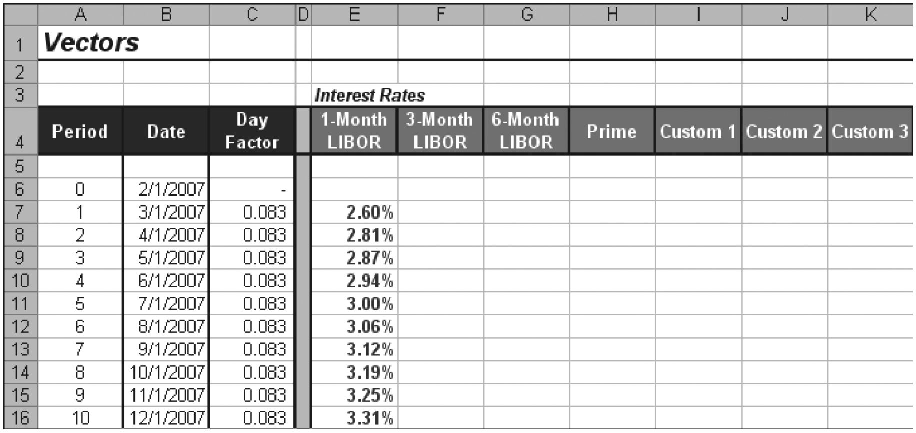
   Copy the cell down to row 366. If there is confusion to why this calculation and the next few begin on row 7 recall that period 0 starts with an ending balance only.
6. The next calculation is the periodic interest rate, which is the most intricate formula in this section and possibly the entire model. The numeric value of the interest rate is what needs to be returned in this section, and depends on many factors. The foremost factor is what interest rate vector the user selects on the Inputs sheet. Therefore the formula needs to reference the Inputs sheet to see the vector selection and the Vectors sheet to reference the projected periodic rate. Also, if the rate is a floating rate there is the possibility of caps, floors, and reset periodicity. All of these factors need to be taken into account to return an accurate interest rate for the period.
7. Before the rate formula is constructed, placeholder vectors should be entered on the Vectors sheet. The typical interest rate vector will be a projected curve of some type and can be obtained a number of ways. Usually a forward curve analysis from Bloomberg or any other financial information program is sufficient. To meet ratings criteria, a stressed curve constructed by the ratings agencies should be used. To be able to precisely follow the example calculations going forward, go to the completed Model Builder 2.2 example and in the Vectors sheet copy the curve for 1-month LIBOR (E6:E366). Paste that curve into the same location in the model in progress.
8. With placeholder rates stored, return to the Cash Flow sheet. Cell F7 is where the formula will be entered. The first part of the formula focuses on selecting a rate depending if the assets are paying fixed or floating. This calls for a simple IF statement referencing cell E12 of the Inputs sheet. If cell E12 is set to fixed, then the periodic rate will be the WA Fixed rate in cell F12 of the Inputs sheet, if not then it must be a floating rate and should be the interest rate from the Vectors sheet that corresponds to the correct period and curve selected. The fixed rate selection of the IF statement is simple:

   = IF(AssetIntType1= "Fixed",AssetFxdRate1,

9. Getting the false value of the formula, in this case the floating rate, requires referencing two pieces of information: the period and the curve selected. A glance at the Vectors sheet reveals that both of those pieces of information are there. All that needs to be made is a connection between the period that the cell corresponds to and the rate curve that the user selected on the Inputs sheet. This can be accomplished using a powerful lookup combination of the OFFSET and MATCH functions. Using these functions in tandem is better than using specific lookup functions such as VLOOKUP because the specific lookup functions typically require lookup values to be in order. OFFSET and MATCH have no such requirements. If you are unfamiliar with either of these functions, refer to the Toolbox section at the end of this chapter.

   To help understand how this method works, it is useful to think about the Vector sheet as a table with curve names on the x axis and the periods on the y axis. The OFFSET function is going to locate the correct cell on that table depending on the curve selected and the period. Since OFFSET is essentially a grid lookup tool that starts with a reference point and uses numerical values to find a cell, it starts to become clearer that there needs to be numerical values to represent the period and the curve. Given that the period is already in numeric format, the only difficult variable to conceive of as a number is the curve that is selected. However, since there is a list of the curves on the Vectors sheet (recall that the range E4:K4 was named lstInterestRates), there is an implied ordering to the list. Specifically in the named range 1-Month LIBOR (E4) comes first, 3-Month LIBOR (F4) comes second, etc. See Figure 2.9 for more detail.

   Looking at Figure 2.9 it makes sense then that a grid reference using D6 as a starting point, to 1-Month LIBOR, one period out (2.60 percent) should be (1,1). This method of grid referencing can be used within the context of the



	A	B	C	D	E	F	G	H	I	J	K
1	**Vectors**										
2											
3					***Interest Rates***						
4	Period	Date	Day Factor		1-Month LIBOR	3-Month LIBOR	6-Month LIBOR	Prime	Custom 1	Custom 2	Custom 3
5											
6	0	2/1/2007	-								
7	1	3/1/2007	0.083		2.60%						
8	2	4/1/2007	0.083		2.81%						
9	3	5/1/2007	0.083		2.87%						
10	4	6/1/2007	0.083		2.94%						
11	5	7/1/2007	0.083		3.00%						
12	6	8/1/2007	0.083		3.06%						
13	7	9/1/2007	0.083		3.12%						
14	8	10/1/2007	0.083		3.19%						
15	9	11/1/2007	0.083		3.25%						
16	10	12/1/2007	0.083		3.31%						

FIGURE 2.9 The Vectors sheet contains the list for the interest rate, which can be referenced to determine the correct periodic interest rate.

OFFSET function. If cell D6 was offset one cell to the right and one cell down, the active cell would be cell E6. This is synonymous with the formula:

= OFFSET($D$6,1,1)

That formula reads: offset cell D6 one cell to the right and one cell down. It would return the value in cell E7, 2.60 percent. Note that cell D6 is an absolute—or fixed—cell reference with $, so the reference never changes. Once it is clear that cell $D$6 is the focal point of the OFFSET, a function needs to be implemented to translate the text variable that selects the interest rate curve into a numerical value.

The function that accomplishes this for the interest rate curve variable is the MATCH function. The MATCH function works very well for lists with any value and any order. Conveniently, a list for the interest rate curve was created (lstInterestRates). If one were given the interest rate curve text variable of 1-Month LIBOR, it could be matched exactly against the list lstInterestRate. The MATCH function returns a numerical value for the placement of the matching value in the list. In this case, the label 1-Month LIBOR is the first value in lstInterestRate, and therefore a MATCH of the text 1-Month LIBOR returns a 1.

Since the periods are already in a numeric value, they do not need to be looked up using the MATCH function. If the offset originates in the cell above the first period, offsetting that cell one period down would require a numeric value of one. Notice that the period value for that corresponding cell is 1. Likewise to offset the cell two periods down, the corresponding period value is 2, and so on. If all of this seems confusing do not worry, the combined formula should start to make sense:

```
= IF(AssetIntType1= "Fixed",AssetFxdRate1,
   OFFSET(Vectors!$D$6,A7,MATCH(AssetFltIndx1,lstInterestRates,0))
```

This addition to the formula reads, offset cell D6 on the Vectors sheet by one cell down and by one cell to the right. Determining that the offset should go one cell down is done by the period value. Determining that the offset should go one cell to the right is done by matching the selected interest rate curve from the inputs sheet with the list of interest rates. Since 1-Month LIBOR is the first value on the list, the offset goes one cell to the right. Notice that if 3-Month LIBOR was selected on the Inputs sheet, this is second on the list and the offset would go two cells over to cell F7.

10. The next step in constructing the formula is to take care of instances when there is a cap. A cap is a hedge that is purchased to prevent the rate from going above a desired limit. With floating rate financing, there can be either a periodic cap that prevents the rate from going above a fixed amount each period or a lifetime cap that prevents the rate from going above a fixed amount over the lifetime of

the deal. The perfect function to take care of this is MIN. If a floating rate is being used the rate each period should be the minimum of the projected rate or the capped rate. The MIN statement (shown in bold) should be inserted into the formula as shown:

= IF(AssetIntType1= "Fixed",AssetFxdRate1,**MIN**(OFFSET
(Vectors!$D$6,A7, MATCH(AssetFltIndx1,lstInterestRates,0)),
(F6−AssetMarg1)+ **AssetPdCapFl1,AssetLifeCap1**)

The new section of the formula reads: return the minimum of the projected rate on the Vectors sheet or the previous rate plus the periodic cap or the lifetime cap. Notice that the margin is subtracted out of the previous month's rate. This is because at the end of the formula the margin will be added and any comparison for a cap must be of solely the projected rate. The previous month's rate already has the margin added. So, for comparison purposes, it must be subtracted out.

11. An opposite approach must be taken for floors. Floors are the opposite of caps and are hedges that prevent the rate from going below a certain amount. Given that the hedge is the opposite it makes sense that an opposite function should be used; the MAX function. The MAX function (shown in bold) should be added right before the MIN function. The periodic floor should be included, but the floor should be subtracted instead of added:

    = IF(AssetIntType1= "Fixed",AssetFxdRate1,**MAX**(MIN(OFFSET
    (Vectors!$D$6,A7,MATCH(AssetFltIndx1,lstInterestRates,0)),
    (F6−AssetMarg1)+AssetPdCapFl1, AssetLifeCap1),
    (F6−AssetMarg1)**−AssetPdCapFl1,AssetLifeFloor1**)

12. The next part of the formula must figure out if the floating rate should be changed depending on the reset frequency. Some floating rates only change quarterly or semiannually. The Inputs sheet anticipated this with cell M12 (AssetRateReset). The ideal function for determining when to do something according to timing is the MOD function. If unfamiliar with MOD, consult the Toolbox section for detail. Essentially MOD (shown in bold) is used to see if the current period is perfectly divisible with the rate reset frequency established on the Inputs sheet.[1] If it is then the rate needs to be looked up, otherwise the rate remains the same as last month. The wording of the last sentence should be a sign that an IF statement should also be used:

    = IF(AssetIntType1= "Fixed",AssetFxdRate1,**IF(MOD($A7,**
    AssetRateReset1)=**0,**MAX(MIN(OFFSET(Vectors!$D$6,A7,

[1] Public Securities Association, *Standard Formulas for the Analysis of Mortgage-Backed Securities and Other Related Securities*, 1 June 1990, p. SF-5.

MATCH(AssetFltIndx1,lstInterestRates,0)),(F6−AssetMarg1)
+AssetPdCapFl1,AssetLifeCap1),(F6−AssetMarg1)−AssetPdCapFl1,
AssetLifeFloor1),**F6**)

This addition to the formula reads, if the current period divides by the given reset frequency perfectly then look up the rate on the Vectors sheet, otherwise use the previous month's rate (F6).

13. The last section is primarily clean up to take into account the first period not having a previous month's rate. If the first period is a floating rate, it should simply reference the starting floating rate. This can easily be done with an IF statement referencing the current period and the TRUE value being the same rate lookup method we used earlier (i.e., the OFFSET MATCH combination). Insert the IF statement (shown in bold) and copy and paste the OFFSET MATCH combination (shown in bold) to produce the following:

= IF(AssetIntType1= "Fixed",AssetFxdRate1,**IF(A7=1,OFFSET(Vectors!$D$6,**
A7,MATCH(AssetFltIndx1,lstInterestRates,0))),IF(MOD($A7,
AssetRateReset1)=0,MAX(MIN(OFFSET(Vectors!$D$6,A7,
MATCH(AssetFltIndx1,lstInterestRates,0)),(F6−AssetMarg1)
+AssetPdCapFl1, AssetLifeCap1),(F6−AssetMarg1)
−AssetPdCapFl1,AssetLifeFloor1),F6)))

14. The final bit to add is the margin. No matter what rate is returned, fixed or floating, the margin needs to be added at the end. The only caveat is that the lifetime floor and cap can be exceeded by the margin. To take this into account the margin (shown in bold) needs to be subtracted from the lifetime cap and floor. The final formula should look as follows:

= IF(AssetIntType1= "Fixed",AssetFxdRate1,IF(A7=1,OFFSET(Vectors!$D$6,
A7,MATCH(AssetFltIndx1,lstInterestRates,0)),IF(MOD($A7,
AssetRateReset1)=0,MAX(MIN(OFFSET(Vectors!$D$6,A7,
MATCH(AssetFltIndx1,lstInterestRates,0)),(F6−AssetMarg1)
+AssetPdCapFl1, AssetLifeCap1−AssetMarg1),(F6−**AssetMarg1**)
−AssetPdCapFl1,AssetLifeFloor1−**AssetMarg1**), F6)))
+**AssetMarg1**

15. The next formula calculates the total payment that is due each period. The PMT function in Excel is specifically designed for this purpose. The PMT function

uses balance, term, and rate information to provide a periodic payment. In Project Model Builder, the payment is level in fixed rate scenarios and variable when the rate is floating. This is accomplished by making sure the payment formula references the rate vector, which will be the same rate for each period in a fixed rate system and a differing rate each period for a floating rate system.

Another consideration for the payment formula is that in a floating rate system the payment can be higher than the balance of the assets. To take care of this, an IF statement needs to be inserted to check whether the balance is less than the payment and, if so, calculate the payment as the sum of the balance and the interest due that period. In cell G7 enter the following formula and copy it down to cell G366:

= IF(J6<=G6,H7+E7,−PMT($F$7*$C$7,AssetTerm1,AssetOrgBal1))

One final note on the formula is that there is a negative sign in front of the PMT function. This is because PMT will return a payment as a negative value and for consistency all numbers in the model will be positive.

16. Once the payment formula is done, the next formula to create is the interest calculation. This is very simple since the periodic interest rate is available in column F. The interest per period is the beginning balance multiplied by the interest rate. Remember that the periodic interest rate is still an annual rate and must be calculated as a periodic rate by multiplying by the day factor. The final formula to enter in cell H7 (and copy down to cell H366) is:

    = F7*C7*E7

17. The last column in this section is the principal payment. Given that the total payment and the current period's interest have already been calculated, the principal for that period is the subtraction of the interest from the total payment. In cell I7 enter and copy down to cell I366 the following formula:

    = G7−H7

18. The notional amortization schedule is now complete. Try changing the variables on the Inputs sheet to see how the amortization schedule is affected. Make sure this section is well understood since the cash that will flow through the rest of the model is completely dependent on the notional amortization schedule.

TOOLBOX

OFFSET

Perhaps one of the most useful types of functions in financial modeling is one that can reference other cells. The OFFSET function takes a selected cell and references a different cell depending on the parameters given. The OFFSET function needs the following information:

= OFFSET(starting point cell, number of rows to move up or down,
number of columns to move back or forth)

For example, if the following cells and values were used with the function

= OFFSET(A1,1,1)

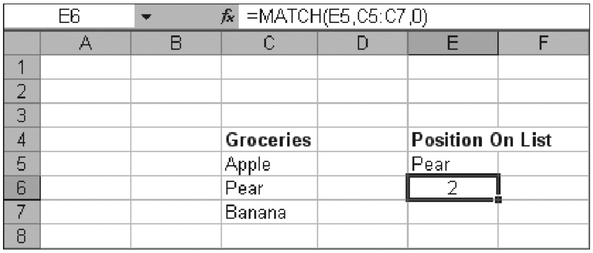
the return value would be the value in cell B2 since the OFFSET function started with cell A1 and offset the cell by 1 row and 1 column.

MATCH

The MATCH function is often overlooked, but it is an incredibly powerful function. The concept is to compare a given value against an array or list of values that may contain the given value. If there is a match, the function returns a numerical value that represents the order that the matched value is in the array or list. MATCH takes the following parameters:

= MATCH(selected cell for matching, array to be matched against,
accuracy of search)

In Figure 2.10, there is a list of groceries in range C5:C7. If the location of the word "Pear" on the list was required the MATCH function could be used to find it. The last parameter, 0 in this example, is how accurate the search should be. A 0 indicates that the word "Pear" needs to be found exactly, if not a #N/A error will be returned. The other two options, 1 and -1, are mainly for matches involving numbers. A 1 indicates that MATCH should find the largest value that is less than or equal to the given value. A -1 indicates that MATCH should find the smallest



E6 =MATCH(E5,C5:C7,0)

	A	B	C	D	E	F
1						
2						
3						
4			**Groceries**		**Position On List**	
5			Apple		Pear	
6			Pear		2	
7			Banana			
8						

FIGURE 2.10 An example of MATCH using a grocery list and an item from it.

value that is greater than or equal to the lookup value. Using 1 or −1 requires the array or list to be placed in order, so try to use 0 if possible.

MOD

The MOD function is a mathematical function that returns the remainder after a number has been divided by a divisor. There are only two parameters for MOD, the number that is to be divided and the divisor.

PMT

A useful function for finance is the PMT function. This function takes a rate, a term, and a principal amount and returns the payment necessary to amortize the principal amount within the term given. The specific parameters include:

= PMT(rate, term, principal balance)

Always be certain that the rate corresponds to the term. For example, if the term is on a monthly basis the rate needs to be adjusted so that it is a monthly rate. Also, the PMT function's default setting is to return the payment as a negative value, so either a negative needs to be put in front of the formula or other formulas have to work with a negative payment value. Project Model Builder assumes all positive values and requires the negative to be in front of the function.

CHAPTER 3

Prepayments

In Chapter 2, a notional amortization schedule was created that provided a basis for cash flowing into a transaction. This cash flow assumes that every loan in the pool will make payments exactly as they are scheduled. If it were that simple many finance professionals would be out of a job. In reality, cash flow becomes irregular when obligors choose to pay more than the scheduled amount, a prepayment, or when they do not pay at all, a delinquency, which sometimes becomes a default.

This chapter focuses on prepayments, how they are tracked, how they are projected, and how they are used in a cash flow model. First, it should be clear exactly what is considered a prepayment. A prepayment can either be "complete," where the outstanding balance is paid off in full, or "partial," where only a portion of the outstanding balance is paid. For example, an individual mortgage obligor might have extra money one month and decide to pay $2,000 towards her mortgage when the scheduled payment is $1,800. Since the periodic principal and interest is calculated in the $1,800 due, the only use of the additional $200 is to reduce the outstanding balance of the loan. This would be considered a partial prepayment. The other type of prepayment is where the entire balance is paid off. This can be the result of refinancing, credit-related events, and, on occasion, calculated due to foreclosure.

The latter part of this chapter features Model Builder exercises; but some of these exercises are not directly involved in building the example model. One exercise demonstrates how to take a file of raw historical prepayment information and create a projected prepayment curve. This projected prepayment curve then is used for prepayments in Project Model Builder. The purpose of this is to show a reader three important parts of structured finance modeling:

1. The ideal format for prepayment data.
2. How to convert the prepayment data into projections.
3. How to apply and integrate the projections in a working financial model.

HOW PREPAYMENTS ARE TRACKED

While the concept of a prepayment is relatively simple, there are numerous methods of tracking prepayments and calculating prepayment rates. Prepayment terminology cannot only differ between asset classes, but even between companies within the same industry. It is important to understand what each prepayment expression means and the underlying calculation involved so historical data can be interpreted and used to make projections. Prior to exploring how to work with prepayment data, it would be useful to define some of the primary terms.

SMM: Single Monthly Mortality

The most basic measurement of prepayment is known as *single monthly mortality* (SMM). The name sounds a bit morbid, but it comes from the idea that the prepayment is actually getting rid of or killing off the asset. In most modeling, whether it is a mortgage or auto model, the prepayment rate is typically converted to SMM for calculating prepayments. This is done because most models are done on a monthly basis and as the name implies SMM is a monthly calculation. It is calculated by taking the dollar amount of the prepayment over the previous month's outstanding balance (less the scheduled principal payment). The formula for SMM is:

$$SMM = \text{Dollar amount of period } n \text{ prepayment/(Beginning of period } n \text{ current balance} - \text{Period } n \text{ scheduled principal payment)}$$

CPR: Conditional Prepayment Rate

Since most transactions are monthly, SMM is appropriate for calculations. However, for mortgage models and other long-term assets the prepayment rate is often expressed as an annual rate known as the *conditional prepayment rate* (CPR). CPR is an often misunderstood term with incorrect acronym explanations such as Cumulative Prepayment Rate or confusion between CDR, which is a default term that will be described in Chapter 4. The most basic definition of CPR is that it is an annualized calculation of SMM. To convert SMM to CPR use the following formula:

$$CPR = 1 - (1 - SMM)^{12}$$

This calculation can be set up in a spreadsheet or is referenced in numerous books in a table format (i.e., SMM to CPR conversion tables).

PSA: Public Securities Association

With many years of historical prepayment data on mortgages, standardized prepayment amounts have been developed by the *Public Securities Association* (PSA). These curves or multiples of these curves are often used to model mortgage transactions. One-hundred percent PSA assumes prepayment rates of 0.2 percent CPR in the first month following origination of the mortgage loans (not the pool) and an additional 0.2 percent CPR in each succeeding month until the 30th month. In the

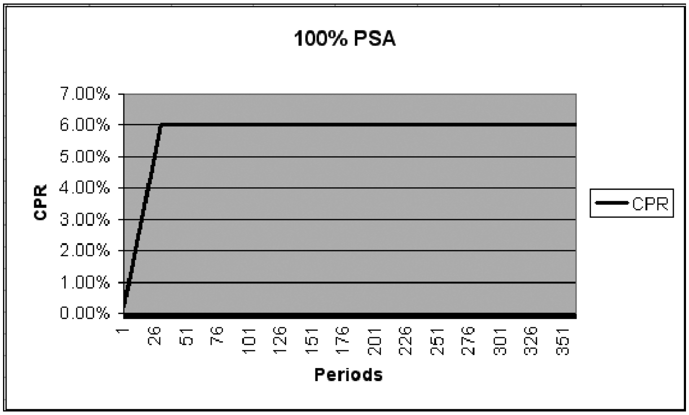


FIGURE 3.1 100 percent PSA expressed as a line graph.

30th month and beyond, 100 percent PSA assumes a fixed annual prepayment rate of 6.0 percent CPR. See Figure 3.1 for a graph of the standard PSA curve.

To calculate the prepayment rate for any specific multiple of PSA, adjust the annual rate at 100 percent PSA by that multiple. For example, 200 percent PSA assumes prepayment rates equal to twice the CPRs from the 100 percent PSA model, on a pool-by-pool basis.[1] Specifically, 0.2 percent CPR would instead be 0.4 percent CPR for the first month, following the origination of the mortgage loans, and grow by 0.4 percent CPR for each of the successive 29 months and 12.0 percent CPR for all of the remaining months.

ABS: Absolute Prepayment Speed

For shorter term assets such as auto loans, a more appropriate measure of prepayment is the *absolute prepayment speed* (ABS). Due to the short terms of the asset, larger amounts of principal are paid faster causing an increasing prepayment rate as the pool seasons. To use ABS in a model it should be converted to SMM. This is done by the following formula:

$$ABS = 100^{*}SMM/(100 + SMM^{*}(n - 1))$$

In this formula n is the number of periods that have transpired since origination. For instance, if a pool is known to have an ABS of 1.50 percent, the corresponding SMM to use in a calculation five months out would be 1.60 percent.

[1] Public Securities Association, *Standard Formulas for the Analysis of Mortgage-Backed Securities and Other Related Securities*, 6/01/90, p. SF-5.

HISTORICAL PREPAYMENT DATA FORMATS

Projecting prepayments can get very complex with interest rate and economic expectations. The most basic starting point is to analyze how the assets have prepaid in the past. Obtaining sufficient historical prepayment data (at least three years worth) on the specific assets that are to be analyzed is the best approach to take. Depending on the company, this data will come in a variety of formats, but must include at minimum: a historical account of the asset's principal balance each period and the amount of principal that was prepaid each period. Ideally, though, the company provides a vintage analysis of the prepayments.

As the name implies, a *vintage analysis* tracks information by the month, quarter, or year that the assets were originated. For example, assets originated in August 1998 would have an original pool balance from that month. From August to September, there are prepayments for just the loans originated in August 1998, which should be tracked separately from the entire book of assets. Similarly, the current principal balance of the August 1998 vintage should be tracked separately. This type of tracking should be done for every month of origination because it allows for a proper trend analysis.

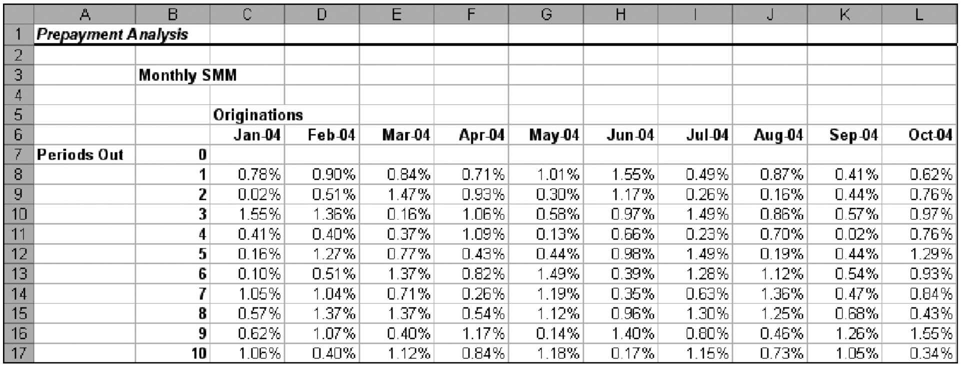
It should be noted that the data presented in the Model Builder section is an ideal example of how prepayment data is tracked. Often times this data is part of a larger analysis and may require manipulation to bring into a usable format.

BUILDING PREPAYMENT CURVES

With suitable prepayment data the next step is to build prepayment curves that represent each vintage and an aggregated curve that represents the assets as a whole. The first point to remember is that prepayment amounts used in Project Model Builder will be calculated off of SMM, which is the prepayment per period over the current balance of the period. If this is not done, then the rates will have to be calculated. Once they are calculated there should be a table of rates that have origination periods on one axis and periods on the other axis. This becomes clearer as the Model Builder exercise for this chapter is worked. An example for you to reference now is shown in Figure 3.2.

The table of monthly SMM rates can then be aggregated by each period using a weighted average methodology. To account for differences in balances, each SMM should be weighted by the periodic principal balance from the vintage that the SMM is derived from. The weighted average aggregate curve can then be summed to create a cumulative curve. Be aware that this is the most fundamental method for creating prepayment curves and more advanced modeling would include an array of different factors and sophisticated techniques.

As an introduction the weighted average curve is the most basic representation of the assets prepayment behavior. Using this curve to project prepayments assumes the following:



	A	B	C	D	E	F	G	H	I	J	K	L
1	***Prepayment Analysis***											
2												
3		**Monthly SMM**										
4												
5			**Originations**									
6			**Jan-04**	**Feb-04**	**Mar-04**	**Apr-04**	**May-04**	**Jun-04**	**Jul-04**	**Aug-04**	**Sep-04**	**Oct-04**
7	**Periods Out**	**0**										
8		**1**	0.78%	0.90%	0.84%	0.71%	1.01%	1.55%	0.49%	0.87%	0.41%	0.62%
9		**2**	0.02%	0.51%	1.47%	0.93%	0.30%	1.17%	0.26%	0.16%	0.44%	0.76%
10		**3**	1.55%	1.36%	0.16%	1.06%	0.58%	0.97%	1.49%	0.86%	0.57%	0.97%
11		**4**	0.41%	0.40%	0.37%	1.09%	0.13%	0.66%	0.23%	0.70%	0.02%	0.76%
12		**5**	0.16%	1.27%	0.77%	0.43%	0.44%	0.98%	1.49%	0.19%	0.44%	1.29%
13		**6**	0.10%	0.51%	1.37%	0.82%	1.49%	0.39%	1.28%	1.12%	0.54%	0.93%
14		**7**	1.05%	1.04%	0.71%	0.26%	1.19%	0.35%	0.63%	1.36%	0.47%	0.84%
15		**8**	0.57%	1.37%	1.37%	0.54%	1.12%	0.96%	1.30%	1.25%	0.68%	0.43%
16		**9**	0.62%	1.07%	0.40%	1.17%	0.14%	1.40%	0.80%	0.46%	1.26%	1.55%
17		**10**	1.06%	0.40%	1.12%	0.84%	1.18%	0.17%	1.15%	0.73%	1.05%	0.34%

FIGURE 3.2 A standard analysis of prepayment rates is completed by tracking each origination period separately over time.

1. That the assets requiring prepayment projections are the same as the assets used to create the projected prepayment curve. This might sound like simple logic, but if the asset class is for example mortgages and the historical prepayment data is generated off of fixed rate mortgages, the prepayment curve calculated should only be used for fixed rate mortgages from that company. This distinction should even be as granular as the precise type of mortgage, since prepayment rates for mortgages are very particular to the type of product.
2. That there is no significant trend in the prepayment rates. If a noticeable increasing or decreasing trend is apparent in the historical prepayment rates and this trend is expected to maintain, it should be accounted for in the projected curve. Sometimes a shift in market focus, origination process, or economic environment causes such a trend, which could affect future prepayment performance.
3. While more complex than this text will demonstrate in the examples, a consideration for any prepayment curve should be interest rates. Prepayment behavior is closely linked to interest rates. A simple example is a homeowner with a mortgage, which could be fixed or floating. If the mortgage is fixed rate and it was originated during a high interest rate environment, the homeowner will be more inclined to refinance when interest rates decline. Or, if a floating rate mortgage was originated during a low interest rate environment, the obligor would want to refinance to a fixed rate if he or she expects interest rates to increase. In more advanced prepayment analyses, expected interest rate scenarios should be run using a variety of prepayment curves.

PREPAYMENT CURVES IN PROJECT MODEL BUILDER

Once the projected curve is created it will be stored in the model. For maximum flexibility the model should have room for multiple custom curves, conversion formulas for the main types of prepayment rates (as described above), and standard prepayment curves. Also, the model should be developed in a way that it is easy to select which curve to use, the option to use different curves for specific groups of assets, and a method of automating scenarios so curves can be selected and run without much adjustment. All of this will be demonstrated and worked through in the Model Builder exercises.

THE EFFECT OF PREPAYMENTS ON STRUCTURED TRANSACTIONS

Prior to the step-by-step exercises in the Model Builder sections a moment should be taken to understand the effects that prepayments have in a structured finance transaction. Prepayments accelerate the repayment of principal for the assets and, in pass-through structures, the principal of the notes or certificates. At first thought an investor might think prepayments are good since they will have their principal returned to them faster.

This is true if the assets are consistently performing poorly and there is risk that there will not be enough cash flow to repay the investor. In such a case, one would want their money as fast as possible. However, when assets are performing as they are expected, there should be more asset interest than *liability cost*—also known as *excess spread*. This excess spread provides credit protection over the life of a transaction. Obligors who tend to prepay the fastest are the better credits with high interest rates that, when, eliminated, reduce excess spread. If those higher interest obligors prepay quickly, the transaction terminates faster and overall there will be less excess spread in the deal than if the assets prepaid at a slower rate. This is extremely important because if a comparison were made between two transactions—A and B—where everything was the same including default rate but the only difference was transaction A prepaid faster than transaction B, then transaction A would have less excess spread.

Another consideration is that the prepayment rate needs to be modeled accurately to give a correct estimation of the weighted average life of the transaction and the total return provided to investors. Institutional investors often purchase asset-backed securities with certain tenors and returns in mind. Prepayments can drastically affect the weighted average life of a transaction and can reduce the overall interest paid, thereby reducing the total return of investment.

MODEL BUILDER 3.1: HISTORICAL PREPAYMENT ANALYSIS AND CREATING A PROJECTED PREPAYMENT CURVE

1. The first step in any prepayment analysis is to review the data that exists. Open *MB3-1_Raw_Data.xls* from the Ch03 folder on the CD-ROM. There is only one sheet in this workbook and it contains very pertinent and organized data for

a prepayment analysis. The data included on the sheet is the minimum amount necessary to complete a prepayment analysis: pool balances and prepayment amounts.

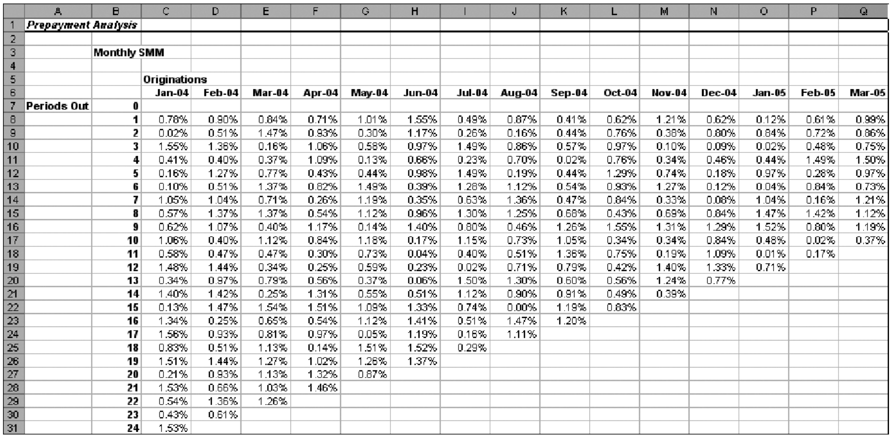
2. To get started with the analysis, create another sheet and name it **Prepay Analysis**. This sheet is where the SMM prepayment rates will be calculated and aggregated by pool. Since the data will be organized similar to the raw data, copy cells A5:AA31 from the Raw Data sheet and paste them on A5 of the Prepay Analysis sheet.
3. Clear the contents of cells A7:AA7 because there will be no prepay rates in time 0. The formula to calculate the SMM rates is the prepay amount for the current period divided by the beginning of period balance. In the official formula, scheduled principal should be deducted from the beginning of period balance; however, in this case there is no information on how the balance is reduced each month (that is, what portion is scheduled principal versus default). The difference should not have a significant impact so using the beginning of period balance is sufficient. The Excel formula to enter in cell C8 is:

   ```
   ='Raw Data'!C39/'Raw Data'!C7
   ```

4. When this is copied from C8:AA31, there is a minor problem with #DIV/0 errors. This can be easily taken care of with an IF statement. Modify cell C8 as follows:

   ```
   =IF('Raw Data'!C39="","",'Raw Data'!C39/'Raw Data'!C7)
   ```

   This will prevent any 0 balances from trying to calculate and stop division by zero errors from populating across the sheet. At this point, the analysis should look like an upside down triangle as pictured in Figure 3.3.
5. With each vintage's prepayment rates laid out over time, the next step is to create an aggregate curve that represents how the asset's prepay on average. Since the vintages have different balances a weighted average should be used for the rates. The easiest way to calculate a weighted average in Excel is using a SUMPRODUCT-SUM combination. If this has never been done before, there is a detailed explanation in the Toolbox section of this chapter. There is a slight difference when using this combination for curves because there could be zero values that could skew the average. To eliminate this potential problem, a count of relevant cells needs to be created.
6. Label cell AC6 **WA Count**. The cells in AC8 through AC31 will contain a numeric value that represents the relevant number of cells of data for each period that should be calculated in the weighted average. For example, row 8 contains data for the first period of each vintage, row 9 the data for the second vintage, and so on. Notice though that the data is a triangle and the further the periods out, the less data exists. This is logical because if the current month is the beginning of January 2006 there should be no data for January 2006 one month out nor two months of data for loans originated in December 2005. To



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	***Prepayment Analysis***																
2																	
3		**Monthly SMM**															
4																	
5			**Originations**														
6			**Jan-04**	**Feb-04**	**Mar-04**	**Apr-04**	**May-04**	**Jun-04**	**Jul-04**	**Aug-04**	**Sep-04**	**Oct-04**	**Nov-04**	**Dec-04**	**Jan-05**	**Feb-05**	**Mar-05**
7	**Periods Out**	**0**															
8		**1**	0.78%	0.90%	0.84%	0.71%	1.01%	1.55%	0.49%	0.87%	0.41%	0.62%	1.21%	0.62%	0.12%	0.61%	0.99%
9		**2**	0.02%	0.51%	1.47%	0.93%	0.30%	1.17%	0.26%	0.16%	0.44%	0.76%	0.38%	0.80%	0.84%	0.72%	0.86%
10		**3**	1.55%	1.36%	0.16%	1.06%	0.58%	0.97%	1.49%	0.86%	0.57%	0.97%	0.10%	0.09%	0.02%	0.48%	0.75%
11		**4**	0.41%	0.40%	0.37%	1.09%	0.13%	0.66%	0.23%	0.70%	0.02%	0.76%	0.34%	0.46%	0.44%	1.49%	1.50%
12		**5**	0.16%	1.27%	0.77%	0.43%	0.44%	0.98%	1.49%	0.19%	0.44%	1.29%	0.74%	0.18%	0.97%	0.28%	0.97%
13		**6**	0.10%	0.51%	1.37%	0.82%	1.49%	0.39%	1.28%	1.12%	0.54%	0.93%	1.27%	0.12%	0.04%	0.84%	0.73%
14		**7**	1.05%	1.04%	0.71%	0.26%	1.19%	0.35%	0.63%	1.36%	0.47%	0.84%	0.33%	0.08%	1.04%	0.16%	1.21%
15		**8**	0.57%	1.37%	1.37%	0.54%	1.12%	0.96%	1.30%	1.25%	0.68%	0.43%	0.69%	0.84%	1.47%	1.42%	1.12%
16		**9**	0.62%	1.07%	0.40%	1.17%	0.14%	1.40%	0.80%	0.46%	1.26%	1.55%	1.31%	1.29%	1.52%	0.80%	1.19%
17		**10**	1.06%	0.40%	1.12%	0.84%	1.18%	0.17%	1.15%	0.73%	1.05%	0.34%	0.34%	0.84%	0.48%	0.02%	0.37%
18		**11**	0.58%	0.47%	0.47%	0.30%	0.73%	0.04%	0.40%	0.51%	1.36%	0.75%	0.19%	1.09%	0.01%	0.17%	
19		**12**	1.48%	1.44%	0.34%	0.25%	0.59%	0.23%	0.02%	0.71%	0.79%	0.42%	1.40%	1.33%	0.71%		
20		**13**	0.34%	0.97%	0.79%	0.56%	0.37%	0.06%	1.50%	1.30%	0.60%	0.56%	1.24%	0.77%			
21		**14**	1.40%	1.42%	0.25%	1.31%	0.55%	0.51%	1.12%	0.90%	0.91%	0.49%	0.39%				
22		**15**	0.13%	1.47%	1.54%	1.51%	1.09%	1.33%	0.74%	0.00%	1.19%	0.83%					
23		**16**	1.34%	0.25%	0.65%	0.54%	1.12%	1.41%	0.51%	1.47%	1.20%						
24		**17**	1.56%	0.93%	0.81%	0.97%	0.05%	1.19%	0.16%	1.11%							
25		**18**	0.83%	0.51%	1.13%	0.14%	1.51%	1.52%	0.29%								
26		**19**	1.51%	1.44%	1.27%	1.02%	1.26%	1.37%									
27		**20**	0.21%	0.93%	1.13%	1.32%	0.87%										
28		**21**	1.53%	0.66%	1.03%	1.46%											
29		**22**	0.54%	1.36%	1.26%												
30		**23**	0.43%	0.61%													
31		**24**	1.53%														

FIGURE 3.3 The raw prepayment data should be converted into percents and cleaned up at this point.

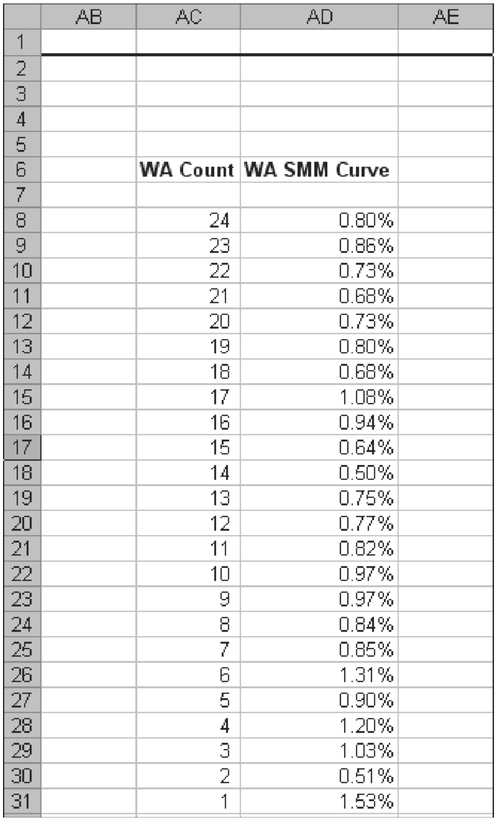
help with the SUMPRODUCT-SUM combination a numeric count of the data that should be included in the calculation is necessary. This can be accomplished by using the COUNT formula in conjunction with data specific modifications.

7. In cell AC7, use COUNT on the cells C6 through AA6. This produces a value of 25. Lock this reference down. For the first period in cell AC7, there is one period of unknown data: January 2006. In the next period there will be two periods of unknown data: January 2006 and December 2005. To use a single formula to not count those months, subtract the current period from the count. Since the numeric periods exist in column B, use those as the reference. The final formula should look like:

   =COUNT($C$6:$AA$6)−B8

   Copy and paste this formula into cells AC8 to AC31. The result should be a vertical row of numbers that decrease as the periods decrease as shown in Figure 3.4.
8. Next label AD6 **WA SMM Curve**. This is where the weighted averages (WA) are calculated. Any weighted average of A data is the sum of the products of A data and B data divided by the sum of B data. This can be accomplished in Excel using SUMPRODUCT and SUM. An additional element of complexity is making sure not to average blank cells. This is done using the OFFSET function within the other formulas. In cell AD8 enter the following formula:

   = SUMPRODUCT(C8:OFFSET($B8,0,$AC8),'Raw Data'!C7:OFFSET
   ('Raw Data'!B7,0,$AC8))/SUM('Raw Data'!C7:OFFSET('Raw
   Data'!B7,0,$AC8))



	AB	AC	AD	AE
1				
2				
3				
4				
5				
6		WA Count	WA SMM Curve	
7				
8		24	0.80%	
9		23	0.86%	
10		22	0.73%	
11		21	0.68%	
12		20	0.73%	
13		19	0.80%	
14		18	0.68%	
15		17	1.08%	
16		16	0.94%	
17		15	0.64%	
18		14	0.50%	
19		13	0.75%	
20		12	0.77%	
21		11	0.82%	
22		10	0.97%	
23		9	0.97%	
24		8	0.84%	
25		7	0.85%	
26		6	1.31%	
27		5	0.90%	
28		4	1.20%	
29		3	1.03%	
30		2	0.51%	
31		1	1.53%	

FIGURE 3.4 Notice the count function and the WA SMM Curve.

The SUMPRODUCT is taking two rows of data: the monthly SMMs (row 8 of the Prepay Analysis sheet) and the beginning of period balances for each vintage (row 7 of the Raw Data sheet). Instead of taking the entire row of data though, the SUMPRODUCT reference uses an OFFSET function to instruct the SUMPRODUCT to only take a certain amount of cells. This OFFSET uses the count system of relevant data created in column AC. This way the SUMPRODUCT only references the relevant data. Similarly, the SUM of the balances that is used as the divisor only takes in the relevant balances. If this OFFSET didn't exist the SUM would be much higher than necessary. Copy and paste the formula into cells AD8:AD31.

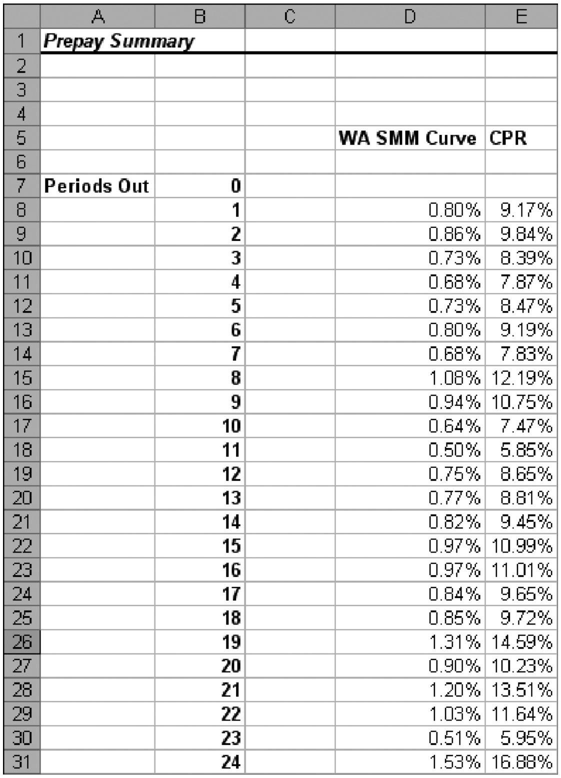
9. Create an additional sheet and name it **Summary**. This is where the curves that will be used for the projected modeling will be stored. Copy cells A7:B31 from the Prepay Analysis sheet and paste them in A7 of the Summary sheet. On the Summary sheet, label D5 **WA SMM Curve**. In D8 through D31, reference the

WA SMM curve from the Prepay Analysis sheet. Typically this is what will be used in a model to forecast prepayments. It has been simplified with only two years of data to keep the example simple, but in models like Project Model Builder a longer curve or estimation is typically necessary. This can be achieved by using more data, using a standard curve like PSA, or using rating agency assumptions that are often given in terms of CPR.

10. To see how SMM translates into CPR and vice versa, label cell E5 **CPR**. In cell E8, use the following formula to convert to CPR:

$$=1-(1-\text{D8})\hat{\ }12$$

Copy and paste this formula over the range E8:E31. These are the annualized rates of the SMMs. Rating agencies often give CPR assumptions for assets, so it is important to understand how to go back and forth between calculations. The final prepayment curve should look like Figure 3.5.



	A	B	C	D	E
1	***Prepay Summary***				
2					
3					
4					
5				**WA SMM Curve**	**CPR**
6					
7	**Periods Out**	**0**			
8		**1**		0.80%	9.17%
9		**2**		0.86%	9.84%
10		**3**		0.73%	8.39%
11		**4**		0.68%	7.87%
12		**5**		0.73%	8.47%
13		**6**		0.80%	9.19%
14		**7**		0.68%	7.83%
15		**8**		1.08%	12.19%
16		**9**		0.94%	10.75%
17		**10**		0.64%	7.47%
18		**11**		0.50%	5.85%
19		**12**		0.75%	8.65%
20		**13**		0.77%	8.81%
21		**14**		0.82%	9.45%
22		**15**		0.97%	10.99%
23		**16**		0.97%	11.01%
24		**17**		0.84%	9.65%
25		**18**		0.85%	9.72%
26		**19**		1.31%	14.59%
27		**20**		0.90%	10.23%
28		**21**		1.20%	13.51%
29		**22**		1.03%	11.64%
30		**23**		0.51%	5.95%
31		**24**		1.53%	16.88%

FIGURE 3.5 The final prepayment curve is expressed in SMM and CPR.

MODEL BUILDER 3.2: INTEGRATING PROJECTED PREPAYMENTS IN ASSET AMORTIZATION

1. The last part that was completed in Project Model Builder was the creation of a notional amortization schedule. This Model Builder section will begin creating the actual amortization schedule beginning with prepayments. Go to the Cash Flow sheet in Project Model Builder and label cell L3 **Actual Amortization**. Below this, in row 4, is a series of column headers. Label the following cells as described:
   L4: **Beginning Balance**
   M4: **Default Rate**
   N4: **New Defaults**
   O4: **Amort Factor**
   P4: **Prepay Rate**
   Q4: **Voluntary Prepay**
   R4: **Actual Amort**
   S4: **Interest Rate**
   T4: **Actual Interest**
   U4: **Principal Recovery**
   V4: **Ending Balance**
   Many of these columns are placeholders for chapters to come; but they should be created now rather than inserting columns later.
2. Two of these columns contain formulas in period 0 (row 6): the Amort Factor and the Ending Balance, which as mentioned earlier is the starting balance when it is in time period 0. The Amort Factor is the ratio of the current notional balance to the original notional balance. This is important because it is an indication of how the asset balance is reduced on a scheduled basis and will be used to help determine prepay amounts. In period 0, however, it is always 1 since no amortization has taken place. Otherwise it is the end of period scheduled balance over the original scheduled balance. This is the formula that should be entered in O6 and copied and pasted through O366:

   =IF(A6=0,1,J6/$J$6)

   The Ending Balance will always be the starting asset current balance for period 0, but will reduce by scheduled amortization, prepayments, and defaults. Even though the cells are empty right now use the following references in the formula for cell V6:

   =IF(A6=0,AssetCurBal1,L6−N6−Q6−R6)

   Copy and paste this formula into the range V6:V366.
3. The next column to work on is column L, the Beginning Balance. As in the notional schedule, the beginning balance will always be the ending balance from

the period prior. In cell L7, enter the reference **= V6** and copy and paste it down through the range L6:L366.

4. Before proceeding to the Prepay Rate, the Inputs sheet needs to be updated so Prepayment Inputs can be changed quickly. On the Inputs sheet, merge cells B16:O16 and use the long cell as a label that states **PREPAYMENT/DEFAULTS/RECOVERY INPUTS**. Also add the following labels:
   B17: **Description**
   C17: **Prepay Curve**
   D17: **Prepay Stress**
5. The Description is a reference to the description of the assets that the prepayment, default, and recovery information will affect. This is Asset Pool 1, which was named **AssetDes1**. In B18, enter **=AssetDes1.** Name this cell **pdrDes1**.
6. C18 is where the prepayment curve will be selected from the Vectors sheet. This is very similar to the asset interest rate curve selection. Like that one there needs to be a selection of possible curves on the Vectors sheet. Go to the Vectors sheet and enter the following labels into these cells:
   M5: **SMM 1**
   N5: **SMM 2**
   O5: **SMM 3**
   P5: **Custom CPR 1 DATA ENTRY ONLY**
   Q5: **Custom CPR 2 DATA ENTRY ONLY**
   R5: **Custom CPR 3 DATA ENTRY ONLY**
   Select and name the range M5:O5 **lstPrepayCurve**.
7. Go back to the Inputs sheet and select C18. Using data validation, make the possible inputs for this cell the range lstPrepayCurve. There should only be three possible inputs because the range only included the SMM ranges. The CPR columns created on the vector sheet, which are labeled DATA ENTRY ONLY, are used to put CPR curves in, but must be converted to SMM for use in the model. Name C18 **pdrPrepay1**.
8. D18 contains the stressor for prepayments. This is a numeric multiple that is multiplied against the prepayment curve at each period. For now enter a **1** in cell D18 and name the cell **pdrPrepayStress1**.
9. To follow in the example calculations, copy and paste the existing prepayment curve, which is present in the completed section of the Vectors sheet in this *MB3-2.xls* in the Ch03 folder on the CD-ROM. This curve should be pasted in the same place, cells M7:M366. (In the Ch03 folder on the CD-ROM, there is an example in the Additional Files subfolder that contains the PSA curve calculation.)
10. The next step is having the correct prepayment rate show up on the Cash Flow sheet. This is accomplished in a similar manner to the asset interest rate, using an OFFSET-MATCH combination, but is much simpler since there are no caps, floors, or rate resets. The numerical value that should show up in P7 depends on the prepay curve selected on the Inputs sheet and the period (in this

case 1). Since the values are stored on the Vectors sheet, the OFFSET will start there with Vectors L5. To get the correct rate Vectors L5 needs to be offset by the period that the formula is in (1 in the case of Cash Flow P7) and the prepay curve (the selected one in this example is SMM 1, which is the first curve in the list). An additional factor that needs to be taken into consideration is the seasoning. Seasoned assets need to reference their correct timing on the prepayment curve, otherwise the wrong historically created prepayment percents will be applied to the asset balances. The following formula will accomplish this and change as different curves are selected on the input sheet and as the periods change:

```
= OFFSET(Vectors!$L$6,A7+Age1,MATCH(pdrPrepay1,
  lstPrepayCurve,0))*pdrPrepayStress1
```

Copy and paste this formula from P7 through P366.

11. The final formula of this Model Builder exercise is the most critical. It determines the actual dollar amount of prepayments for the period. The first part of this formula is an IF function for clean up purposes. If there is a zero balance and the prepay calculation is attempted then there will be #DIV/0 errors. To prevent this start the formula in Q7 with:

```
=IF(L7=0,0,
```

The next parts of the formula are MAX and MIN formulas. The MAX formula ensures that in cases of high default there is no negative balance that could skew prepayments. The MIN formula takes the lesser of the balance less defaults and the calculated prepay amount. The balance less defaults is the Beginning Balance minus New Defaults (L7–N7). The prepayment calculation is the Beginning Balance multiplied by the percentage asset reduction in the notional schedule multiplied by the monthly prepayment rate (as measured in SMM). A confusing part of the last statement is what "the percentage asset reduction in the notional schedule" is and why it is part of the calculation. This is the current amort factor divided by the last month's amort factor. This represents the scheduled amortization on a fractional basis and removes it from the prepayment amount. The final formula should look like:

```
=IF(L7=0,0,MAX(MIN((L7–N7),L7*O7/O6*P7),0))
```

In particular, notice that defaults (N7) are removed from the Beginning Balance as part of the MIN function. This is important because as loans approach their final periods they will have to be cleaned up. Defaults take precedence in calculations since it is assumed that a defaulted loan will not voluntarily prepay.

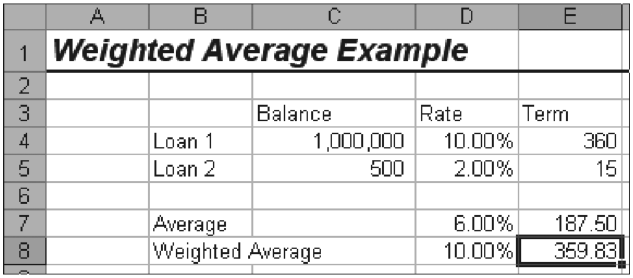
TOOLBOX

Weighted Averages Using SUMPRODUCT and SUM

When doing any analysis in finance on a pool of assets the word “average” comes up often: What is the average interest rate? How long is the average remaining term? What is the average age of the loans? The answers to all of these questions must be provided in terms of the *weighted average*, otherwise an incorrect estimation will most likely be made. The reason for this is that financial assets that are pooled together in a transaction rarely have homogenous principal balances from the start, with differences being more exacerbated as time progresses.

Principal balances are important because in a pool of assets, those with larger principal balances will have more influence on the characteristics and performance of the pool than assets with smaller balances. An easy example is to take a look at a pool of two loans, their balances, and their remaining terms. In the examples, the loans are quite different. One is fairly new and has a high principal balance and a long remaining term while the other is very seasoned and has a low principal balance and short remaining term. As seen in Figure 3.6 (and in Ch03’s Additional Files subfolder for the Model Builder files on the CD-ROM), the average remaining term and rate of this pool is taken by using the AVERAGE function in Microsoft Excel. This produces a result of 187.50 periods and a rate of 6.00 percent.

This is a misleading result because most of the pool’s money will be outstanding for much longer than 187.50 periods and at a far higher rate than 6.00 percent. The correct way to describe the average remaining term is to weight the average by the current principal balance. By using a weighted average based on current balance, the remaining term would be nearly 360 periods with a rate of 10 percent. This is done by multiplying the terms by the balances, summing up those products, and then dividing that sum by the sum of the principal. The weighted average can be calculated without using functions, but the calculation is made much easier by using SUMPRODUCT and SUM.



	A	B	C	D	E
1	**Weighted Average Example**				
2					
3			Balance	Rate	Term
4		Loan 1	1,000,000	10.00%	360
5		Loan 2	500	2.00%	15
6					
7		Average		6.00%	187.50
8		Weighted Average		10.00%	359.83

FIGURE 3.6 There is a noticeable difference between taking an arithmetic average and a weighted average.

SUMPRODUCT takes two arrays of equal length and multiplies each value in one array by the corresponding value of the second array, and then sums up all of those products. In Figure 3.6, for example, the remaining term array (E4:E5) is multiplied by the corresponding balances (C4:C5). SUMPRODUCT finishes by calculating the sum of all of these products. The arrays should be referenced in this manner:

=SUMPRODUCT(Array Reference 1, Array Reference 2)

SUMPRODUCT takes care of the first two parts of calculating a weighted average, so all that is left is to divide by the sum of the balances using the SUM function. This function combination can be used for calculating weighted averages for any loan characteristic and typically uses the current principal balance as the weighting.

Also, as seen in this chapter, the weighted average should be used for summarizing historical rates. Often, for such purpose, the OFFSET function needs to be used on the arrays to prevent zeros from skewing the average.

CHAPTER 4

Delinquency, Default, and Loss Analysis

Nearly every asset class in structured finance is subject to loan obligors not paying. This eventually generates losses. While the concept of an obligor defaulting and causing a loss is not difficult to understand, historical loss reporting, future loss projection, and how a structured transaction model is constructed to handle these are often a cause of confusion. Because there is no standardized method for reporting, projecting, or modeling loss, data is often presented in inconsistent and sometimes misleading formats. The most favored format of understanding historical loss information is the *static loss report*, which tracks losses for each origination period over time. This format is ideal for building historical loss curves and producing a projected loss curve with severity and timing that is predicated on prior asset performance.

Incorporating loss into a model can also be done a number of ways depending on certain rating agency methodologies and subjective formula techniques. This chapter lays the groundwork for understanding loss and shows one possible method for integrating the concept into a functioning model. Later in this book there are discussions about other methodologies.

DELINQUENCIES VERSUS DEFAULTS VERSUS LOSS

One reason understanding loss can be so confusing is the terminology that is involved in the process leading up to a loss. To be clear prior to any use, the following related terms are defined:

- *Delinquency*. When a payment is due on a specific date and the obligor fails to pay the payment by that date the obligor is delinquent. A more granular analysis of delinquency typically splits delinquency into monthly intervals such as 1 to 30 days delinquent, 31 to 60 days delinquent, 61 to 90 days delinquent, and so on. A key point to an obligor being considered delinquent is that there is an expectation of more payments and the possibility that the obligor will repay enough to no longer be considered delinquent.

- *Default*. Depending on company and legal definitions, a delinquency eventually becomes a default. When obligors are considered defaulted, they are no longer expected to make regular payments and legal action typically begins. It is very important to realize that the timeframe it takes for an obligor to transition from delinquency to default is usually set by legal definitions in structured transactions and that loss expectations can change depending on the time it takes to classify an obligor as a default.
- *Loss*. When a loan is considered a default, some companies instantly write the principal balance of the loan off. This is known as a *gross loss*. As time progresses, sometimes a recovery is made either through the sale of a repossessed asset or continued pursuance of credit collection.

The well-known fixed income expert, Frank J. Fabozzi, stresses the importance of terminology by defining a loss as follows:

> *A default is essentially a loan that does not get paid back. A loan may go into the 90-day delinquent category or even into foreclosure, and still be made current again. So only those loans that are ultimately liquidated for non-payment are classified as defaulted loans. Losses refer to the dollar amounts lost on defaulted loans.*[1]

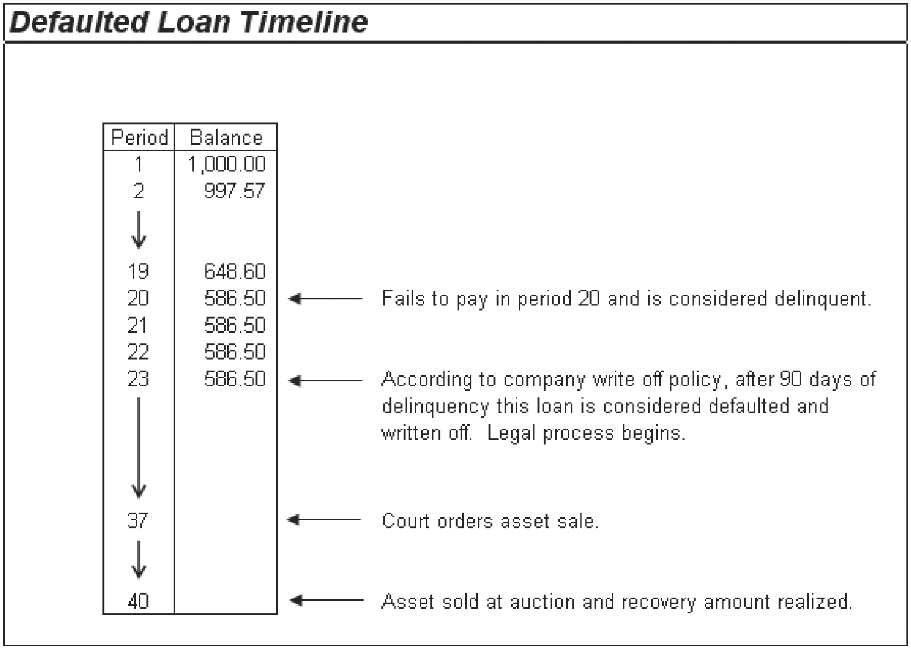
A useful example is to examine one loan in a hypothetical loss situation. Imagine a mortgage loan that had an original balance of $1,000.00, a current balance of $586.50, is in its 20th month and stops paying. Figure 4.1 depicts the timeline of events and classifications that the loan would go through.

As can be seen from Figure 4.1, the loan is considered delinquent until 90 days have passed by and then it becomes classified as a default. In this case, it is written off instantly and the legal process begins. After 14 months, a court orders a foreclosure. The asset takes three months to sell and an asset recovery is realized.

THE IMPORTANCE OF ANALYZING DELINQUENCY

Understanding the asset pool's tendency to initially stop paying on time is a logical starting point for a loss analysis. This is done by examining historical delinquency data on similar assets and is important for two reasons: *credit trends* and *liquidity*. Credit trends are an indication of whether assets will perform better or worse and whether a greater or lesser amount of loss should be expected in the future. Liquidity is a transaction-specific consideration that refers to the amount of actual cash in any given period.

[1] Frank J. Fabozzi, *Bond Credit Analysis: Framework and Case Studies* (New Hope, PA: Frank J. Fabozzi Associates, 2001), 245.



Defaulted Loan Timeline

Period	Balance
1	1,000.00
2	997.57
↓	
19	648.60
20	586.50
21	586.50
22	586.50
23	586.50
↓	
37	
↓	
40	

Fails to pay in period 20 and is considered delinquent.

According to company write off policy, after 90 days of delinquency this loan is considered defaulted and written off. Legal process begins.

Court orders asset sale.

Asset sold at auction and recovery amount realized.

FIGURE 4.1 A timeline of the events that take place when an asset defaults.

The first concern, credit trends, is important because it could mean that a future loss projection might have to be increased. Credit trends are noticeable when a delinquency analysis is performed on existing assets. If, over the course of origination, assets progressively have increased delinquency, the loss expectation may be higher. Greater delinquency sometimes means that the company originating the assets is chasing riskier assets, possibly for greater returns. In this case, the obligors' credit scores and interest rates should be examined more closely.

The second concern, liquidity, is extremely important in building a structure that is sized and risk rated correctly. If assets consistently have a high delinquency, this needs to be accounted for in a model; otherwise there might not be enough cash to cover loss, principal, interest, and possibly fees. Delinquency directly affects cash flow because a delinquent loan does not have any cash flow; but it is also not considered a default. This means that its balance is not written down, excess spread does not cover the loss, and no cash is received. After the liability side of the model is complete, this concept becomes clearer.

Both concerns require the creation of historical delinquency curves. These curves are useful in determining credit trends and the severity of delinquency. The curves are not actually used in Project Model Builder, but they are created using a Model Builder exercise since the analysis is very important.

MODEL BUILDER 4.1: BUILDING HISTORICAL DELINQUENCY CURVES

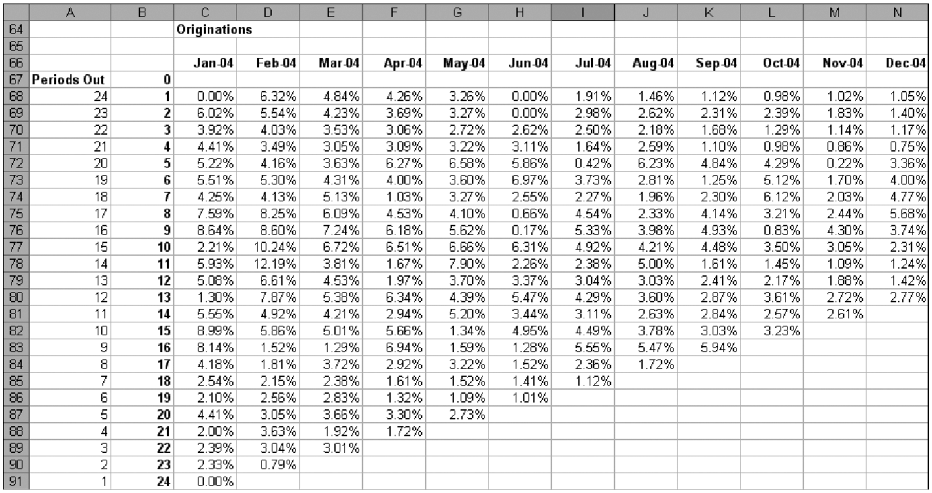
1. Ideally a historical delinquency analysis begins with data from assets similar to those in the transaction. In this exercise, data has been provided in the Ch04 folder on the CD-ROM. Open the file *MB4-1_Raw_Data* to access the raw data. Save the file as **Historical Delinquency Curves.xls**.
2. There are two sections of data provided: The top section is the current balance of loans, grouped by vintage, reported on a periods-out basis. The bottom section contains the current balance of loans that are in the 30-day delinquency bucket (that is, they have missed their payment date by 1 to 30 days). The best way to understand the magnitude and trend of the delinquencies is to calculate them on a percentage basis. First, to create an area for the rates, copy **C37:AA37** and paste this cell range in **C66:AA66**. Also copy **B38:B62** and paste this cell range in **B67:B91**. Labels for the axis of the ranges can also be copied down, but make sure to leave row 65 blank for now.
3. The proper delinquency calculation is the current month's delinquency over the prior month's ending balance. The formula for this is a simple division equation; but it needs some cleaning up to prevent #DIV/0 errors. In cell C68 enter:

   =IF(C39="","",C39/C7)

   This formula cleans up zero-balance periods and divides the delinquency by the current balance of the prior period.
4. To get a sense for all of the vintages, a weighted average is useful. As seen in Chapter 3, calculating a weighted average is most simply done using a SUMPRODUCT-SUM combination and is discussed in more detail in the Toolbox section of that chapter. Also, to prevent picking up blank cells, an OFFSET function is needed. Prior to creating the formula, the OFFSET needs a reference value to determine how many cells of data each vintage contains. Since January 2004 contains the maximum periods of data (24) and each successive month has one less period of data, a column of decreasing values is necessary. Starting in A68, enter **24**, and then in B68 enter **23**. Highlight both of these cells and drag the decreasing value down to A91. The worksheet should look like Figure 4.2 at this point.
5. Cell AC68 will contain the weighted average of the first period, AC69 the second period, and so on. In cell AC68 enter:

   =SUMPRODUCT(C68:OFFSET(B68,0,A68),C7:
   OFFSET(B7,0,A68))/SUM(C7:OFFSET(B7,0,A68))

   This weighted average formula multiplies each delinquency rate by the respective vintage's prior period current balance. This creates the weighting and then divides the sum of those values by the sum of the balances to get the weighted



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
64			Originations											
65														
66			Jan-04	Feb-04	Mar-04	Apr-04	May-04	Jun-04	Jul-04	Aug-04	Sep-04	Oct-04	Nov-04	Dec-04
67	Periods Out	0												
68	24	1	0.00%	6.32%	4.84%	4.26%	3.26%	0.00%	1.91%	1.46%	1.12%	0.98%	1.02%	1.05%
69	23	2	6.02%	5.54%	4.23%	3.69%	3.27%	0.00%	2.98%	2.62%	2.31%	2.39%	1.83%	1.40%
70	22	3	3.92%	4.03%	3.53%	3.06%	2.72%	2.62%	2.50%	2.18%	1.68%	1.29%	1.14%	1.17%
71	21	4	4.41%	3.49%	3.05%	3.09%	3.22%	3.11%	1.64%	2.59%	1.10%	0.98%	0.86%	0.75%
72	20	5	5.22%	4.16%	3.63%	6.27%	6.58%	5.86%	0.42%	6.23%	4.84%	4.29%	0.22%	3.36%
73	19	6	5.51%	5.30%	4.31%	4.00%	3.60%	6.97%	3.73%	2.81%	1.25%	5.12%	1.70%	4.00%
74	18	7	4.25%	4.13%	5.13%	1.03%	3.27%	2.55%	2.27%	1.96%	2.30%	6.12%	2.03%	4.77%
75	17	8	7.59%	8.25%	6.09%	4.53%	4.10%	0.66%	4.54%	2.33%	4.14%	3.21%	2.44%	5.68%
76	16	9	8.64%	8.60%	7.24%	6.18%	5.62%	0.17%	5.33%	3.98%	4.93%	0.83%	4.30%	3.74%
77	15	10	2.21%	10.24%	6.72%	6.51%	6.66%	6.31%	4.92%	4.21%	4.48%	3.50%	3.05%	2.31%
78	14	11	5.93%	12.19%	3.81%	1.67%	7.90%	2.26%	2.38%	5.00%	1.61%	1.45%	1.09%	1.24%
79	13	12	5.08%	6.61%	4.53%	1.97%	3.70%	3.37%	3.04%	3.03%	2.41%	2.17%	1.88%	1.42%
80	12	13	1.30%	7.87%	5.38%	6.34%	4.39%	5.47%	4.29%	3.60%	2.87%	3.61%	2.72%	2.77%
81	11	14	5.55%	4.92%	4.21%	2.94%	5.20%	3.44%	3.11%	2.63%	2.84%	2.57%	2.61%	
82	10	15	8.99%	5.86%	5.01%	5.66%	1.34%	4.95%	4.49%	3.78%	3.03%	3.23%		
83	9	16	8.14%	1.52%	1.29%	6.94%	1.59%	1.28%	5.55%	5.47%	5.94%			
84	8	17	4.18%	1.81%	3.72%	2.92%	3.22%	1.52%	2.36%	1.72%				
85	7	18	2.54%	2.15%	2.38%	1.61%	1.52%	1.41%	1.12%					
86	6	19	2.10%	2.56%	2.83%	1.32%	1.09%	1.01%						
87	5	20	4.41%	3.05%	3.66%	3.30%	2.73%							
88	4	21	2.00%	3.63%	1.92%	1.72%								
89	3	22	2.39%	3.04%	3.01%									
90	2	23	2.33%	0.79%										
91	1	24	0.00%											

FIGURE 4.2 A standard set of delinquency curves organized by periods out.

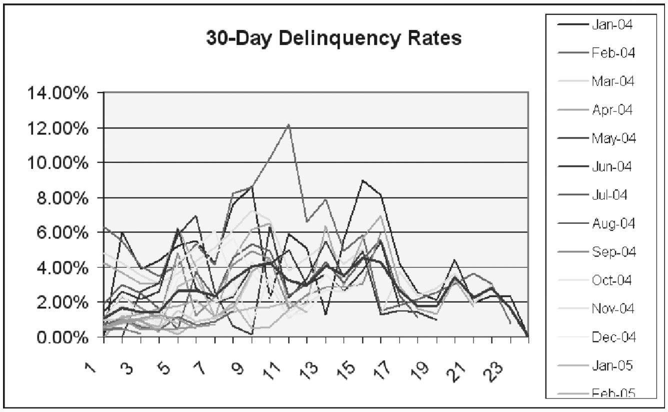


FIGURE 4.3 Delinquency performance can be quickly understood from a chart.

average rate. An OFFSET is used to make sure that no blank values are picked up.

6. Charts are often seen to represent this data as seen in Figure 4.3.

The delinquency data in Model Builder 4.1 is useful because it shows the amount that is delinquent 1 to 30 days. This is useful knowledge for later if a liquidity reserve needs to be sized in the modeling. Also, notice that there is a slight trend in the data. Earlier vintages have experienced slightly more delinquency than later vintages, suggesting an improving trend in credit quality. This could possibly be an argument for reducing loss projections.

Keep in mind that this is the most basic delinquency analysis that can be completed. Often the process is repeated for each delinquency interval (i.e., 1 to 30 days, 31 to 60 days, and so on). Delinquency information can also be used to create transition matrices, which can be used as a proxy for loss if there is a limited amount of historical data. However, if there is static vintage data, historical loss curves should be created and analyzed.

DERIVING HISTORICAL LOSS CURVES

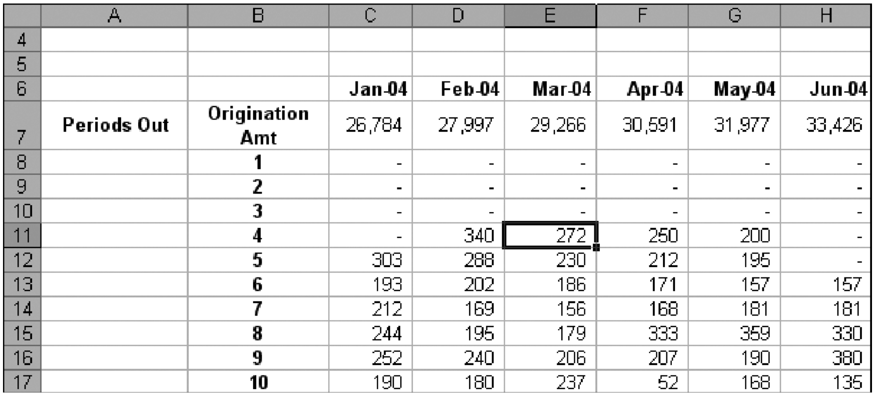
Delinquencies that eventually pay only create liquidity issues and negative drag on the transaction and are not as important as the loans that completely stop paying and are considered defaults. Similar to delinquency, default expectations for a pool of assets can be assumed from historical data. In fact, the analysis is very similar, using a static type analysis.

The importance of a static type analysis should not be understated. To get a comprehensive understanding of the performance of an asset it must be tracked from its origination to maturity, without the influence of other loans from different origination periods. This is the core concept behind a static analysis. The term *static* is used because individual vintages are examined independent of other vintages. The losses for all loans in a vintage are tracked and recorded as time progresses. The comparison between loss and balance is only within the vintage under analysis. No other loan's losses or balances are added at any time. For example, in Figure 4.4, originations are displayed across row 7. The date of these originations is directly above in row 6. For each origination period the losses can be seen going vertically in their respective columns.

A quick example of interpreting Figure 4.4 is looking at loans originated in March 2004. In that month, there were $272 in losses four periods from origination. The four periods out corresponds to July 2004, which is cell E11 in Figure 4.4.

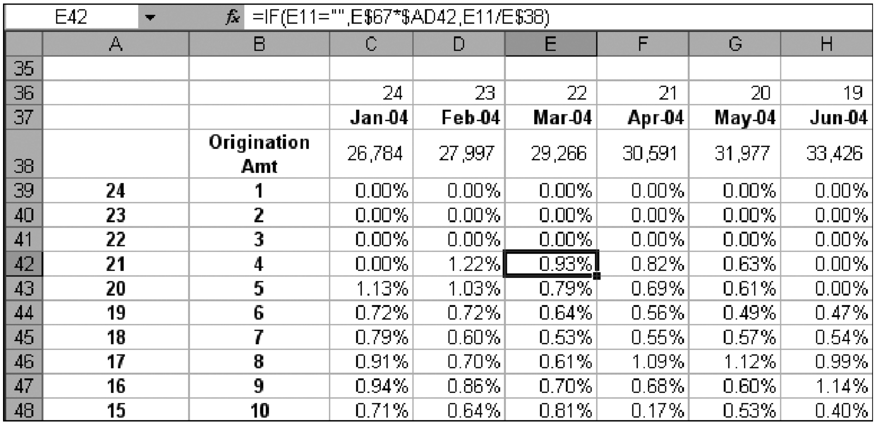
It is important to fully understand what the $272 represents. In this example, that amount is the gross loss reported in that month. *Gross loss* is the dollar amount of loan balances that are assumed to remain unpaid. The typical method of understanding the magnitude of the loss per period is by representing it as a fraction of origination. Specifically, dividing the historical loss per period associated with a specific origination by the origination amount results in a percentage of loss that the loans in the origination pool experienced. Taking the earlier example of the $272 in July 2004, divide the loss amount by the originations for March 2004 (i.e., 29,266). This results in 0.93 percent as seen in Figure 4.5.

The rationale for presenting the loss data this way is that it allows for a direct comparison of losses between origination pools. Also, this method of reporting



	A	B	C	D	E	F	G	H
4								
5								
6			**Jan-04**	**Feb-04**	**Mar-04**	**Apr-04**	**May-04**	**Jun-04**
7	**Periods Out**	**Origination Amt**	26,784	27,997	29,266	30,591	31,977	33,426
8		**1**	-	-	-	-	-	-
9		**2**	-	-	-	-	-	-
10		**3**	-	-	-	-	-	-
11		**4**	-	340	272	250	200	-
12		**5**	303	288	230	212	195	-
13		**6**	193	202	186	171	157	157
14		**7**	212	169	156	168	181	181
15		**8**	244	195	179	333	359	330
16		**9**	252	240	206	207	190	380
17		**10**	190	180	237	52	168	135

FIGURE 4.4 Raw loss data is best presented on a static loss basis.



E42 ▾ fx =IF(E11="",E$67*$AD42,E11/E$38)

	A	B	C	D	E	F	G	H
35								
36			24	23	22	21	20	19
37			**Jan-04**	**Feb-04**	**Mar-04**	**Apr-04**	**May-04**	**Jun-04**
38		**Origination Amt**	26,784	27,997	29,266	30,591	31,977	33,426
39	**24**	**1**	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
40	**23**	**2**	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
41	**22**	**3**	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
42	**21**	**4**	0.00%	1.22%	0.93%	0.82%	0.63%	0.00%
43	**20**	**5**	1.13%	1.03%	0.79%	0.69%	0.61%	0.00%
44	**19**	**6**	0.72%	0.72%	0.64%	0.56%	0.49%	0.47%
45	**18**	**7**	0.79%	0.60%	0.53%	0.55%	0.57%	0.54%
46	**17**	**8**	0.91%	0.70%	0.61%	1.09%	1.12%	0.99%
47	**16**	**9**	0.94%	0.86%	0.70%	0.68%	0.60%	1.14%
48	**15**	**10**	0.71%	0.64%	0.81%	0.17%	0.53%	0.40%

FIGURE 4.5 The raw loss data should be converted to percentages of original vintage balance.

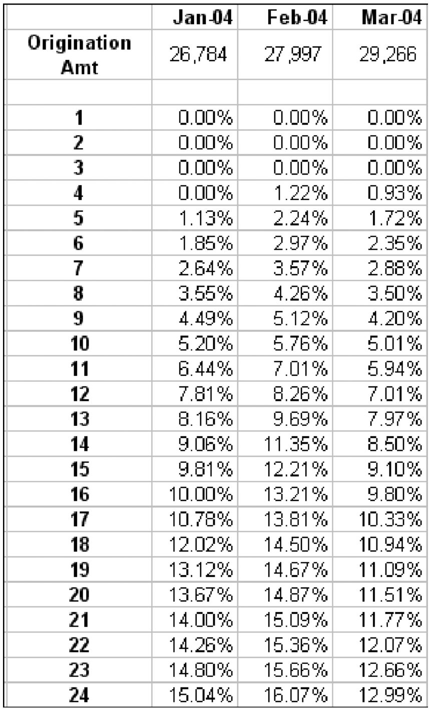
prevents inaccurate loss percentages that could occur if the loss percentage was calculated in a dynamic manner. Standard & Poor's provides excellent reasoning on why static loss data is preferred:

> *An analysis of static pool data is preferred, since it demonstrates loss performance over the full liquidation period of a pool. Static pool data also can provide insight into changing portfolio characteristics, underwriting, or collection policies. Movement in such variables may not be readily apparent in dynamic portfolio data because of the constantly changing mix in new and aged receivables. Additionally, data based on annual losses can understate*

> *loss performance during periods of rapid portfolio growth. When analyzing high-growth portfolios, the current period's losses may be compared against the prior year's portfolio. This "growth adjustment" takes into account that losses on new originations may not happen immediately and, therefore, occur in the next annual period.*[2]

Tracking the loss history of loans based on origination gives a better estimation of the asset performance over time.

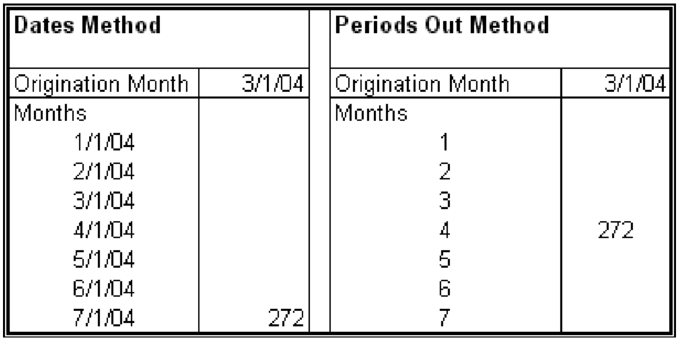
Once the monthly loss amount is determined, it is very easy to calculate the cumulative loss percentage, which is the sum of all the monthly loss percentages, up to the current period. Using the same origination pool as the other examples (March 2004) it is evident that the cumulative loss through March 2006 is 12.99 percent (see Figure 4.6).



	Jan-04	Feb-04	Mar-04
Origination Amt	26,784	27,997	29,266
1	0.00%	0.00%	0.00%
2	0.00%	0.00%	0.00%
3	0.00%	0.00%	0.00%
4	0.00%	1.22%	0.93%
5	1.13%	2.24%	1.72%
6	1.85%	2.97%	2.35%
7	2.64%	3.57%	2.88%
8	3.55%	4.26%	3.50%
9	4.49%	5.12%	4.20%
10	5.20%	5.76%	5.01%
11	6.44%	7.01%	5.94%
12	7.81%	8.26%	7.01%
13	8.16%	9.69%	7.97%
14	9.06%	11.35%	8.50%
15	9.81%	12.21%	9.10%
16	10.00%	13.21%	9.80%
17	10.78%	13.81%	10.33%
18	12.02%	14.50%	10.94%
19	13.12%	14.67%	11.09%
20	13.67%	14.87%	11.51%
21	14.00%	15.09%	11.77%
22	14.26%	15.36%	12.07%
23	14.80%	15.66%	12.66%
24	15.04%	16.07%	12.99%

FIGURE 4.6 A cumulative loss curve sums the periodic loss amounts.

[2] Standard & Poor's, *Structured Finance: Equipment Leasing Criteria* (New York: McGraw-Hill, 1999), 17.



Dates Method	
Origination Month	3/1/04
Months	
1/1/04	
2/1/04	
3/1/04	
4/1/04	
5/1/04	
6/1/04	
7/1/04	272

Periods Out Method	
Origination Month	3/1/04
Months	
1	
2	
3	
4	272
5	
6	
7	

FIGURE 4.7 The difference between organizing data using a periods-out versus a dates methodology.

As this point, it is worth noting that there can be some confusion as to how the loss curves progress over time. One part to be mindful of is whether the unit of time is represented as a period of time or a reference date. The method, used for the examples thus far, represents time progression using periods and is called the *periods-out method*. This method means that loans originated in March 2004 had .93 percent of loss four periods out (July 2004); .79 percent of loss five periods out (August 2004); and so on. However, loss data can also be provided without the periods but with dates instead. The dates typically start with the earliest origination period and continue vertically on a monthly basis. The key difference is that the dates method sets the time reference for all of the origination periods, while the periods-out method has an independent time reference for each origination period.

In the dates method loans originated in March 2004 show $272 of loss in July 2004, seven cells down. The same origination period in the periods-out method shows the $272 of loss four cells down. Essentially, in a periods-out method each time progression of a cell is based off each origination pool date, while a “dates” method is based off of the very first origination date. To help make this clear, look at Figure 4.7.

MODEL BUILDER 4.2: BUILDING HISTORICAL AND PROJECTED LOSS CURVES

1. Open the file *MB4-2_Raw_Data.xls* from the Ch04 folder on the CD-ROM. This file is similar to the delinquency one, but with changes relevant to loss analysis. The first noticeable difference is that there are only origination balances for each of the pools. This is because static loss analysis is done off of the original balance of the loan. Save this file under a different name so it can be worked in.

2. Similar to the delinquency analysis, copy the origination dates and balances to rows 37 and 38. Also copy the periods in column B so that it starts with Origination Amt in row 38.
3. Starting in cell C39 enter the following formula:

   =IF(C8="","",C8/C$38)

   This is similar to the delinquency formula; but it bases all of the loss on original balance instead of current balance. Copy this formula over the range C39:AA62.
4. So far monthly loss percentages have been created. This data is better analyzed looking at cumulative percentages to see the "curve." Copy the vintage dates and origination amounts to row 65 and 66. Also copy the periods in column B to start at row 66. The cumulative percentage is the current period's loss plus the previous period's cumulative percentage. Before entering a formula, insert a blank row in row 67. This should put a row of empty cells between the first period and the origination amounts so the formula that will be entered does not add the origination amounts to the cumulative loss percentage. With that complete enter the following formula in C68:

   =IF(C39="","",C39+C67)

   Copy this formula over the range C68:AA91. The cumulative curves are often summarized in a line graph that looks like Figure 4.8.

Before continuing further with Model Builder 4.2, it is important to understand how to analyze the curves to use them for projections.

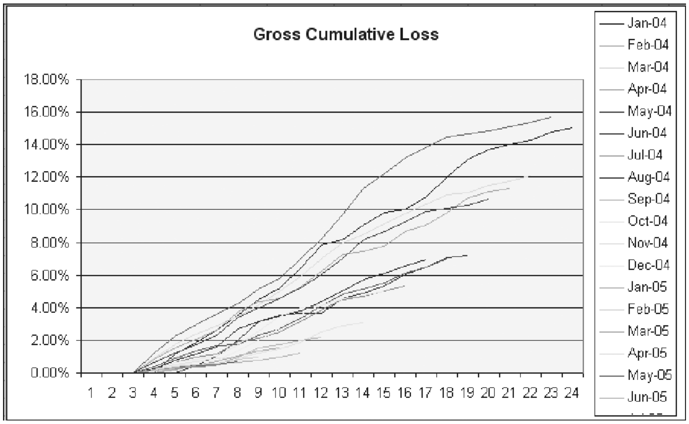


FIGURE 4.8 A graph of cumulative loss curves.

ANALYZING HISTORICAL LOSS CURVES

Vintage loss curves created using a static methodology have two important characteristics that should be identified: *severity* and *timing*. The severity is the final cumulative loss percent per vintage. This is how much of the original balance of a particular vintage is assumed to be defaulted and uncollectible. The timing is how much loss has been taken by a certain point in time, ending at the final maturity of the assets. If the assets in the Model Builder example had final maturities of 24 months, then the timing of loss for any period can be determined by dividing the cumulative loss percentage in that period by the final cumulative loss percentage (period 24).

Loss timing is important to understand because it can have profound effects on structured transactions. If the loss timing is front loaded, which means that losses take place quickly the assets will erode quickly. This directly impacts excess spread in a transaction, which is the first source of protection against loss. A transaction modeled with a front-loaded curve versus a regular curve will require more enhancement since there is less time for excess spread to generate. Back-loaded curves, where losses take place near the end of the tenor of the assets also have special effects on structured transactions. If loss does not take place until late in the transaction, enhancement needs to be sized and kept for those periods. If a transaction was modeled with a regular loss curve and losses were actually back-loaded, important structural features such as triggers and reserve accounts might be inadequate to protect against the back-loaded loss.

MODEL BUILDER 4.2 CONTINUED

1. Label cell AC38 **Weighted Avg Curve.** To get a summary of the severity of the historical loss curves a weighted average curve needs to be created. This is done using the following formula starting in AC39:

   =SUMPRODUCT(C39:OFFSET(B39,0,A39),$C$38:
   OFFSET($B$38,0,A39))/SUM($C$38:OFFSET($B$38,0,A39))

   Copy this formula down to AC62. Also, since these are the monthly losses, sum them up in AC64 to get the weighted average loss.
2. Timing should be analyzed on a monthly basis first and then cumulative. Take the first period's monthly loss amount and divide it by the sum of all the monthly loss. Label cell AD38 **Timing**, and start the following formula in AD39:

   =AC39/$AC$64

   Copy this formula down to AD62. A sum of this column in AD64 should equal 100 percent.

3. A useful way to observe the timing is to make a cumulative timing curve. Do this by entering the following formula in AE40:

   =AE39+AD40

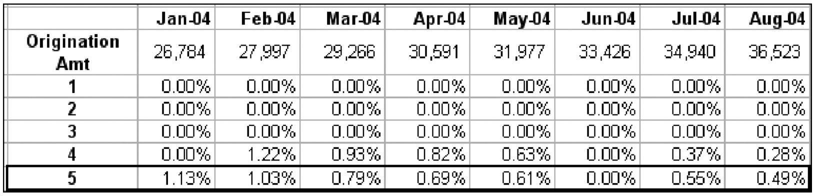
   Notice that this started one more row down than the other formulas to avoid having the label added in a formula. Copy this formula down to AE62. Model Builder 4.2 will finish up after the next section on projecting loss curves.

PROJECTING LOSS CURVES

If no trend is evident and there are years of data that encompass the tenor of the asset, then the weighted average curve created in the previous section can be used as a projected loss curve. However, most of the time industries and companies go through cycles of increasing and decreasing loss. Also, particularly with assets in emerging markets, a relatively short time span of data is available. Both of these issues create the need to project out loss curves.

The first issue, trending, is observed by looking at the same period for each vintage. In Model Builder 4.2, the monthly losses have a noticeable decreasing trend. Look at period 5 in Figure 4.9 and notice that in general each successive vintage after January 2004 has a decreasing loss amount. Most of the periods are experiencing such a trend. The company could argue that a weighted average curve based solely on the data overstates loss because the newer vintages are expected to have a lower loss amount in later periods, but these amounts are not reported and therefore not captured in the weighted average loss curve.

A thorough loss analysis when trending is involved requires the ability to observe the full spectrum of loss an asset may experience from origination to maturity. Taking the weighted average losses for each period will only produce accurate curves depending on the breadth of the historical loss data vis-à-vis the age of the assets. The usefulness of the historical loss curves can be assessed by determining how many of the loss curves have tracked data from origination to maturity. As an example, assume the current date is January 2006 and in our examples the data is provided as early as January 2004. Also assume that the final maturity of the assets is 24 periods. This means that if originations and loss data is



	Jan-04	Feb-04	Mar-04	Apr-04	May-04	Jun-04	Jul-04	Aug-04
Origination Amt	26,784	27,997	29,266	30,591	31,977	33,426	34,940	36,523
1	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
2	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
3	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
4	0.00%	1.22%	0.93%	0.82%	0.63%	0.00%	0.37%	0.28%
5	1.13%	1.03%	0.79%	0.69%	0.61%	0.00%	0.55%	0.49%

FIGURE 4.9 Trends should be looked for in vintages across periods.

provided monthly, there could be one vintage that has reached maturity or "termed out." For instance, loans originated in January 2004, with a final maturity of 24 months should have all matured by January 2006. Since the loss data is from January 2004 through January 2006, there is loss history from every part of the loans' term. However, loans originated in April 2004 will only have a partial loss history, since there would only be 21 months of data (May 2004 to January 2006).

If there is a trend in the data and there are few vintages that have "termed out," the earlier vintages will have a strong impact on the weighted average curve. To account for such trends, the newer vintages need to be adjusted. For instance, if losses are trending upwards and the later vintages aren't "grossed up" for expected loss, the weighted average method will understate loss. The opposite will occur if losses are trending downwards, resulting in an overstatement of loss.

To account for trends, later vintages need to be adjusted using a timing curve extrapolated from a set of "base" originations. A "base" origination should be a historical origination from the static loss data that is demonstrative of the expected timing of the assets. As long as the asset performance is not extremely volatile, it would be logical to assume that future assets will take losses in a similar manner. Third-party timing curves, such as those produced by the Public Securities Association (PSA) or rating agencies can be used to adjust losses. Also, more sophisticated statistical analyses can be performed on the loss data to determine trends. The results of such analyses would provide a basis for trending. The continuation of Model Builder 4.2 takes the most fundamental approach to projecting loss.

MODEL BUILDER 4.2 CONTINUED

1. The final step in a complete static loss analysis is adjusting newer vintages to account for trending. To do this, the monthly loss for each vintage that is not complete needs to be extrapolated based on timing. First, make room to work underneath the monthly loss percentage area. Insert enough rows so rows 64 through 67 are clear.
2. Label row 64 in B64 as **Loss Sev. Taken**. This is how much loss as a percent of original balance has been taken for each vintage. To get the correct amount a SUM formula with the OFFSET function needs to be used. For the OFFSET to reference the correct amount of information per vintage create a row of descending values starting with 24 in C36, 23 in D36, and so on. Descend the values until Z36 where the value should be 1. In C64 enter the following formula:

   =SUM(C39:OFFSET(C38,C36,0))

   This formula will only sum the severities that are derived from historical data. The importance of the OFFSET becomes clearer later as projected severities are created in the area.

3. In the next row down, label cell B65 **Loss % Taken**. This row is a percentage calculation of how much loss the vintage under analysis has taken compared to the weighted average timing curve. For instance, the January 2004 vintage has a full 24 months reported, so it has taken 100 percent of loss that it is expected. The February 2004 vintage is only 23 months so it is short one month of loss and has taken slightly less than 100 percent loss. To calculate the percentage of loss that has been taken, enter the following formula starting in C65:

   =OFFSET($AE$38,C36,0)

   This formula is a basic OFFSET for the timing curve, depending on the seasoning of the vintage. Copy this formula through to Z65.
4. By knowing the percentage of loss that has been taken, the calculation for the percentage of loss that needs to be distributed is determined by subtracting the prior value by one. Label cell B66 **Loss to be Dist**. Enter the following formula in cell C66 and copy it across to Z66:

   =1−C65

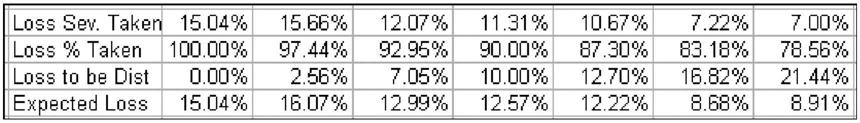
5. The expected loss is the loss severity taken divided by the loss percent taken so far. If a vintage has taken 100 percent of its loss, then it will be the same loss severity, however for vintages that have taken less than 100 percent the severity will be grossed up. Label cell B67 **Expected Loss** and enter the following formula in C67 and copy it across to Z67:

   =C64/C65

6. With the expected loss for each vintage calculated, the next step is to project the monthly loss for periods in the future. This can be done directly in the monthly loss formula since there is already an IF statement set up. Click on cell C39 and recall that an IF statement was set so that if there was no data (that is “”), then no data should be populated. However, it is now known that if there is no data, there should be a projection. The projection is going to be the expected loss amount multiplied by the projected timing of loss. This is summarized by the following formula that cell C39 should be updated to:

   =IF(C8="",C$67*$AD39,C8/C$38)

   This formula reads: If there is no monthly loss data project it by taking a projected timing curve and multiplying that curve by the expected loss amount, otherwise the loss is based on historical data. Copy this formula across the range C39:W62. Only this range should be used since October 2005 onwards has so few data points that the calculations will cause #DIV/0 errors. At this point the bottom part of the monthly loss table should look like Figure 4.10.



Loss Sev. Taken	15.04%	15.66%	12.07%	11.31%	10.67%	7.22%	7.00%
Loss % Taken	100.00%	97.44%	92.95%	90.00%	87.30%	83.18%	78.56%
Loss to be Dist	0.00%	2.56%	7.05%	10.00%	12.70%	16.82%	21.44%
Expected Loss	15.04%	16.07%	12.99%	12.57%	12.22%	8.68%	8.91%

FIGURE 4.10 The additional rows are used to project expected loss.

7. The last step is to create a new weighted average curve, taking into account the projected amounts. Label cell AG38 **Adj. WA Curve** and in AG39 enter the following formula:

   =SUMPRODUCT(C39:W39,$C$38:$W$38)/SUM($C$38:$W$38)

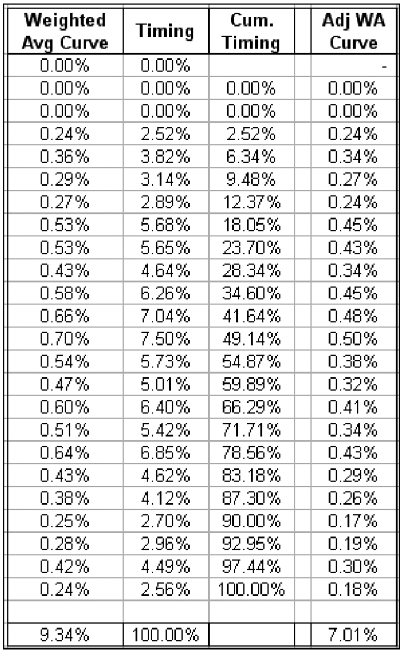
   Copy this formula down to cell AG62. This is a straightforward weighted average formula, taking into account ALL of the data for each period (up to column W). When the individual monthly data is summed in cell AG69, the difference is apparent between using an adjusted curve and a purely historical curve when trending is taking place. In the latter example, a loss curve of 9.34 percent would be used, while in the former case a much lower loss curve of 7.01 percent would be used due to trending. See Figure 4.11 for a comparison.

The previous sections described in detail the most common analyses performed on static loss data, however it is by no means exhaustive. There are many situations that will require different methodologies such as extremely volatile data, an insufficient quantity of data, a change in assets, etc. Understanding the fine detail of each situation and what drives loss is the key to choosing the right methodology. Two different static loss histories may appear very similar, but the methodology that should be employed often depends on information that is not on the data tape. These other methodologies can range from calculation intensive analysis, such as examining the slopes of the worst vintages to a very simple comparables study.

Regardless of the methodology that is used to analyze loss, understanding loss and what causes it in a transaction is possibly the most important component of structured finance modeling. A majority of the structure revolves around the loss and exists to mitigate it. This will become more apparent as loss expectations are implemented in the model.

INTEGRATING LOSS PROJECTIONS

The first part of this chapter focuses on understanding loss from a historical perspective and attempting to extrapolate future loss from the history. This second part takes the knowledge garnered from the history and applies it so loss can be



Weighted Avg Curve	Timing	Cum. Timing		Adj WA Curve
0.00%	0.00%			-
0.00%	0.00%	0.00%		0.00%
0.00%	0.00%	0.00%		0.00%
0.24%	2.52%	2.52%		0.24%
0.36%	3.82%	6.34%		0.34%
0.29%	3.14%	9.48%		0.27%
0.27%	2.89%	12.37%		0.24%
0.53%	5.68%	18.05%		0.45%
0.53%	5.65%	23.70%		0.43%
0.43%	4.64%	28.34%		0.34%
0.58%	6.26%	34.60%		0.45%
0.66%	7.04%	41.64%		0.48%
0.70%	7.50%	49.14%		0.50%
0.54%	5.73%	54.87%		0.38%
0.47%	5.01%	59.89%		0.32%
0.60%	6.40%	66.29%		0.41%
0.51%	5.42%	71.71%		0.34%
0.64%	6.85%	78.56%		0.43%
0.43%	4.62%	83.18%		0.29%
0.38%	4.12%	87.30%		0.26%
0.25%	2.70%	90.00%		0.17%
0.28%	2.96%	92.95%		0.19%
0.42%	4.49%	97.44%		0.30%
0.24%	2.56%	100.00%		0.18%
9.34%	100.00%			7.01%

FIGURE 4.11 The new adjusted weighted average curve is less than the original weighted average curve after a decreasing loss trend is taken into account.

taken into account when generating cash flows. Two methods of calculating loss exist for structured finance modeling: original balance calculation and current balance calculation. The correct one to use depends on the type of loss curve that is integrated into the model.

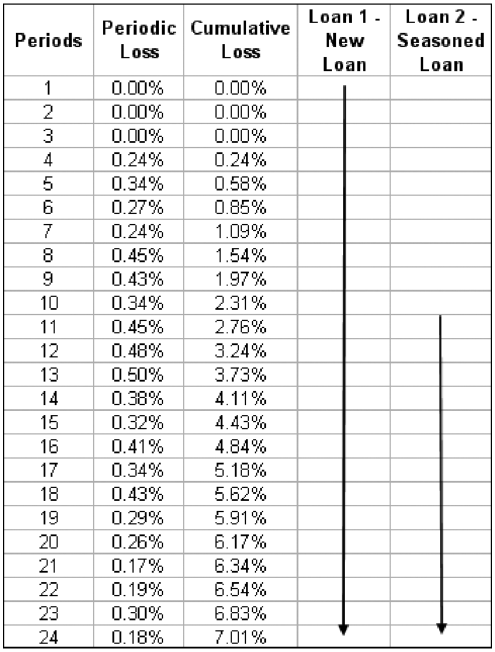
The first method, original balance calculation, multiplies a monthly loss severity by the original balance of the assets. This is used when historical loss analysis has been completed on assets and when historical loss severities have been calculated off of original balance. If 100 percent of the timing curve is taken and there is no credit for seasoning, the dollar loss amount as a percentage of the asset original balance should be exactly the same as the gross cumulative loss assumption.

The other method calculates loss by multiplying a *monthly default rate* (MDR) by the current balance. Monthly default rates are primarily employed when using a *Standard Default Assumption* (SDA) curve as the loss projection. In this case the dollar amount of loss will not be related to a percent of the original balance.

Regardless of the methodology, something to realize about loss projection is that it is a percentage of the asset balance. This does not seem that unusual when using a representative line style of amortization. The assets have been aggregated and should therefore have percentages of loss taken out. However, it may seem unusual when using a loan level style of amortization because a percentage of loss is taken out of an individual loan. In reality a loan will either default or not. There is no concept of part of a loan defaulting. In modeling, however, a loss curve will be applied to each loan and the results aggregated. This concept becomes more important when thinking about seasoning and default timing.

The Effects of Seasoning and Default Timing

When a loan has begun to amortize or is seasoned, the expected loss amount will change because a seasoned loan is on a different part of the loss curve than a new loan. For example, a loan that is brand new with a final maturity of 24 months might have a loss curve that is 24 periods in length. By month 24 the loan will have taken 100 percent of its expected loss. Imagine that the loan was already 10 months old when it was sold into the transaction. This means that 10 months of loss should be expected to already have taken place. Figure 4.12 shows the difference of two loans with different seasoning and their expected remaining loss.



Periods	Periodic Loss	Cumulative Loss	Loan 1 - New Loan	Loan 2 - Seasoned Loan
1	0.00%	0.00%	↓	
2	0.00%	0.00%		
3	0.00%	0.00%		
4	0.24%	0.24%		
5	0.34%	0.58%		
6	0.27%	0.85%		
7	0.24%	1.09%		
8	0.45%	1.54%		
9	0.43%	1.97%		
10	0.34%	2.31%		
11	0.45%	2.76%		↓
12	0.48%	3.24%		
13	0.50%	3.73%		
14	0.38%	4.11%		
15	0.32%	4.43%		
16	0.41%	4.84%		
17	0.34%	5.18%		
18	0.43%	5.62%		
19	0.29%	5.91%		
20	0.26%	6.17%		
21	0.17%	6.34%		
22	0.19%	6.54%		
23	0.30%	6.83%		
24	0.18%	7.01%		

FIGURE 4.12 A new loan will be expected to take a full 7.01 percent of loss, while a loan seasoned 10 months is assumed to have already taken 2.31 percent loss, leaving the expectation of 4.70 percent of loss to be incurred.

The effects of seasoning are accounted for in a model by calculating the seasoning of a representative line or individual loan and making sure that the loss applied for each period corresponds to the correct place on the default curve.

Seasoned loans can also have very different loss expectations depending on the default timing curve. Earlier, default timing and the problems that can arise from different default timing curves was discussed. However, all of that analysis assumed a new loan. If a loan is seasoned and the default timing curve is front loaded, there is a good chance that the loan has already taken a significant amount of its expected loss. Once Project Model Builder is complete, the differences in loss expectation due to seasoning and default timing can be examined by varying the loan age and timing curve.

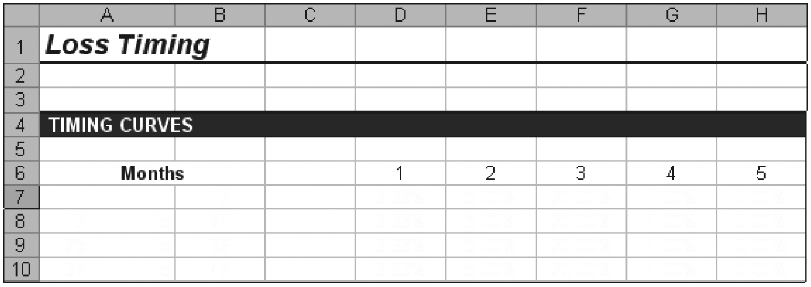
MODEL BUILDER 4.3: INTEGRATING DEFAULTS IN ASSET AMORTIZATION

1. Start on the Inputs sheet and label the following cells:
   E17: **Gross Cumulative Loss**
   F17: **Loss Stress**
   G17: **Loss Timing Curve**
   H17: **SDA Curves**
   Underneath each label is where the values will be entered. For now enter **1.00%** in cell E18 and name this cell **pdrCumLoss1**, enter 1 in F18 and name the cell **pdrLossStress1**. Before cells G18 and H18 can be created, some work needs to be done on the Vectors sheet.
2. On the Vectors sheet Chapter 3 ended on column R. Leave column S blank for spacing purposes and label cell T4 **Defaults**. Columns T through X are where the timing curves will be stored. Label cells T5 through X5 **Timing Curve 1**, **Timing Curve 2**, and so on. Name the range S5:X5 **lstDefaultCurve**. It is important to include the blank cell S5 so the data validation list will have the option of a blank value.
3. While on the Vectors sheet move on to cells Z5:AD5. Label these cells **Default Rate 1**, **Default Rate 2**, and so on. Make sure to leave Y5 and AE5 blank for spacing purposes. Move on to cell AF5 and label that cell **SDA 50%**, AG5 **SDA 100%**, and AH5 **SDA 200 %**. Name the range AF5:AI5 **lstSDA**.
4. Go back to the Inputs sheet and create a data validation list in cell G18 using **lstDefaultCurve** as the list range. Name cell G18 **pdrLossTime1**. Create another data validation list in H18 using **lstSDA**. Name cell H18 **SDA_Loss**.
5. At this point there is an input for the loss severity and a selector for timing. The severity can be entered and changed quickly depending on the historical loss analysis results. The timing curve has been set up so there are five curves to choose from. Up to this point only the labeling has been created, so an actual system of determining timing needs to be implemented. This is best done with a table that allows time to be parsed in a flexible manner, with the timing of loss varying between time increments. Since this table takes up room and is different

from most of the other items in the model, insert a new sheet after the Cash Flow sheet and name it **Loss Timing**. On the Loss Timing sheet, label cell A4 **Loss Timing.** Label cell A6 **Months**. Cells D6 through H6 will be the labels for the loss timing curves. Use a numbering system from 1 to 5, **1** being the number entered for cell D6, **2** for E6, and so on. At this point, the sheet should look like Figure 4.13.

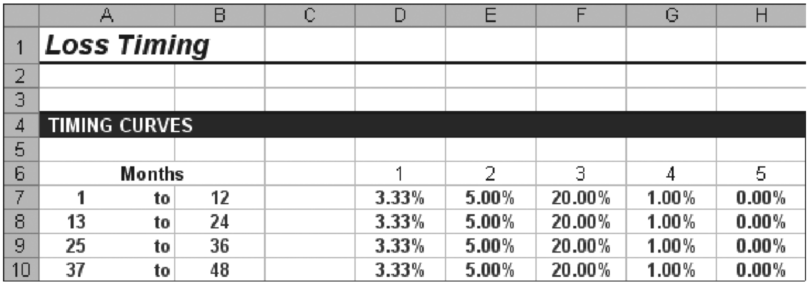
6. Still on the Loss Timing sheet enter a **1** in cell A7. This represents the first period that the loss timing starts with. In cell B7 enter **12**. This represents period 12 on the loss timing curve. What is being created here is the parsing of time that will be referenced later; in this case period 1 through period 12. A quick method of making this appear as a label, but retain the number values for referencing purposes later is to use a custom format for the cell. Right-click cell A7 and click **Format Cells**. In the Format Cells dialog box, click the **Number** tab, select **Custom** as the category. In the Type text box enter **#,## "to"**. This should make the cell look like the cell in Figure 4.14.

   The cell will still have a numerical value, but can be read quickly as a parsing of time. The cells below A7 and B7 should increase according to the interval of



	A	B	C	D	E	F	G	H
1	*Loss Timing*							
2								
3								
4	TIMING CURVES							
5								
6	**Months**			1	2	3	4	5
7								
8								
9								
10								

FIGURE 4.13 The loss timing sheet is structured so loss scenarios can be toggled quickly.



	A	B	C	D	E	F	G	H
1	*Loss Timing*							
2								
3								
4	TIMING CURVES							
5								
6	**Months**			1	2	3	4	5
7	**1 to**	**12**		**3.33%**	**5.00%**	**20.00%**	**1.00%**	**0.00%**
8	**13 to**	**24**		**3.33%**	**5.00%**	**20.00%**	**1.00%**	**0.00%**
9	**25 to**	**36**		**3.33%**	**5.00%**	**20.00%**	**1.00%**	**0.00%**
10	**37 to**	**48**		**3.33%**	**5.00%**	**20.00%**	**1.00%**	**0.00%**

FIGURE 4.14 Using a custom cell format retains the numerical value creating greater functionality for references later.

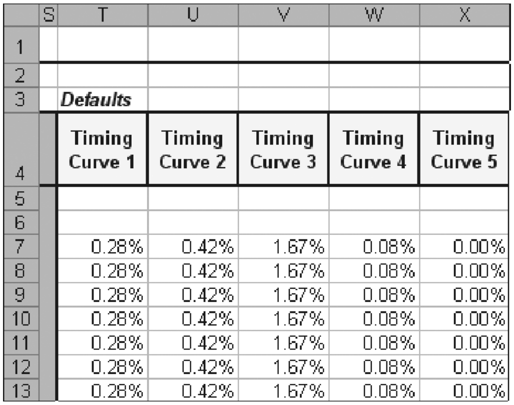
time. In this case, cells A8 and B8 will be 13 and 24 respectively. Continue this pattern down through row 36 so there is a maximum of 360 periods.

7. The purpose of the table made in step 6 is to create possible loss timing scenarios. Scenario 1 (labeled so in cell D6) will have percentages in cells D8 through D36 that represent the timing of loss during each interval that was set up in the A and B columns. For example, enter **3.33333333%**—or simply enter **= 100/30** as an easier way to get this value—in cell D8. This means that 3.33333333 percent of the loss severity will be applied to assets in the first year of their term. For instance, if the loss severity over the life of an asset is expected to be 10 percent, .33333333 percent (10% * 3.33%) would be expected to occur in the first 12 months. For now assume that 3.33333333 percent of loss will occur in each interval for Scenario 1 (D8:D36). For 360 periods parsed equally into years this should equal 100 percent. In fact, a complete timing curve should always equal 100 percent, otherwise an incorrect loss amount is being applied. The other loss timing scenarios can be left blank for now. Later in the book, when scenario selection is explained, the other timing scenarios will be entered.
8. Loss timing is often expressed as intervals of time (such as 3.33333333 percent in months 1 to 12), but models are typically run more granularly such as monthly, therefore loss timing needs to be converted to the model's periodicity. Ultimately a monthly vector will be created so the most logical place to store this vector is on the Vectors sheet. Remember that in step 2 an area was created for five Timing Curves (columns T through X). An OFFSET-MATCH combination is the formula that will be used to pull the correct periodic loss timing. In cell T7 on the Vectors sheet, enter the following formula:

```
=OFFSET('Loss Timing'!D$6,MATCH($A7,'Loss Timing'!$A$7:
    $A$36,1),0)/12*PmtFreqAdd
```

This formula is similar to the others that use OFFSET-MATCH, with a few exceptions. In this case the start of each loss timing scenario is referenced by column (D$6). That reference cell is offset by matching the current period on the Vector sheet against the intervals in column A on the Loss Timing sheet only. The fact that column A is only used is extremely important for this formula to work correctly. The reason this column is only used is because the type of MATCH that is being used is set to a 1. This means that the formula will find the largest value that is less than or equal to the look up value. If the rate for period 14 were trying to be determined, the largest value on the Loss Timing sheet's cells A7:A36 is 13. This corresponds to the second interval of timing on the Loss Timing sheet, which is the correct interval to be referenced (13 to 24). A 1 match type works only in the case of referencing the lower bound of the intervals.

The other exceptions are the divisors in the formula. The amount returned from the OFFSET-MATCH is based on the interval. To get to the periodic amount the interval amount needs to be divided by the model's periodicity. If



	S	T	U	V	W	X
1						
2						
3		*Defaults*				
4		Timing Curve 1	Timing Curve 2	Timing Curve 3	Timing Curve 4	Timing Curve 5
5						
6						
7		0.28%	0.42%	1.67%	0.08%	0.00%
8		0.28%	0.42%	1.67%	0.08%	0.00%
9		0.28%	0.42%	1.67%	0.08%	0.00%
10		0.28%	0.42%	1.67%	0.08%	0.00%
11		0.28%	0.42%	1.67%	0.08%	0.00%
12		0.28%	0.42%	1.67%	0.08%	0.00%
13		0.28%	0.42%	1.67%	0.08%	0.00%

FIGURE 4.15 The timing curve is represented on a monthly basis on the Vectors sheet.

the model was always monthly then all that needs to be done is to divide by 12. However, to automate the model in case the periodicity is quarterly, semiannual, or annual multiplying by the Payment Frequency Additive is necessary. Make sure to copy the completed formula through T366. So far this area should look like Figure 4.15.

9. Still on the Vectors sheet, the next step is to come up with the correct periodic default rate. This is the final rate that will be applied to a balance to come up with a dollar amount of loss. This rate consists of severity multiplied against periodic timing. Also, this area is where any stress should be applied to the loss curve. Recall that in Step 3 columns Z:AD on the Vector sheet were set aside for this purpose. In Z7 enter the following formula:

   =(pdrCumLoss1*pdrLossStress1)*T7

   The formula takes the overall loss severity from the Inputs sheet (pdrCumLoss1), multiplies it by a stress factor if desired (pdrLossStress1), and then multiplies that product by the current period's timing. This formula will produce the rate that should be applied against the dollar balance to derive the dollar loss amount for a period. Copy this formula into the range Z7:AD366.

10. So far this section has focused on user-generated loss curves; however, there are times when a preexisting loss curve should be used, particularly with long-term assets such as mortgages. Earlier an area was set aside for Standard Default Assumption (SDA) curves. These curves are fixed amounts that have been determined by the Public Securities Association (PSA) using decades of historical data from the U.S. mortgage market. They serve as excellent proxies to determine loss for mortgage products and occasionally other long term assets.

    The most basic SDA curve is 100 percent SDA, which assumes an increase of .02 percent annual default in the first 29 months (starting with .02 percent),

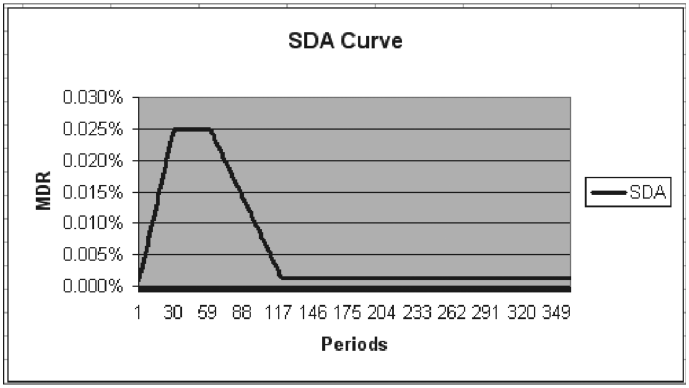


FIGURE 4.16 100 percent SDA displayed as a line graph.

a level .60 percent annual default for months 30 to 60, and then a decrease of .0095 percent annual default for months 61 to 120, and finally a level .03 percent for months 121 through 360. 100 percent SDA has a very recognizable shape in the mortgage industry when graphed as in Figure 4.16.

Multiplying or dividing the values of the 100 percent curve creates variations of the curve. So a 50 percent SDA curve contains half of the values for each period of the 100 percent curve, while a 200 percent SDA curve contains twice the values for each period of the 100 percent curve.

It is important to note that the values from these SDA curves are not the ones used in a monthly model. Remember that SDA uses a monthly default rate and the curve constructed above is created with annual rates. Use the following formula to convert from an annual default rate to a monthly default rate:

$$\text{Monthly Default Rate} = 100^{*}(1-(1-(\text{Annual Default Rate}/100))^{(1/12)})$$

The values for 50 percent, 100 percent, and 200 percent SDA are stored in the completed model in the Vectors sheet (AF7:AH366). Copy and paste these values into the same section of the model being created.

11. The next step that starts bringing the assumptions together is on the Cash Flow sheet. Go to the Cash Flow sheet and recall that columns M and N were created and labeled for default information. Cell M7 needs to contain the correct default rate for the asset pool depending on the selections from the Inputs sheet. Since there are two types of loss curves that can be used, user generated or SDA, the formula to determine the default rate will need to have a function that selects the correct curve based on the Inputs sheet.

One method to select the correct curve is to check to see if one of the Input sheet cells that select the curve is blank. If a user selected no value for one of the curves, the other curve must be the one being used. To make sure this is possible, blank cells were included in the ranges as part of the data validation lists. To make this clearer, the beginning of the formula that should be entered in M7 starts with an IF statement involving one of the curve assumptions on the Inputs sheet:

```
=IF(pdrLossTime1="",
```

The IF statement checks to see if the Timing Curve assumption is blank. If this is true, then an SDA curve must be in use; however, if it is false then a user-generated curve must be in use. It is important as a model operator to always keep one of the values blank, otherwise there can be confusion and formulas may not work as intended. The completed formula should appear as:

```
=IF(pdrLossTime1="",OFFSET(Vectors!$AE$6,A7+Age1,
   MATCH(SDA_Loss,lstSDA,0)),OFFSET(Vectors!$X$6,A7+Age1,
   MATCH(pdrLossTime1,lstDefaultCurve,0)))
```

OFFSET-MATCH combinations are used to look up the correct default rate from the Vectors sheet depending on the type of curve and the age-adjusted period. Copy and paste this cell down to M366.

12. The final cell to complete the default section is the dollar value calculation of the periodic defaults in N7. This appears to be a simple formula involving default rate multiplied by balance, but involves a couple of nuances depending on the type of default rate being used. A user-generated default curve is calculated against the original balance of each vintage. If such a curve is being used as the default assumption, then the default rate should be multiplied against the original balance of the assets. However, the SDA curve was originally calculated using the current balance to produce a monthly default rate. If an SDA curve is being used then the default rate should be multiplied against the current balance of the assets. The necessary toggle between the user-generated and SDA curve assumption is implemented with an IF statement similar to step 11:

```
IF(pdrLossTime1="",L7*M7,$L$7*M7)
```

If an SDA curve is being used then the current balance is multiplied by the default rate. If a user-generated curve is being used then the original balance is multiplied by the default rate. In the case of basing defaults against original balance, there is a chance that the calculated periodic default could be higher than the current principal balance. Since it is impossible to default more asset than exists, this needs to be cleaned up using a MIN formula:

```
=MIN(L7,IF(pdrLossTime1="",L7*M7,$L$7*M7))
```

Copy and paste this formula over the range N7:N366.

13. Two additional concepts on the Cash Flow sheet can be finished off once the default amount is known: actual amortization and actual interest. First, actual amortization is completed in cell R7. The proper amount of principal is calculated by taking the beginning balance of the assets less the defaulted amount and multiplying that by a ratio that represents the notional amortization for the period. Entering the following formula in R7 accomplishes this:

    =(L7−N7)*(1−O7/O6)

    Notice that the notional amortization ratio is the current period's amortization factor divided by the prior period's. Copy and paste this formula over the range R7:R366.
14. Finally this section is completed by entering formulas in two columns for the actual interest calculation. The first formula carriers over the interest rate from the notional schedule. Enter the following in cell S7 and copy and paste it over the range S7:S366:

    =F7

    Next enter the following formula in T7:

    =S7*C7*(L7−N7)

    This calculates the dollar amount of interest without the defaulted loans. Copy and paste this formula over the range T7:T366.

With loss calculations complete, it is important to take a step back and understand how the loss affects the model in general. Defaults are the principal balance of loans that are assumed to be losses. Since this principal amount is uncollectible, it should be removed from the asset pool balance, which is done in the Ending Balance formula in column V of the Cash Flow sheet. It is critical to realize that the asset pool balance is reduced each period by defaults. The actual amortization and interest are two concepts that are immediately and most noticeably affected by defaults. Many other concepts later in this text are affected by loss so its calculation and application should be well understood.

The importance of analyzing delinquency, default, and loss needs to be reiterated. Loss is a *core determinant* of the performance of a transaction and should be examined carefully. Such analysis is particularly critical for correctly sizing risk-rated tranches and understanding the risk of owning an asset that has the possibility of loss. While this chapter is an excellent introduction to these topics, asset specific characteristics could require additional analysis and unique methodologies.

Prior to moving on to the liability side of the model, there is one more section related to the assets: recoveries. After a default occurs, there is the possibility that the repossession and sale of the defaulted asset and the continued pursuance of credit collections will recover cash. This concept is the focus of Chapter 5.

CHAPTER 5

Recoveries

Although a defaulted asset generates a loss, there is often an opportunity to recover cash. Repossessing the asset and selling it is the primary method to achieve this recovery. The cash received from the sale flows into the structured transaction from which the asset came and is available for liabilities. While this process is very straightforward, a number of details need to be understood in order to model recoveries accurately.

The best approach to explain the nuances of recoveries is to first understand the relevant terminology. In Chapter 4, a gross loss was defined as an asset that is defaulted and assumed not to pay. Once that asset is repossessed and sold, the cash recovered can be subtracted from the original loss amount. The gross loss minus recovery is known as a net loss.

$$\text{Net loss} = \text{Gross loss amount} - \text{Recovery amount}$$

Related to net loss are two terms that are often confused: *loss severity* and *recovery rate*. These are actually inverse concepts. The recovery rate is the amount recovered divided by the gross loss amount. Loss severity is the ratio between net loss and gross loss amount. For instance, assume that an asset defaulted and created a $100 loss. If $80 is recovered from the sale of the asset then there is an 80 percent recovery rate or a 20 percent loss severity.

$$\text{Recovery rate} = \text{Recovery amount/Gross loss amount}$$

$$\text{Loss severity} = \text{Net loss/Gross loss amount}$$

A final term to define is *recovery lag*. A recovery does not take place immediately since certain legal proceedings need to occur, the asset physically needs to be repossessed, and the sale process executed. A recovery can take from a few months to a number of years before cash is actually realized. The length of time it takes from the default date to the recovery of actual cash is known as recovery lag.

An excellent method to understand recoveries is to take a loan from default to recovery and build a timeline of events. For this example, assume that a U.S. auto loan begins missing payments in May 2005. The loan continued to miss payments and was classified as a default in August 2005. The balance of the loan at that time was $1,580. At this point, from a transaction point of view, a default has occurred and a gross loss of $1,580 would be recorded for August 2005. In the same month, legal action to repossess the asset begins.

For auto loans in the United States, the legal action can proceed relatively quickly and a judgment can take place within a few months. Assume in this example that the servicer gets the repossession ruling in two months. Once a repossession ruling is in hand, the asset has to be physically seized and in the case of vehicles and equipment, transported to a liquidation or auction site. Depending on the type of asset, it can take some time to sell the asset and realize the cash. In this example assume that the asset is seized, auctioned, and sold for $850 in one month. See Figure 5.1 for a summary of events.

The above example is only partially complete because some costs need to be deducted. The most significant costs for nearly all assets are liquidation expenses. Hiring a company to repossess an asset costs money. So does physically moving the asset to a liquidation or auction location and having the asset liquidated. These costs can become very significant, particularly if assets are scattered internationally. Structured transactions often put restrictions on certain locations if the area is deemed to be difficult or costly to repossess in and liquidate.

Another cost that is not always included in the recovery assumption is the cost of carry. When a loan is considered defaulted there is an interest cost to the outstanding loan until the asset is liquidated and cash is received. This is sometimes not reported and not included in recovery assumptions. For the purposes of this example there will be no carry cost assumption.

Going back to the example, the cost of liquidation needs to be deducted from the recovery proceeds. In this case assume that liquidation costs $200. The net amount recovered is $650. The recovery rate in this case would be 41.13 percent ($650/$1580) otherwise quantifiable as the loss severity at 58.86 percent ($930/$1580).

While industries have developed typical recovery rates and lag times, a historical study of recoveries should be completed. This would involve aggregating the data that was mentioned above and coming up with average recovery amounts, liquidation

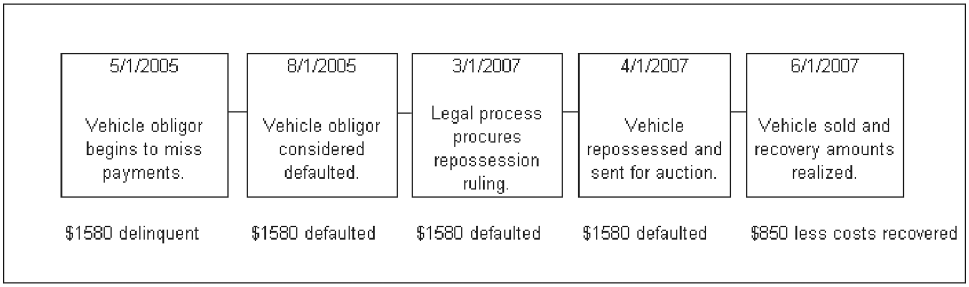

5/1/2005
Vehicle obligor begins to miss payments.
$1580 delinquent
8/1/2005
Vehicle obligor considered defaulted.
$1580 defaulted
3/1/2007
Legal process procures repossession ruling.
$1580 defaulted
4/1/2007
Vehicle repossessed and sent for auction.
$1580 defaulted
6/1/2007
Vehicle sold and recovery amounts realized.
$850 less costs recovered

FIGURE 5.1 A timeline for the recovery process of a typical defaulted auto loan.

costs, and lag times. This historical data analysis differs from those in Chapters 3 and 4 because it is more simplified. Recovery data is often difficult for issuers to accurately put together on a loan level basis. While that methodology would be ideal and should be used if available, a simpler approach is often sufficient.

MODEL BUILDER 5.1: HISTORICAL RECOVERY ANALYSIS

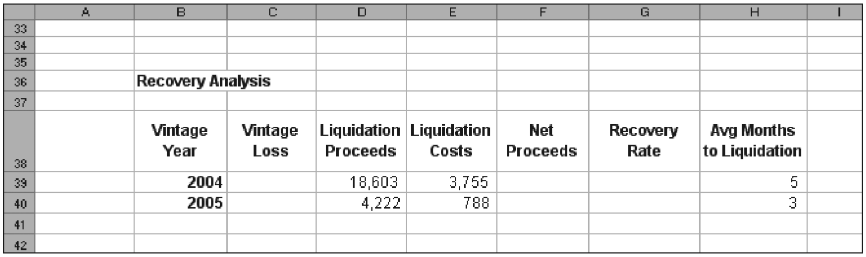
1. From the CD-ROM open the file*MB5-1_Raw_Data* from the Ch05 folder on the CD-ROM. Save this file as **Recovery Data.xls**. Notice that this data is the loss information from the previous chapter, with the addition of some incomplete recovery information below the loss data. See Figure 5.2 for more detail.
2. In this example, the recovery amounts have been given based on origination year. Look at cell D39. The 18,603 is the amount that has been recovered for loans originated in 2004. To make a comparison to the loss amounts, the aggregated loss for each year is necessary. Using the SUM function and starting in cell C33, sum up the loss for each vintage month.
3. The SUM function adds up the loss for each month, but the loss for each year is necessary to compare to the yearly recovery information. This can be achieved using a SUMIF formula. However, first a YEAR formula is necessary. In cell C34, use the YEAR function on cell C6. The formula should look like:

   = YEAR(C6)

   Copy and paste this formula over C34:AA34. This will give the year that each vintage is associated with and allow the SUMIF to be completed.
4. Go to cell C39 and create SUMIF for the total losses depending on the vintage year. The formula should read as follows:

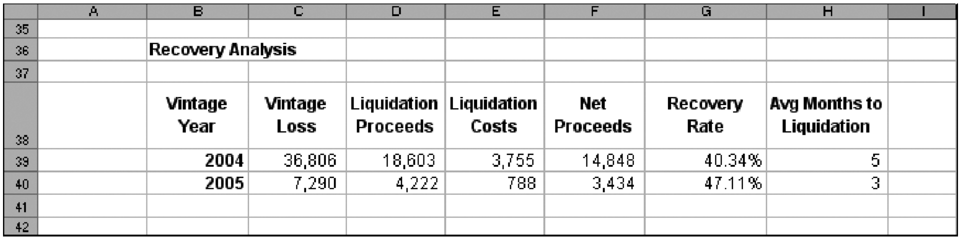
   = SUMIF($C$34:$AA$34,B39,$C$33:$AA$33)

   Copy and paste this formula in cell C40 to get the total for 2006.



	A	B	C	D	E	F	G	H	I
33									
34									
35									
36		Recovery Analysis							
37									
38		Vintage Year	Vintage Loss	Liquidation Proceeds	Liquidation Costs	Net Proceeds	Recovery Rate	Avg Months to Liquidation	
39		2004		18,603	3,755			5	
40		2005		4,222	788			3	
41									
42									

FIGURE 5.2 In addition to the loss data a section on recovery information is contained on the sheet.



	A	B	C	D	E	F	G	H	I
35									
36		Recovery Analysis							
37									
38		Vintage Year	Vintage Loss	Liquidation Proceeds	Liquidation Costs	Net Proceeds	Recovery Rate	Avg Months to Liquidation	
39		2004	36,806	18,603	3,755	14,848	40.34%	5	
40		2005	7,290	4,222	788	3,434	47.11%	3	
41									
42									

FIGURE 5.3 The recovery analysis is shown completed.

5. Liquidation proceeds and costs have been provided in the raw data. Typically this may require data manipulation to aggregate, but for now assume that it is presented in this format. The next step is to calculate the net proceeds from liquidation. This is a simple subtraction formula. In cell F39, enter the following formula:

$$= D39 - E39$$

   Copy and paste this formula down to cell F40.
6. The next step is to calculate the recovery rate by dividing the net proceeds by the amount that was considered lost. This is done in cell G39 using the following formula:

$$= F39/C39$$

   Copy and paste this formula down to cell G40.
7. The final part of the analysis is to notice the recovery lag. This is provided in H39 and H40. Typically the lag should be reported by the servicer or can be calculated by a more granular analysis. Refer to Figure 5.3 for a look at the completed section.

   The takeaway from this analysis is the recovery rate and lag. In this case, there are only two years of data, so it is questionable to use averages. With limited data, it is more conservative to take the lowest recorded recovery and the longest lag. In this example that would be 40.34 percent and five months.

PROJECTING RECOVERIES IN A CASH FLOW MODEL

Since recoveries are not a guaranteed cash flow, they are often looked upon with skepticism among risk adverse analysts. Even with data some analysts don't give credit for recoveries. However, when there is a recovery assumption it is important to implement the cash flow correctly. This includes achieving the correct timing of cash flow depending on lag and directing the cash in a manner consistent with the transaction structure.

MODEL BUILDER 5.2: INTEGRATING RECOVERIES INTO PROJECT MODEL BUILDER

1. Start at the Inputs sheet in cell J17 and label that cell **Recovery Rate**. Label cell K17 **Recovery Lag**. In cell J18 enter **40%** as an approximation of the recovery rate. Name this cell **pdrRecovRate1**. In cell K18, enter **5** for the recovery lag. Name this cell **pdrRecovLag1**.
2. Switch over to the Cash Flow sheet. Earlier Column U was designated as the area for recoveries. The formula for the recovery amount has to take into account the recovery rate and the lag. This can be achieved using an IF statement in conjunction with an OFFSET function. An IF statement is required first to make sure there is no calculation prior to the first possible recovery period. Cell U7 should start as follows:

   = IF(A7 <= pdrRecovLag1,0

   Without such a statement, there would be an error in the periods prior to the lag period when the OFFSET is implemented. With the IF statement in place, the OFFSET needs to reference the defaulted amount that occurred in the past. The periods back should be the recovery lag. Once that defaulted amount is determined, it should then be multiplied by the recovery rate to get the recovery amount. The following formula does exactly this:

   = IF(A7 <= pdrRecovLag1,0,OFFSET(N7,-pdrRecovLag1,0)*pdrRecovRate1)

   The OFFSET starts with the current period's defaulted amount (cell N7) and goes back in time using the recovery lag. The lag variable has a negative in front of it so it seeks a reference cell that is back in time, versus forward in time if it was positive. Finally, the defaulted amount is multiplied by the recovery rate to get the recovery amount. Copy this cell down the range U7:U366.

   Occasionally, the question of seasoned loans brings up a debate in recoveries when using the above formula. The debate is that by using this formula one assumes that no loans default prior to the start of the transaction. Some will argue that loans could have defaulted in the month prior to the transaction and if they were in the pool their recovery amount would come in earlier than the IF statement allows. The deciding point on this debate is that most structured transactions do not include defaulted loans in the pool of assets. This would mean that all loans going into the pool are current and the IF statement assumption holds, even with seasoned loans.
3. At this point the recovery section is complete, but to finish the entire Asset Amortization section one more column on the Cash Flow sheet needs to be created. Column X is the final column and is where all the cash flow that is generated each period will be aggregated. Label cell X4 **Total Cash Flow Available for Liabilities**. In cell X7 enter the following formula and copy and paste it over the range X7:X366:

   = Q7+R7+T7+U7

Scheduled Principal, Prepaid Principal, Interest, and Recoveries are real cash flows in each month and are what are available to the waterfall of liabilities. An important point about recoveries is that they are included as cash flow for the waterfall, but do not amortize the asset principal. This means that the cash is similar to interest in that it helps create excess spread.

FINAL POINTS REGARDING RECOVERIES

The reason the recovery analysis is not as detailed as prepayments or defaults is that recovery values can be very volatile depending on asset type, geographic region, and general economic environment. More advanced methods exist for assets, such as *market value decline* (MVD), which focuses on home values depending on specific regions or *Automotive Lease Guide* (ALG) values for estimating lease residual value, but these are all advanced analyses that are very dependent on highly detailed research in a specific industry.

Also, structured transactions are traditionally based on cash flow from the assets, not recoveries from defaults. As competition increases, newer deals have been giving more and more credit to recovery values, which can be dangerous if not properly analyzed. These types of deals begin to look like transactions that should be done by *nonperforming loan* (NPL) *groups*. Structured professionals seeking to do such deals should understand the recovery analyses performed by NPL groups, because they are often very comprehensive and detailed. In most instances, those analyses include loan level recovery and specific collateral analysis. For general modeling purposes or when first starting out, a more basic, conservative approach should be taken.

CHAPTER 6

Liabilities and the Cash Flow Waterfall

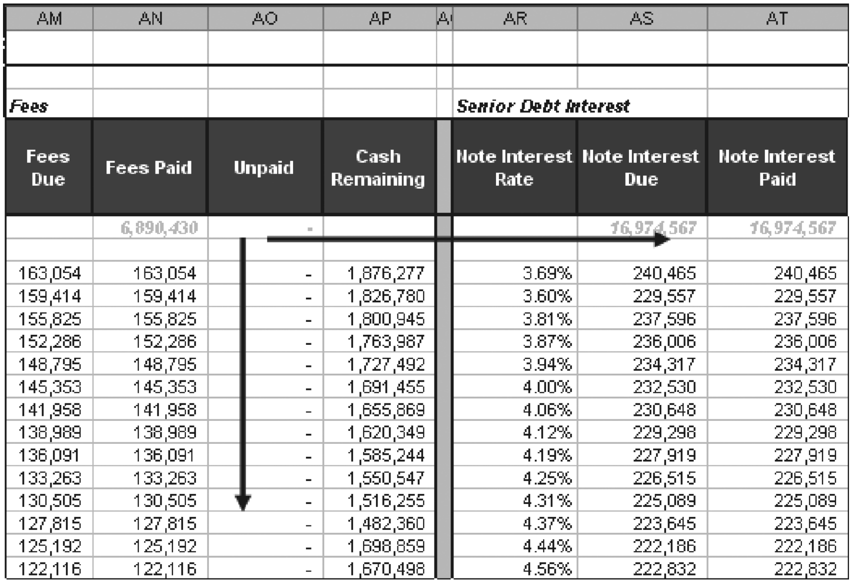
With asset generation complete, this book now turns to the liability side of the model. While the asset side is fairly standard for a level payment amortization, the liability section can vary greatly depending on the unique structure of the transaction. Due to the variability between deals, the liability side needs to be constructed with as much flexibility as possible. This is achieved by breaking down the liabilities into individual components that work similarly and can be moved around quickly.

Before going into the mechanics of how liabilities are paid, it should be understood what constitutes a liability. For a structured transaction, any cost that is to be paid from the cash generated by the assets is a liability. Foremost are costs, which keep the transaction in existence such as trust, servicer, and rating agency fees. Next is interest and principal due to the parties who funded the assets. This section can be very complicated depending on the different risk classes of debt known as *tranches*. The method in which these tranches are returned principal can also vary between transactions and as will be seen in the next chapter, within a transaction.

PRIORITY OF PAYMENTS AND THE CASH FLOW WATERFALL

In actual transactions, how the liability structure functions is dictated by the priority of payments section of a term sheet. This section is often referred to as a *waterfall* because the available cash starts at the top and then trickles down through an ordered series of liabilities. Each section in the written waterfall explains exactly how the cash should be allocated before moving onto the next section of the waterfall.

Conceptually, the movement of cash through the waterfall transfers well to two-dimensional spreadsheet-based modeling because time and order can be organized using rows and columns. The asset side only used rows for time progression, meaning that a starting balance would begin on row 1 and the next period the balance would be adjusted for amortization and displayed on row 2. Although asset interest, prepayments, defaults, and regular amortization do have a logical order, the vertical column arrangement does not necessarily suggest an organization based



AM	AN	AO	AP	A	AR	AS	AT
Fees					*Senior Debt Interest*		
Fees Due	**Fees Paid**	**Unpaid**	**Cash Remaining**		**Note Interest Rate**	**Note Interest Due**	**Note Interest Paid**
	6,890,430	-				16,974,567	16,974,567
163,054	163,054	-	1,876,277		3.69%	240,465	240,465
159,414	159,414	-	1,826,780		3.60%	229,557	229,557
155,825	155,825	-	1,800,945		3.81%	237,596	237,596
152,286	152,286	-	1,763,987		3.87%	236,006	236,006
148,795	148,795	-	1,727,492		3.94%	234,317	234,317
145,353	145,353	-	1,691,455		4.00%	232,530	232,530
141,958	141,958	-	1,655,869		4.06%	230,648	230,648
138,989	138,989	-	1,620,349		4.12%	229,298	229,298
136,091	136,091	-	1,585,244		4.19%	227,919	227,919
133,263	133,263	-	1,550,547		4.25%	226,515	226,515
130,505	130,505	-	1,516,255		4.31%	225,089	225,089
127,815	127,815	-	1,482,360		4.37%	223,645	223,645
125,192	125,192	-	1,698,859		4.44%	222,186	222,186
122,116	122,116	-	1,670,498		4.56%	222,832	222,832

FIGURE 6.1 Cash moves through each period from left to right and then down to the next period.

on time or priority. For the liability side of the model and the cash flow waterfall that is created on a spreadsheet, the periods still progress by rows; but the items in columns are arranged by timing. For example, on the Cash Flow sheet starting from the left after asset amortization, where cash is available, the liabilities are organized in priority as one progresses through the columns to the right. Each period, the cash moves from left to right through the waterfall before progressing to the next period one row below. See Figure 6.1 for more detail.

The Movement of Cash for an Individual Liability

A standard liability has an assumption such as a rate, fixed amount, or a vector of rates on the Inputs or Vectors sheets. This assumption is then integrated into the Cash Flow sheet similar to assets. However, the difference is that each liability will have a certain priority and can either be paid or not paid depending on how much cash is available. In such a system, the concept of "What You Have and What You Need" is essential to understand and make clear.

Barring extremely high default scenarios, every period a certain amount of cash should be available to pay liabilities. The first liability in the priority of payments will have all of the cash that is available from the assets for payment. Once that liability is paid, the cash available is appropriately reduced and the next liability can use the remaining amount for payment. This continues until the waterfall is done and either all of the cash for the period is used or excess remains and is released or trapped.

Each individual liability needs to be set up in a way that shows exactly what is needed, what can be paid based on what is available, what is unpaid, and what (if any) amount remains. While all of these calculations can be done in one cell it is easier for a model operator to see the ‘‘movement’’ of cash by separating each one of these concepts into different cells.

The concept of ‘‘What You Have and What You Need’’ is demonstrated in the Model Builder exercises for this chapter, which are inserted between explanations of the different types of liabilities. Both trends and unique nuances will become evident as one progresses through each type of liability.

TYPES OF LIABILITIES

A liability is anything that uses cash in the model. While there can be many different types of liabilities the three basic ones that will be examined are fees, interest, and principal.

Fees

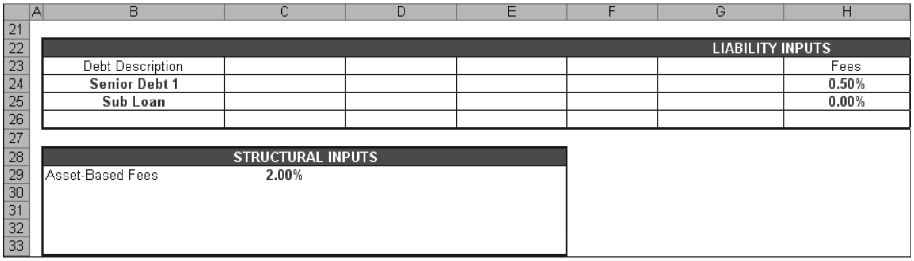
Every structured transaction has some type of fees involved. *Servicers* charge a fee to service loans. *Trusts* charge for taking in cash and distributing it correctly, rating agencies charge for performing analyses and assigning ratings. *Banks* charge for almost anything they do. Also, as will be seen in the later chapters, there are advanced features such as reserve accounts and hedging mechanisms that can incur fees.

Knowing how to calculate the periodic fee can be tricky sometimes because it can vary from deal to deal. Whole dollar amount fees are simple because they are a fixed amount charged per period. However, in a transaction where assets and liabilities taper off over time it is rare to see many whole dollar amount fees. Typically fees can be charged as a percent of the asset or debt balance. Determining the correct basis for a fee is important because a percentage charged against the asset balance versus a debt balance will result in different amounts. As mentioned earlier the debt is normally split up into different tranches, which are a percentage of the assets. A fee charged off of a senior debt balance that is 90 percent of the assets will be lower than a fee charged directly off of the assets.

MODEL BUILDER 6.1: CALCULATING FEES IN THE WATERFALL

1. Before any specific work can be done on the fees, a liability section needs to be created on the Inputs sheet. On that sheet, merge cells B22:O22 and enter **Liability Inputs** as the label for that cell. Row 23 will be where the labels are stored for the liability assumptions.

2. Two liability input sections need to be created for this Model Builder exercise. The first section establishes a description for the different types of debt. Project Model Builder uses a *senior-subordinated* structure, which is explained later in this chapter, but requires two tranches of debt. Enter the assumption label **Debt Description** in cell B23. Enter the label **Senior Debt 1** in cell B24 and name it **LiabDes1**. Also enter the label **Sub Loan** in cell B25 and name it **LiabDes2**. Rows 24 and 25 will contain liability assumptions that correspond to the respective tranche.
3. Project Model Builder has three possible fee inputs. There is one input for each debt tranche and one input that is based on the assets. This demonstrates the difference between calculating a fee against the asset balance or a debt balance. Still on the Inputs sheet, enter the label **Fees** in cell H23. In cell H24 enter **0.50%** as a value and name the cell **LiabFees1**. In cell H25 enter **0.0%** as a value and name the cell **LiabFees2**. The first liability fee will pertain to the senior debt, while the second fee (which is zero in this case) applies to the subordinate debt.
4. Remain on the Inputs sheet, but jump over to cell B28. This section will be for Structural Assumptions that are part of the deal structure as a whole and are not functions of the debt tranches. Enter the label **Structural Inputs** in cell B28. Directly below, in cell B29, enter the label **Asset Based Fees**. To the right, in cell C29, enter the value **2.00%**. Name cell C29 **AssetFee**. At this point the updated part of the Inputs page should look like Figure 6.2.
5. Switch over to the Cash Flow sheet. The last Model Builder exercise left off calculating the total cash available in Column X. Columns Y through AL is skipped for now because they are used later in the next chapter, when the advanced liability structure is introduced. The Fees section starts in column AM. Enter the following labels in their corresponding cells:
   AM4: **Fees Due**
   AN4: **Fees Paid**
   AO4: **Unpaid**
   AP4: **Cash Remaining**
6. The first period calculation for Fees Due begins in cell AM7. There are three distinct fees that need to be paid: the (1) Senior Debt fee, the (2) Subordinate Debt fee, and the (3) Asset Based fee. The Asset Based Fee will be the input rate



	A	B	C	D	E	F	G	H
21								
22							LIABILITY INPUTS	
23		Debt Description						Fees
24		Senior Debt 1						0.50%
25		Sub Loan						0.00%
26								
27								
28			STRUCTURAL INPUTS					
29		Asset-Based Fees	2.00%					
30								
31								
32								
33								

FIGURE 6.2 The fee section of the Inputs sheet.

multiplied against the assets' current balance. Remember that the fees typically quoted are annual rates, so the fee needs to be converted to a periodic rate. The Asset Based Fee is calculated by entering the following formula into cell AM7:

= (C7*AssetFee*L7)

This formula multiplies the Asset Based Fee assumption by the current asset balance and also by the periodic factor.

7. The other two fees that are due will be added to this formula. The calculation is similar, but there are noticeable differences. Modify the formula in cell AM7 to read as follows:

   = (C7*AssetFee*L7) + (C7*LiabFees1*CB6) + (C7*LiabFees2*CF6)

   Notice that two more blocks of fees have been added. The first takes the senior debt's fee (LiabFees1) and multiplies it by the senior debt's current balance. The second takes the subordinate debt's fee (LiabFees2) and multiplies it by the subordinate debt's current balance. Both have also been multiplied by the periodic factor (cell C7). Copy and paste the complete formula over the range AM7:AM366.

   The value that appears is what is due for the fees, not what is actually paid. Comparing this value to the total cash available for liabilities is the beginning of the concept "What You Have and What You Need." In this case the amounts in column X are "What You Have" and the amounts in column AM are "What You Need."

8. Column AN is where the amount that is actually paid is calculated. A beginning modeler often runs into problems by subtracting what is due from what is available. While this logically makes sense, many problems are created when there isn't enough cash available and the amount paid becomes negative. Then IF statements are introduced and the entire formula becomes messy.

   A simple method to use is to always take the least of "What You Have and What You Need." This translates into a MIN formula for the cash available and the amount due. In cell AN7 enter:

   = MIN(X7, AM7)

   Using this technique will ensure that nothing more than what is available can be paid. Copy and paste this formula over the range AN7:AN366.

9. The previous formula only pays from what is available. In stress situations, there could be shortfalls of cash that need to be tracked. Subtracting the amount paid from the amount due will solve that problem. In cell AO7 enter:

   = AM7−AN7

   Copy and paste this formula over the range AO7:AO366.

10. Finally, to keep the cash "moving," a column needs to be used to populate the cash remaining. This calculation is the cash that was available prior to the liability minus what was paid from that cash. In cell AP7 enter:

$$= \text{X7}-\text{AN7}$$

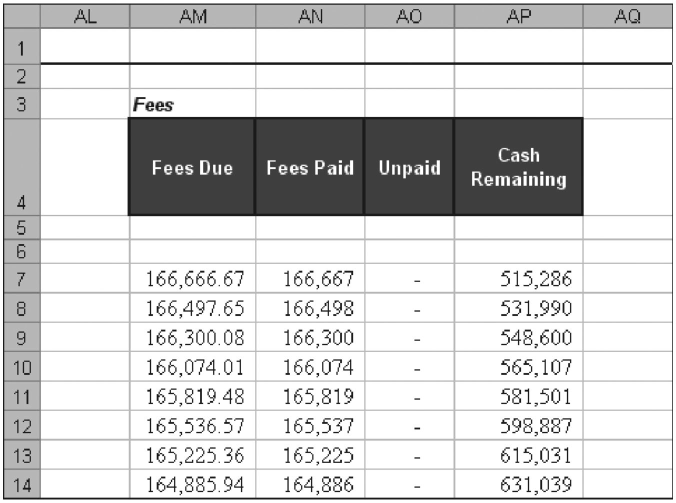
Copy and paste this formula over the range AP7:AP366.

At this point the fee section is complete and should look like the screen in Figure 6.3. Keep in mind the methodology that was developed in this section because any liability that needs to be added to a model can be done so in this fashion. See *MB6-1.xls* in the Ch06 folder on the CD-ROM for a completed example of these steps.

Interest

The primary purpose of an entity lending money in a transaction is to generate a return on capital, which is done by charging an interest rate against the money lent. For a private transaction that is not sold into the public markets, the interest rate usually is a bank's funding rate or swap rate plus a margin. If the deal is sold to investors, the rate will be the rate that investors earn on their principal.

Banks that lend money often do this on a floating rate basis using indexes such as LIBOR or prime as the base rate plus a margin. This means that the rates are



	AL	AM	AN	AO	AP	AQ
1						
2						
3		*Fees*				
4		Fees Due	Fees Paid	Unpaid	Cash Remaining	
5						
6						
7		166,666.67	166,667	-	515,286	
8		166,497.65	166,498	-	531,990	
9		166,300.08	166,300	-	548,600	
10		166,074.01	166,074	-	565,107	
11		165,819.48	165,819	-	581,501	
12		165,536.57	165,537	-	598,887	
13		165,225.36	165,225	-	615,031	
14		164,885.94	164,886	-	631,039	

FIGURE 6.3 The Fee section on the Cash Flow sheet is an introduction to the concept of moving cash and "What You Have vs. What You Need."

susceptible to market fluctuations and can change from period to period. If the assets are generating cash on a fixed rate basis, then there is the possibility of a mismatch between asset and liability rates. For instance, if the assets in a transaction are generating a weighted average rate of 7 percent and LIBOR skyrockets to 10 percent, then a bank that is funding based on LIBOR will lose money. Hedging instruments, for this reason, such as caps and swaps are used. Here the liability side should use the swap rate. Hedging instruments in structured transactions are discussed in more detail later in this book.

The other option to bank funding is investors who typically lend their money by buying bonds that are sold through an investment bank. These bonds are set at a disclosed interest rate that is normally fixed, but can also be floating. With a bond deal a swap is less likely to be needed because the rate can be fixed or set against a floating index.

In both cases, the debt rate is commensurate to the level of risk that the bank or investor is taking. A deal is determined to have a certain level of risk based on the expected loss of the transaction. The rate paid to debt in a deal should be proportionate to the risk of the transaction. However, structured transactions are specifically set up to mitigate and parse risk, so there can be different rates within a transaction.

Managing the risk introduces a concept called *credit enhancement*, which protects an investor against loss. Credit enhancement can take many forms such as excess interest being generated by the assets, reserve accounts, and/or subordinated debt or equity. Depending on the amount of credit enhancement a transaction will be able to withstand a certain amount of loss. Rating agencies have set certain standards for each asset class to determine a risk rating for a transaction, depending on how the structure holds up to certain stresses. This will be covered later in the text, but for now it is important to understand that the rate debt earns depends on how vulnerable the debt is to loss in the transaction.

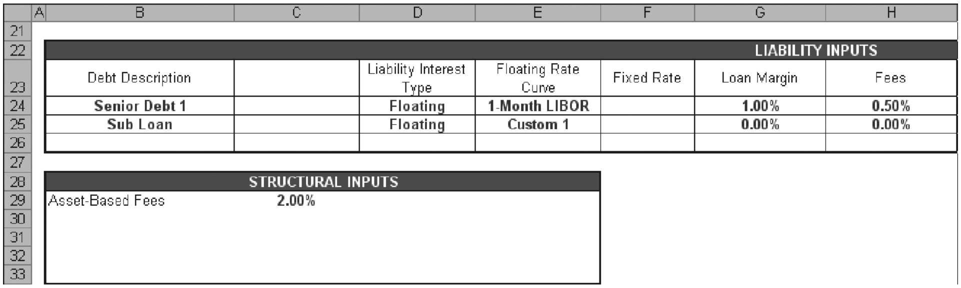
MODEL BUILDER 6.2: CALCULATING INTEREST IN THE WATERFALL

1. For each tranche of debt four inputs need to be known to calculate the correct amount of interest: whether the interest being charged is fixed or floating, the index if it is floating, the fixed rate if it is fixed, and finally the margin on top of either the floating or fixed rate. It may seem confusing to have a margin added to a fixed rate, but sometimes a fixed swap rate is used and a bank will charge a margin. Enter the labels for these four assumptions on the Inputs sheet in the following cells:
   D23: **Liability Interest Type**
   E23: **Floating Rate Curve**
   F23: **Fixed Rate**
   G23: **Loan Margin**

2. The Liability Interest Type for a given debt tranche can be one of three types:
   - *Floating*, when an index is used.
   - *Fixed*, either a fixed bond or swap rate.
   - *Custom*, a rate that changes over time based on the user or a third party. Custom is also used because rating agencies often derive and make available their own stressed interest rate curves.

   In the chapter on asset amortization, the range named lstIntType was already created for these three types of interest. Go to the Inputs sheet and create a data validation list in cell D24 using lstIntType as the range. Name cell D24 **LiabIntType1**. Repeat this in cell D25, but name that cell **LiabIntType2**.
3. Cells E24 and E25 need to produce the name of the curve that is being used if the debt is floating rate. This will be done exactly as was done for the interest rates on the asset side. Create a data validation list in cell E24 using lstInterestRates as the range. Name cell E24 **LiabLoanIndex1**. Repeat this in cell E25, but name that cell **LiabLoanIndex2**.
4. The fixed rates will be stored in cells F24 and F25 for each tranche. Name cell F24 **LiabFxdRate1** and F25 **LiabFxdRate2**. Keep both of these cells empty for now.
5. The loan margin is the final input that is needed for this section. Name cell G24 **LiabMarg1** and G25 **LiabMarg2**. Enter **1.00%** as a value for cell G24 and **0.00%** for cell G25. The Inputs sheet should now look like the screen in Figure 6.4.
6. The next step is to switch over to the Cash Flow sheet and calculate the debt interest. A standard structured transaction will pay the most senior debt interest first. In Project Model Builder columns AR:AX will be used for the Senior Debt Interest. Column AQ will be left blank as a separator.

   The first piece of information needed is the correct annual interest rate for the period. Enter the label **Note Interest Rate** in cell AR4. While step 2 created the possibility of three types of interest rates, there are actually only two options for storing the rates: on the Inputs sheet or on the Vectors sheet. This is important to know because a formula needs to know when and where to look for a specific data point. The simplest situation for rates is when a fixed rate is used, which is stored on the Inputs sheet. To account for that possibility, begin the formula in



	A	B	C	D	E	F	G	H
21								
22							LIABILITY INPUTS	
23		Debt Description		Liability Interest Type	Floating Rate Curve	Fixed Rate	Loan Margin	Fees
24		**Senior Debt 1**		**Floating**	**1-Month LIBOR**		**1.00%**	**0.50%**
25		**Sub Loan**		**Floating**	**Custom 1**		**0.00%**	**0.00%**
26								
27								
28			STRUCTURAL INPUTS					
29		Asset-Based Fees	**2.00%**					
30								
31								
32								
33								

FIGURE 6.4 The liability interest rate section complete on the Inputs sheet.

cell AR7 as follows:

```
= IF(LiabIntType1="Fixed",LiabFxdRate1
```

This part of the formula checks to see if a fixed rate is selected. If it is, the interest rate should be the fixed rate on the Inputs sheet. However, if the rate is floating or custom it will be stored in a curve on the Vectors sheet. The correct vector to use will then be determined by the floating rate curve that is selected on the Inputs sheet. This requires an OFFSET-MATCH combination as seen before:

```
= IF(LiabIntType1="Fixed",LiabFxdRate1,OFFSET(Vectors!$D$6,
   Vectors!A7,MATCH(LiabLoanIndex1,lstInterestRates,0))
```

This addition to the formula will select the correct annual rate for the period on the Vectors sheet depending on the name of the curve that is selected in the range LiabLoanIndex1 on the Inputs sheet. Finally adding the margin, if one exists, completes the formula:

```
= IF(LiabIntType1="Fixed",LiabFxdRate1,OFFSET(Vectors!$D$6,
   Vectors!A7,MATCH(LiabLoanIndex1,lstInterestRates,0))+LiabMarg1)
```

Copy this formula over the range AR7:AR366.

7. The interest due is easy to calculate once the annual interest rate for the period is known. However, no principal balance information is known until Model Builder 6.3, so an actual value will not show up. A proxy value will be used until that section is complete. Still on the Cash Flow sheet skip over to cell CB6 and enter the value 95,000,000. This is the assumed starting principal balance of the senior debt for now.

   Go back to cell AS4 and enter the label **Note Interest Due**. In cell AS7 enter the following formula:

   ```
   = C7*AR7*CB6
   ```

   This formula takes the annual interest rate for the period, converts it to a periodic interest rate and then multiplies that value against the prior period's ending principal balance. It is important to understand the difference between end of period and beginning of period. In this model, the balance referenced is always one row back because that balance is an end of period balance. Always make sure that the balance an interest rate is being applied to is either the end balance for the prior period or the beginning balance of the current period. Copy this formula over the range AS7:AS366. Since there is no principal balance information yet, it is normal that the values for the cells below row 7 will be zero for most of the columns in this section. This will change after Model Builder 6.3.

8. In cell AT4 enter the label **Note Interest Paid**. This "paying" formula will be similar to the one used in the fee section. Remember the rule, "take the lesser of what is available and what is needed" and enter the following in cell AT7:

   = MIN(AP7, AS7)

   This formula takes the lesser of the cash remaining after fees have been paid and the note interest that is due for the period. Copy this formula over the range AT7:AT366.
9. Column AU will track the unpaid amounts. Enter the label **Unpaid** in cell AU4. In cell AU7 enter:

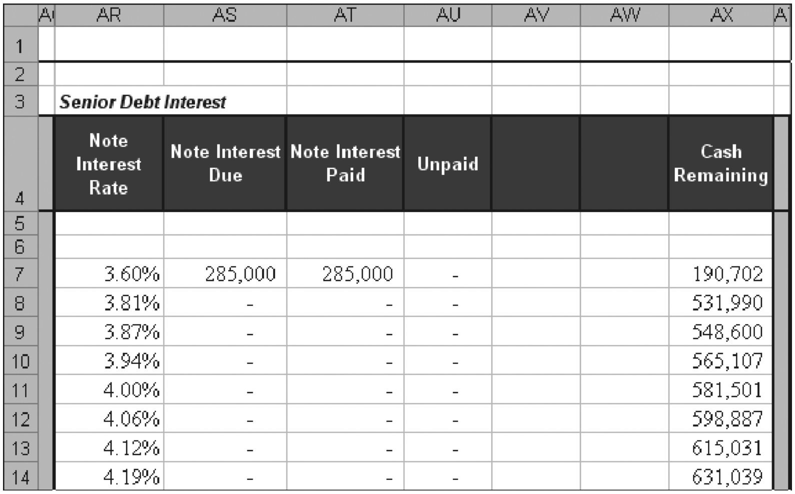
   = AS7−AT7

   As done before in the fee section the paid amount is subtracted from the due amount to determine the unpaid amount. Copy this formula over the range AU7:AU366.
10. Skip over columns AV and AW for now and leave them blank. Those will be reserved for an advanced liability structure in the next chapter. In cell AX4 enter the label **Cash Remaining** and in cell AX7 enter:

    = AP7−AT7

    This formula subtracts the amount paid to interest from the previous cash remaining. Copy this formula over the range AX7:AX366. The senior debt interest section should look like Figure 6.5.
11. The senior debt interest calculations are now complete, but the subordinated debt interest remains unfinished. Subordinated debt is usually lower in the waterfall than many other items, so it will appear further to the right on the Cash Flow sheet. Since Project Model Builder's waterfall is preplanned; the exact columns that the subordinated debt fits into are known. However, keep in mind that when creating a scratch model the final columns, where liabilities end up, may not be clear as it is being built and may require inserting or deleting columns.

    To complete the subordinated debt, stay on the Cash Flow sheet and move over to column BN. All of the formulas will be very similar to the senior debt interest formulas, so this section should be quick to make. Enter the following labels in their respective cells:
    BN4: **Sub Loan Rate**
    BO4: **Loan Interest Due**
    BP4: **Loan Interest Paid**
    BQ4: **Unpaid Interest**
    BR4: **Cash Remaining**



	AR	AS	AT	AU	AV	AW	AX
1							
2							
3	***Senior Debt Interest***						
4	Note Interest Rate	Note Interest Due	Note Interest Paid	Unpaid			Cash Remaining
5							
6							
7	3.60%	285,000	285,000	-			190,702
8	3.81%	-	-	-			531,990
9	3.87%	-	-	-			548,600
10	3.94%	-	-	-			565,107
11	4.00%	-	-	-			581,501
12	4.06%	-	-	-			598,887
13	4.12%	-	-	-			615,031
14	4.19%	-	-	-			631,039

FIGURE 6.5 The nearly complete senior interest section of the Cash Flow sheet.

12. As with the senior debt balance, a proxy value should be entered for the subordinate debt. Enter 5,000,000 in CF6. Next, enter the following formulas in the cells as noted:

    BN7: = IF(LiabIntType2 = "Fixed",LiabFxdRate2,OFFSET(Vectors!$D$6,
    Vectors!A7,MATCH(LiabLoanIndex2,lstInterestRates,0))
    +LiabMarg2)

    BO7: = C7*BN7*CF6

    BP7: = MIN(BL7,BO7)

    BQ7: = BO7 – BP7

    BR7: = BE7 – BP7

    Copy the range BN7:BR7 and paste it over the range BN7:BR366. Do not be concerned if many of the cells have zero values. The cash flow waterfall is being constructed using a conceptual methodology, not ordinal. This requires many blank and zero value cells until the entire waterfall is complete.

    Also, a final note on interest relates to unpaid amounts. The example model does not capitalize unpaid interest nor does it make the unpaid interest due the next period. Many transactions are structured this way and the modeling should reflect such details. Also, as with the other Model Builder sections, the Ch06 folder on the CD-ROM features a corresponding completed example.

Principal

In addition to interest, banks and investors expect to have the principal amount they loaned returned. Principal is often returned at different priority levels to mitigate and parse risk. Earlier it was briefly mentioned that there are different risk rated classes of debt—tranches. The way to think about the debt structure is that assets must always equal liabilities. Assets are not free and must be 100 percent funded from the start; however, an investor may not want to take 100 percent of the risk that the assets do not pay back all of the investors' loaned principal. Instead, a bank could sell bonds equal to 90 percent of the assets as senior debt and the other 10 percent as subordinated debt. The reason the first 90 percent is considered to be senior is because it has priority to its principal over the subordinated debt in the cash flow waterfall. Such a set up is also known as a senior subordinated structure.

Senior debt should always have priority when receiving principal versus subordinated debt; but there are two different methods of amortization that differ in regards to tranche principal repayment: *sequential* and *pro rata*. A sequential pay method pays the entire senior principal balance before paying one dollar of the subordinated debt. This means that it could be months or years in a deal until the subordinated debt receives a principal payment. It also makes the senior debt more secure because the subordinate debt does not decrease and, as discussed later, is a source of credit enhancement for the senior debt.

The other type of principal payment methodology is pro rata, which as the name implies pays principal proportionately. A simple example is if there is $100 of assets, funded by a senior loan of $90 and a subordinate loan of $10. The proportion of the debt is 90 percent senior loan and 10 percent subordinate loan. If $5 of principal came in during a period then the senior loan would be due $4.50 ($5*90 percent) and the subordinate loan due $.50 ($5*10 percent). While fixed during normal performance, these proportions can change within a deal if the assets are incurring unexpectedly high levels of default.

However, a change to principal allocation within a deal is a more advanced concept that is discussed in further detail later in the book. At this point, the focus is to understand the basic flow of principal through the cash flow waterfall. Project Model Builder uses a senior-subordinated debt structure with the option for either sequential or pro rata principal payment. The model is also set up with the concept that principal is "passed through" to the debt. This means that if the transaction is performing as expected any amortization of the assets should directly result in the same amortization of the debt.

MODEL BUILDER 6.3: CALCULATING PRINCIPAL IN THE WATERFALL

1. Go to the Inputs sheet and enter the label **Advance Rate** in cell C23. The advance rate is the debt principal amount expressed as a percentage of the assets. If there is $100 of debt and the senior debt is $95 at day one, then the advance rate for

the senior debt is 95 percent. In cell C24 enter 95.00 percent. Name cell C24 **LiabAdvRate1**. Since there is only one other tranche of debt the subordinate amount advanced will always be 100 percent minus the senior advance rate. Enter the following formula in cell C25:

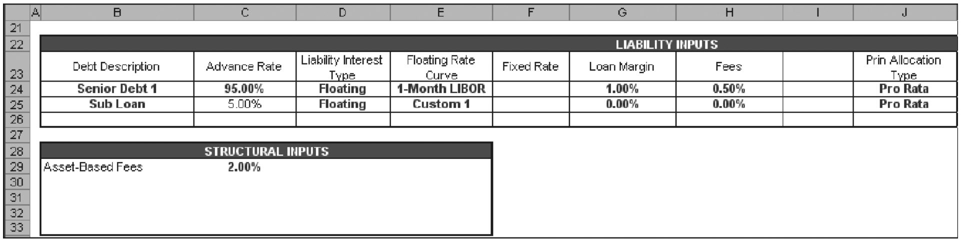
= 1−LiabAdvRate1

Name cell C25 **LiabAdvRate2**.

2. The other necessary input is the principal payment or allocation type. There are only two types discussed so a data validation list works well. Go to the Hidden sheet and enter the label **PrinType** in cell A21. Enter **Sequential** in cell A22 and **Pro rata** in A23. Name the range A22:A23 **lstPrinType**. Go back to the Inputs sheet and enter the label **Prin Allocation Type** in cell J23. Create data validation lists in cells J24 and J25 using lstPrinType as the range. Name cell J24 **LiabPrinType1** and J25 **LiabPrinType2**. So far the section should look like Figure 6.6.
3. Now is the time to change the proxy values for the principal balances that were created in Model Builder 6.2. Go to the Cash Flow sheet and label the following cells:
   CB4: **Senior Loan EOP Balance**
   CC4: **Senior Interest**
   CD4: **Senior Principal**
   The initial senior principal balance will be the advance rate multiplied by the initial asset balance. After the first period the balance will be reduced commensurate to the assets. Since there are two possible states, initial period and after, an IF statement formula is needed. In cell CB6 enter:

   = IF(A6=0,V6*LiabAdvRate1,CB5−CD6)

   This formula checks to see if the period is the initial period, multiplies the advance rate by the asset balance if it is the initial period, or subtracts the current principal payment from the prior period's balance if the period is anything else than 0. Copy this formula over the range CB6:CB366.



	A	B	C	D	E	F	G	H	I	J
21										
22		LIABILITY INPUTS								
23		Debt Description	Advance Rate	Liability Interest Type	Floating Rate Curve	Fixed Rate	Loan Margin	Fees		Prin Allocation Type
24		**Senior Debt 1**	**95.00%**	**Floating**	**1-Month LIBOR**		**1.00%**	**0.50%**		**Pro Rata**
25		**Sub Loan**	5.00%	**Floating**	**Custom 1**		**0.00%**	**0.00%**		**Pro Rata**
26										
27										
28		STRUCTURAL INPUTS								
29		Asset-Based Fees	**2.00%**							
30										
31										
32										
33										

FIGURE 6.6 The principal section of the liabilities on the Inputs sheet.

4. The same should be done for the Sub Loan. Enter these labels in the following cells:
   CF4: **Sub Loan EOP Balance**
   CG4: **Sub Interest**
   CH4: **Sub Principal**
   The only difference in the formula are the references and that LiabAdvRate2 is used instead of LiabAdvRate1. Cell CF6 should have the following formula:

   = IF(A6= 0,V6*LiabAdvRate2,CF5−CH6)

   Copy this formula over the range CF6:CF366. Leave rows 7:366 for columns CC, CD, CG, and CH empty for now.
5. Next the senior debt principal amounts will be calculated in their correct place in the waterfall. Typically senior principal pays after senior interest. Still on the Cash Flow sheet, enter these labels in the following cells:
   AZ4: **Principal Due**
   BA4: **Principal Paid**
   BB4: **Unpaid**
   BE4: **Cash Remaining**
   Leave columns BC and BD blank for now.
6. For now the senior tranche principal due can be either sequential, where all of the asset amortization is due to the senior tranche first or pro rata, where the senior tranche's proportional share of the asset amortization is due. Enter the following formula in cell AZ7:

   = IF(LiabPrinType1="Sequential",MIN((N7+Q7+R7),CB6)

   Quite a bit is taking place in this formula. First notice that an IF statement is used to check what type of principal allocation method is being used for the senior debt. If a sequential method is used, then the principal due to the senior notes will be the amount that the assets amortized in that period. That amount consists of scheduled amortization, voluntary prepayments, and new defaults (columns R + Q + N respectively).

   A point of confusion that comes to even a seasoned structured finance professional is why new defaults are included in the debt principal due calculation. This is the heart of one of the forms of risk mitigation: using excess cash or spread to cover loss. Since the assets have been reduced by the defaults, the debt will need to be reduced by the same amount. The debt of course has access to prepayments and scheduled amortization, which provide cash to the waterfall, but defaults are noncash generating amounts. If there were no other cash in the deal besides the prepayments and scheduled amortization, then the debt principal due could never be paid because the defaulted amount would make the debt principal due calculation too high.

   However, most transactions are structured so the assets generate more interest than is due to fees and debt interest. This concept is officially known as *excess* *spread*. That extra money will trickle down along the waterfall and eventually be available to pay principal due. Since the defaulted amount is built into the debt principal due calculation, that extra amount can be paid if there is excess spread in the transaction. This is how excess spread is typically used to first cover losses. If there is no excess spread then other sources of credit enhancement are necessary to cover the defaulted amount, which will be seen in the next chapter. Going back to the formula, also notice that there is a MIN function for the asset amortization amounts and the debt's prior period ending balance. The MIN function prevents the principal due from exceeding the balance of the debt. This typically occurs in the final period when the debt balance is small and possibly lower than the amount the assets amortized.

7. Complete the formula in AZ7 by adding the following shown in bold:

   = IF(LiabPrinType1="Sequential",MIN((N7+Q7+R7),CB6),
   MIN((N7+Q7+R7)*LiabAdvRate1,CB6))

   This addition to the formula is for a pro rata principal allocation method. Instead of using the entire asset amortization amount for the period, the formula takes a percentage of the asset amortization amount. The MIN function is also used here to cap the principal due by the prior period's debt balance. Copy and paste the complete formula over the range AZ7:AZ366.
8. The remaining calculations revert back to the concept of "What You Have and What You Need." For BA7 enter the familiar MIN formula:

   = MIN(AX7,AZ7)

   This will take the lesser of the cash remaining after interest was paid and the amount due for principal. Copy and paste the formula over BA7:BA366.
9. In cell BB7 enter:

   = AZ7−BA7

   This subtracts the principal paid from the principal due and displays any unpaid amounts. Copy and paste the formula over range BB7:BB366.
10. Enter the following formula in BE7 to determine the cash remaining:

    = AX7−BA7

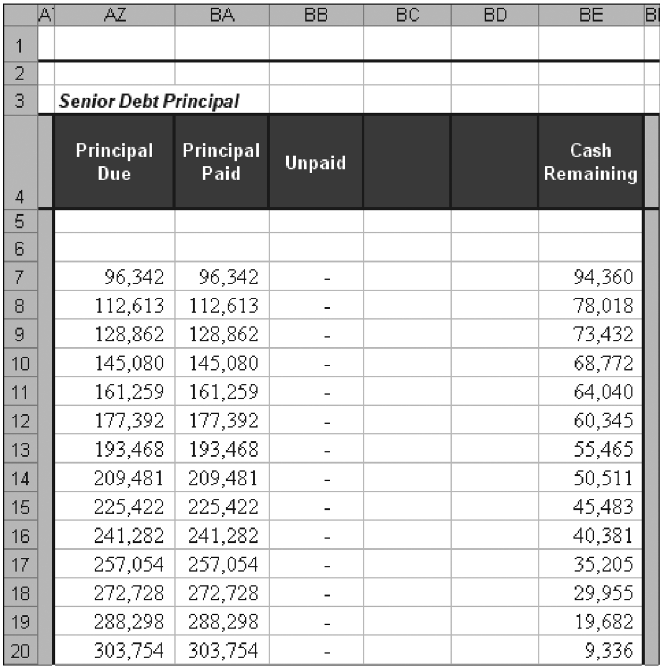
    Copy and paste the formula over BE7:BE366. The screen should now look like Figure 6.7.
11. Still on the Cash Flow sheet, move across to column BT. Add the following labels
    BT4: **Loan Principal Due**
    BU4: **Loan Principal Paid**

BV4: **Unpaid**
BW4: **Cash Remaining**
A similar debt principal calculation needs to be done for the subordinated debt; but there is a major difference. In a sequential principal allocation system, the subordinated tranche should not receive any principal until the senior tranche is completely paid off. Modeling this logic is achieved through an IF-AND combination in cell BT7:

```
= IF(AND(LiabPrinType2="Sequential",CB6>0),0,
```

The beginning of the formula for BT7 will return a 0 value if the principal allocation system is set to "Sequential" and the senior debt has a principal balance. If the senior debt is paid off, then another IF statement is required because a FALSE value for the first IF statement can be caused by either a different principal allocation type or a paid off senior tranche. Add the following shown in bold to cell BT7:

= IF(AND(LiabPrinType2="Sequential",CB6>0),0,
IF(LiabPrinType2="Sequential",MIN((N7+Q7+R7),CF6),



	AZ	BA	BB	BC	BD	BE
1						
2						
3	*Senior Debt Principal*					
4	Principal Due	Principal Paid	Unpaid			Cash Remaining
5						
6						
7	96,342	96,342	-			94,360
8	112,613	112,613	-			78,018
9	128,862	128,862	-			73,432
10	145,080	145,080	-			68,772
11	161,259	161,259	-			64,040
12	177,392	177,392	-			60,345
13	193,468	193,468	-			55,465
14	209,481	209,481	-			50,511
15	225,422	225,422	-			45,483
16	241,282	241,282	-			40,381
17	257,054	257,054	-			35,205
18	272,728	272,728	-			29,955
19	288,298	288,298	-			19,682
20	303,754	303,754	-			9,336

FIGURE 6.7 The Senior Principal section of the Cash Flow sheet.

Now if the principal allocation is set to "Sequential" and the senior tranche is paid off then the subordinate tranche will continue to receive 100% of the asset amortization. Finally, the last part (shown in bold) that needs to be added is when the principal allocation method is set to "Pro rata":

```
= IF(AND(LiabPrinType2="Sequential",CB6>0),
  0,IF(LiabPrinType2="Sequential",MIN((N7+Q7+R7),CF6),
  MIN((N7+Q7+R7)*LiabAdvRate2,CF6)))
```

Similar to the senior tranche, when the principal allocation method is pro rata then the subordinate tranche will only pay down by its proportionate share of the asset amortization. Copy and paste the complete formula over the range BT7:BT366.

12. The rest of the calculations should seem familiar by now—so in the following cells enter:
BU7: **= MIN(BR7,BT7)**
BV7: **= BT7-BU7**
BW7: **= BR7-BU7**
Copy the range BU7:BW7 and paste it over the range BU7:BW366.

13. With all of the interest and principal calculations in place, the debt balances can be completed. At this point many of the columns that have zero values will change to real values since the debt balances will extend over time.

In cell CC7, reference the senior interest that has been paid for the period by entering **= AT7**. In cell CD7, reference the senior principal that has been paid for the period by entering **= BA7**. Copy and paste CC7:CD7 over the range CC7:CD366. The same should be done for the subordinated debt. In cell CG7 enter **= BP7** and in CH7 enter **= BU7**. Copy and paste cells CG7:CH7 over the range CG7:CH366.

This completes the debt principal calculations. At this point, the debt principal should be decreasing as principal payments come in. In fact, the basic liability waterfall is complete. However, the waterfall is not operational because a few advanced structures are missing. Also keep in mind that this is one of many unique liability structures. To accurately model a transaction, the priority of payments needs to be thoroughly understood. Refer to *MB6-3.xls* in the Ch06 folder on the CD-ROM for a complete example of this section.

UNDERSTANDING BASIC ASSET AND LIABILITY INTERACTIONS

With the creation of the basic liability structure, the value of modeling a transaction begins to become clear. Assumptions can be made that replicate the structure and behavior of assets, which generate cash. The amount and timing of the cash depends on the assumptions for asset amortization, prepayments, defaults, and recoveries.

There are endless possibilities for the amount and timing of the available cash in the transactions. The same variability can be seen on the liability side in the cash flow waterfall.

Different liability structures can be put in place to work with any given pool of assets. Stress scenarios are then run to see how the liability structure handles defaulted assets. Up to this point, the only form of protection against loss or credit enhancement is excess spread. When excess spread is not enough to help pay the liabilities off by the final maturity of the transaction, then the debt holders will sustain a loss. To a lesser extent, there can be situations where interest is not completely paid and the debt holders receive less return than anticipated.

While excess spread in a transaction is an excellent source of protection, structured transactions have developed multiple methods of protecting against stressed situations. These methods add another level of complexity to a model, but must be incorporated to accurately model a transaction. The next chapter explains these advanced cash flow structures and shows how to incorporate them into a model.

CHAPTER 7

Advanced Liability Structures

Triggers, Interest Rate Swaps, and Reserve Accounts

Loss protection is the single most important reason for advanced liability structures. All entities that fund transactions are worried about loss and try to anticipate and protect against its different forms. As seen from Chapter 4, nonperforming assets that have stopped generating cash and are considered delinquent or defaulted cause loss. However, there can also be structural issues such as interest rate mismatches, which need some type of protection. Advanced liability structures such as triggers, swaps, and reserve accounts are created to help prevent and mitigate these concerns.

TRIGGERS AND THEIR AFFECT ON THE LIABILITY STRUCTURE

The simplest and most cost-effective method of mitigating loss is by altering the structure of the transaction when problems arise. If a deal is performing as expected, then the liability structure is probably sufficient to ensure that all parties are repaid. However, when assets begin to default, investors worry and become very cognizant of where they stand in the priority of payments. In many structured transactions, a senior investor will have negotiated a change in the priority of payments if the deal begins to perform very poorly. The change is usually caused by a predefined test, officially known as a *trigger*, being breached. This change directs more cash to the senior investor so that the senior obligation receives principal faster.

The speed at which an investor receives principal back is often a point of confusion. While having principal returned faster is more conservative, it is not necessarily more desirable. The faster principal is returned the faster the debt obligation is paid off. A faster paying obligation will have less overall yield than a slower paying one if assets are paying as intended and debt interest and principal can be paid. Also, paying the obligation back faster changes the weighted average life and could cause a mismatch in investment tenors for an investor. This is a problem because many times investors choose which transaction to invest in with maturity and weighted average life in mind.

The opposing duality of payment speed's risk and reward makes defining and setting a trigger very difficult. If a trigger is set up too tightly, then the trigger is breached quickly, the liability structure switches, and the investor receives principal faster than necessary. However, if a trigger is set up too loosely, then the trigger is not breached and, if there is a problem in the transaction, the investor has a higher amount of principal exposed for a longer amount of time.

A classic example of a trigger is one that is based on cumulative default rate. Imagine a set of assets that have a historical default rate of 3 percent. A structurer has decided that the transaction should have a trigger of 5 percent gross cumulative defaults, with the results of a breach being the rapid amortization of senior principal. If the assets perform as expected, historical defaults should remain around 3 percent and the senior investors get their return as expected. If defaults jump over 5 percent, then excess cash in the transaction is used to pay down the senior obligation first.

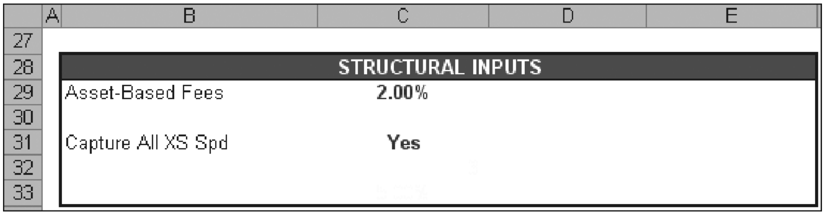
Other types of common triggers include:

- *Negative excess spread*. When excess spread, defined as the difference between the asset yield and the liability fees and interest, becomes negative this trigger is breached.
- *Delinquency*. When the delinquency rate for the assets breaches a predefined level.
- *Rolling average triggers*. These triggers take a common trigger like defaults, but use the average of a certain time period that "rolls" as time progresses, rather than just using one period as the test. This is useful because it prevents temporary spikes from changing a deal when the problem could be a single period anomaly.
- *Qualitative triggers*. There can be multiple nonquantitative triggers like missing a payment to the trust, failing to send in reports, and in general failing to meet a list of preexisting criteria.

Finally, when triggers are breached there can be many different consequences. If the trigger breached is not very severe, it could just mean trapping extra cash for a period. However, if a serious problem is occurring and a major trigger is breached the deal could then go into full rapid amortization and all cash could be redirected to senior investors. Also, triggers can be set up to cure. This means that if the trigger was breached in one period, but in the next period the metric for the trigger passes, then the state of the deal can go back to the prebreached set up. All of these nuances require a thorough understanding of triggers because they can have a very powerful impact on how a deal performs.

MODEL BUILDER 7.1: INCORPORATING TRIGGERS

1. Modeling triggers in a transaction does not necessarily mean setting up each one exactly as the documents read. Particularly in the case of qualitative triggers, this would be time consuming and most likely not worth the time since breaching any of those would be a complete guess. The only triggers that need to be modeled are ones that can be breached when the cash flow is stressed. Project



STRUCTURAL INPUTS

Asset-Based Fees	2.00%
Capture All XS Spd	Yes

FIGURE 7.1 The Capture trigger should be entered in the Structural Inputs section.

Model Builder will have four triggers to show common trigger analysis. The parameters for these triggers are located on the Inputs sheet.

Prior to entering any trigger related assumptions, a named list needs to be created on the Hidden sheet. Go to the Hidden sheet and enter the label **YesNo** in cell A25. Enter "Yes" in cell A26 and "No" in cell A27. Name the range A26:A27 **lstYesNo**.

2. Next, go to the Inputs sheet and enter the label **Capture All XS Spd** in cell B31. In cell C31, create a data validation list with lstYesNo as the range. Name cell C31 **GlobalTrigger**. This trigger is one that will be decided by the model operator. If a "Yes" value is input in cell C31, then all excess spread in the transaction will be used to pay down senior debt. The reason for such a trigger is that a worst-case scenario is often modeled. In such a case one would assume that the assets are performing very poorly and rapid amortization triggers have already been tripped. The Inputs sheet should look like Figure 7.1.
3. It is necessary to track each trigger on the Cash Flow sheet because when a trigger is breached the flow of cash in the waterfall will change. To track whether or not a trigger has been breached each period a Boolean statement (TRUE or FALSE) should be returned.

   Go to the Cash Flow sheet. Enter the label **Capture Trigger** in cell AA4. In AA7 enter the following formula:

   = IF(GlobalTrigger="Yes",TRUE,FALSE)

   This is a simple IF statement that directs a TRUE to be input in the cell if the range GlobalTrigger is set to "Yes" or FALSE if it is not. Copy this formula and paste it over the range AA7:AA366.

   At this point the next logical step may seem to set up switches in the cash flow for when the trigger is breached. However, since there are three more triggers to create, it will be more efficient to set up those assumptions prior to adjusting the cash flow formulas.
4. The next trigger is a more flexible version of the previous one. It allows the model operator to decide which period to begin a rapid amortization state. The reason this is useful is that as a transaction progresses under a normal state,

cash is typically released out of the transaction. Any cash released is cash that is not available for debt repayment.

A scenario that should be run, which will be discussed later in the text, is to release cash for a number of months prior to a rapid amortization event. Often triggers take a few periods to be breached, particularly in the case of default triggers where the definition of a default is three months delinquent. Such a trigger could never be breached in the first three months.

Go to the Inputs sheet and in cell B32 enter the following label, **Post-Default Trigger Month**. In cell C32 enter the value **3** for now. Name cell C32 **PostDefTriggerMo**.

5. Go to the Cash Flow sheet and in cell AB4 enter the label **Post Default Mo Trigger**. In cell AB7 enter the following formula:

   =IF(AND(A7>=PostDefTriggerMo,PostDefTriggerMo<>0),TRUE,FALSE)

   Deconstructing this formula reveals an AND statement that tests the current period against the value input for PostDefTriggerMo and if PostDefTriggerMo is not zero. This statement reads that if the current period in the cash flow is greater than or equal to the trigger assumption on the Inputs sheet, then the trigger has been breached, and a TRUE value should be returned. Otherwise the value is false.

   Notice that when a zero is entered as the assumption the formula will return a FALSE statement. This is so there is an option to always have the trigger off. Copy the formula and paste it over the range AB7:AB366.

6. The most complicated trigger will be one that tracks defaults. If the default percentage experienced in the deal breaches a predefined level set up on the Inputs sheet, then the trigger is tripped.

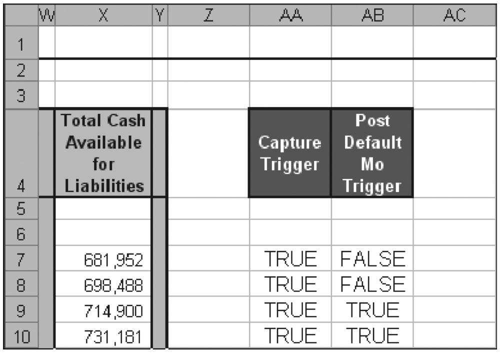
   Go to the Inputs sheet and enter the label **Default Trigger %** in cell B33. For now, enter **5.00%** in cell C33 and name that cell **Trigger_ Def**.

7. Go to the Cash Flow sheet, but go far to the right to column CP. Prior to setting up the actual trigger test a section needs to be created for tracking each period's gross cumulative default percentage. Tracking is typically done to the far right of the waterfall.

   Enter the label **Cumulative Default Percentage** in cell CP4. For most transactions, the formula is going to be the current period's dollar default amount divided by the original balance. To make it cumulative the formula should add the prior period's defaulted percentage. The complete formula in cell CP7 should look as follows:

   =N7/$L$7+CP6

   Copy and paste this formula over the range CP7:CP366. Also, be mindful that when using an SDA curve to generate defaults, those are calculated using the current balance. Check to make sure how triggers read in every case because they can be very customized. This trigger section of the Cash Flow sheet should look like Figure 7.2.



	W	X	Y	Z	AA	AB	AC
1							
2							
3							
4		Total Cash Available for Liabilities			Capture Trigger	Post Default Mo Trigger	
5							
6							
7		681,952			TRUE	FALSE	
8		698,488			TRUE	FALSE	
9		714,900			TRUE	TRUE	
10		731,181			TRUE	TRUE	

FIGURE 7.2 The triggers on the Cash Flow sheet should start taking form.

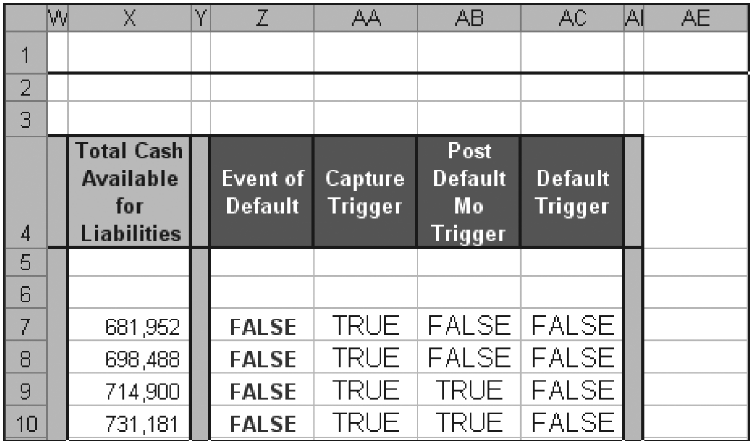
8. Still on the Cash Flow sheet go left to column AC. In cell AC4 enter the label **Default Trigger**. The formula will read very close to how the trigger is designed. When defaults exceed the amount indicated on the Inputs sheet then trip the trigger. The formula in cell AC7 should be:

   =IF(CP7>Trigger_Def,TRUE,FALSE)

   Copy and paste this formula over the range AC7:AC366.
9. The final trigger is very simple and requires no modification to the Inputs sheet. This trigger is a custom Event of Default trigger as determined by the model operator. Occasionally the need arises for a trigger to be assumed tripped at any given point within the deal for any given amount of time. Column Z on the Cash Flow sheet will be used for this.

   Label cell Z4 **Event of Default**. For now enter FALSE in cell Z7 and copy and paste this value over the range Z7:Z366. Make sure this is formatted as an input, since the model operator can change any period's value to assume a tripped trigger. At this point the Cash Flow sheet should look like Figure 7.3.
10. The final part is linking the Boolean values to the cash flow structure. Three of the triggers in Project Model Builder will be used to indicate a full rapid amortization state. This means that if any of those triggers are tripped, all cash is diverted immediately to senior principal. In such a case, the subordinated tranche will be cut off from receiving funds.

    Go to the Senior Principal Due column on the Cash Flow sheet (column AZ). Cell AZ7 needs to be modified to work differently when a trigger is tripped. This is going to require the use of an IF-OR combination. If the OR statement is unclear see the Toolbox section at the end of this chapter. Modify the formula in cell AZ7 as shown in bold—and make sure to enter the terminal close



	W	X	Y	Z	AA	AB	AC	AI	AE
1									
2									
3									
4		Total Cash Available for Liabilities		Event of Default	Capture Trigger	Post Default Mo Trigger	Default Trigger		
5									
6									
7		681,952		FALSE	TRUE	FALSE	FALSE		
8		698,488		FALSE	TRUE	FALSE	FALSE		
9		714,900		FALSE	TRUE	TRUE	FALSE		
10		731,181		FALSE	TRUE	TRUE	FALSE		

FIGURE 7.3 The completed trigger section of the Cash Flow sheet (with formatting).

parenthesis also shown in bold:

= **IF(OR(Z7, AB7,AC7),MIN(AX7,CB6),**IF(LiabPrinType1="Sequential", MIN((N7+Q7+R7),CB6),MIN((N7+Q7+R7)*LiabAdvRate1,CB6)))

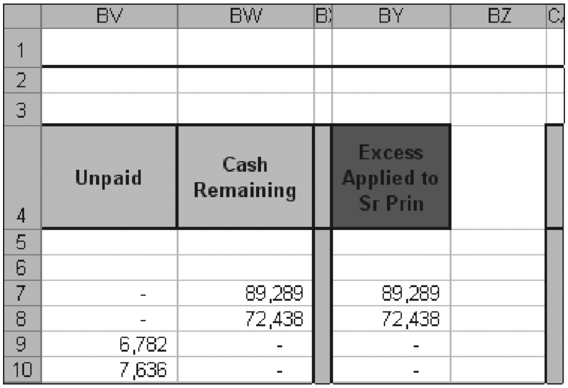
The modified part has been highlighted and includes an IF and OR statement that checks to see if any of the three triggers: Event of Default (column Z), Post Default Mo Trigger (column AB), or Default Trigger (column AC) have been tripped. If any triggers have tripped, the formula calculates the Senior Principal Due as whatever amount is available at that point in the waterfall. This essentially ends the flow of cash through the waterfall at this point until the Senior Principal is completely paid off.

11. The previous triggers are severe and prevent the subordinate tranche from receiving any funds. In certain cases, a trigger only accelerates the senior principal if cash remains at the end of the waterfall. The global trigger will be this type of acceleration trigger in Project Model Builder.

To make an acceleration trigger, an additional column needs to be set up at the end of the waterfall. Go to cell BY4 on the Cash Flow sheet and enter the label **Excess Applied to Sr Prin**. Cell BY7 needs a formula that returns the cash remaining if the trigger is tripped. However, the amount needs to be constrained by the balance of the senior debt. Enter the following formula in cell BY7:

=IF(AA7,MIN(BW7,CB6−BA7),0)

This formula checks to see if the global trigger has been tripped and populates any cash remaining at the end of the waterfall. It is constrained by a MIN



	BV	BW	B	BY	BZ	C
1						
2						
3						
4	Unpaid	Cash Remaining		Excess Applied to Sr Prin		
5						
6						
7	-	89,289		89,289		
8	-	72,438		72,438		
9	6,782	-		-		
10	7,636	-		-		

FIGURE 7.4 Excess amounts that are used for principal acceleration are calculated at the end of the waterfall.

function that takes the lesser of the amount remaining and the current balance less principal paid earlier in the waterfall. Copy this formula over the range BY7:BY366. This section should look like Figure 7.4.

12. The final step is to apply the excess that was just calculated to the senior principal. Modify the formula in cell CD7 so the senior debt balance is also reduced by amounts in BY:

=BA7+BY7

If there is any excess applied to the senior principal it will reduce the balance accordingly. Copy and paste this formula over the range CD7:CD366.

SWAPS

Swaps are confusing to many people because they involve conceptual flip-flopping. Whole books are dedicated to describing what swaps are and how they work. The goal of this section to give a very brief introduction to swaps and then move on to how a basic swap can be modeled in structured transactions.

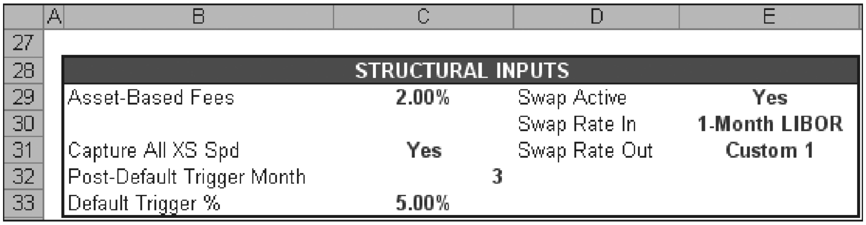
A *swap* is a financial instrument that hedges risk by swapping parties' exposure. For structured transactions an interest rate swap is the most commonly used swap. In a basic structured transaction a bank might have funded a transaction on a floating rate basis, but has structured the transaction with fixed rate assets. If the floating rate on the liabilities were to exceed the weighted average fixed rate of the assets, then the bank could take a loss.

Instead of taking such risk the bank enters into a fixed-for-floating interest rate swap. In such a case, the transaction will pay a fixed rate amount to a swap provider, while the swap provider will pay a floating rate amount to the transaction. All of the amounts are calculated off of a notional amortization schedule, which is a base case amortization of the certificates involved in the swap.

Project Model Builder incorporates a simple fixed-for-floating interest rate swap. It should be understood that many complex features to a swap are not included in this example, such as swap dealer fees, swap termination fees, and other granularities. The purpose is to understand how a swap uses interest rates to affect cash flow in and out of a transaction.

MODEL BUILDER 7.2: INCORPORATING A BASIC INTEREST RATE SWAP

1. There are only three assumptions that need to be manipulated on the Inputs sheet: (1) whether there is a swap in a transaction, (2) the basis for the swap money coming in, and (3) the basis for the swap money going out.
   On the Inputs sheet enter the following labels:
   D29: **Swap Active**
   D30: **Swap Rate In**
   D31: **Swap Rate Out**
   Cell E29 should be a data validation list with lstYesNo and should be named **Swap_ Active**. Cells E30 and E31 should also be data validation lists, but they should use lstInterestRates as the range. Name these cells **Swap_ In** and **Swap_ Out** respectively. Also select "1-Month LIBOR" for the Swap Rate In and "Custom 1" for the Swap Rate Out. The Inputs sheet should look like Figure 7.5.
2. Next go to the Cash Flow sheet, where columns AE:AK are used for the swap calculations. Enter the following labels:
   AE4: **Notional Swap Schedule**
   AF4: **Swap Rate In**
   AG4: **Swap Flow In**
   AH4: **Swap Rate Out**
   AI4 **Swap Flow Out**
   AJ4: **Swap Earn/Pay**
   AK4: **Cash Available**



	A	B	C	D	E
27					
28		STRUCTURAL INPUTS			
29		Asset-Based Fees	2.00%	Swap Active	Yes
30				Swap Rate In	1-Month LIBOR
31		Capture All XS Spd	Yes	Swap Rate Out	Custom 1
32		Post-Default Trigger Month	3		
33		Default Trigger %	5.00%		

FIGURE 7.5 The swap inputs are included within the Structural Inputs section.

3. Column AE is where the Notional Swap Schedule is stored. This is a base case amortization of the senior certificates. For purposes of the example model, use the Notional Swap Schedule provided in Excel file *MB7-2.xls* in the Ch07 folder on the CD-ROM. Copy and paste the schedule from the CD-ROM section to the range AE7:AE366 in the model under construction. This is the assumed amortization that the swap will base cash flow on.
4. Column AF is the rate that the swap counterparty pays the transaction. In this case, the transaction needs floating rate payments so it will be a floating rate as designated on the Inputs sheet. Earlier 1-Month LIBOR was designated as the rate. Similar to the other rate formulas on the Cash Flow sheet, enter the following formula in cell AF7:

   ```
   = IF(Swap_Active="No",0,OFFSET(Vectors!$D$6,Vectors!A7,
     MATCH(Swap_In,lstInterestRates,0)))
   ```

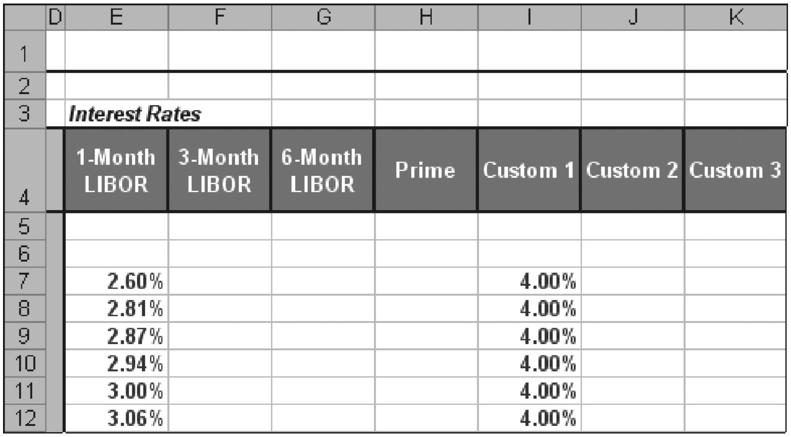
   While most of this formula is an OFFSET-MATCH combination that has been seen before, the beginning is an IF statement that checks to see if there is a swap in the deal or not. If not the rate will be zero, causing all calculations to be zero. Copy and paste this formula over the range AF7:AF366.
5. The dollar amount of swap flow in can be calculated with the swap rate in known. Enter the following formula in cell AG7:

   ```
   =AE7*AF7*C7
   ```

   This multiplies the swap rate in, by the notional schedule, and also by the day factor. Note that swaps sometimes use different day-count systems then transactions. In this example the day-count system was assumed to be the same as the transaction. Copy and paste this formula over the range AG7:AG366.
6. Prior to going to column AH, go to the Vectors sheet. Enter **4.00%** for cell I7. Copy and paste this value over the range I7:I366 so that every periods' value is 4.00 percent. This should look like Figure 7.6.
7. Go back to the Cash Flow sheet and to cell AH7. The formula here needs to return the swap rate that the transaction is paying to the swap counterparty. In this case it is a fixed rate because, on the Inputs sheet, the vector assumed is Custom 1, which in the previous step was assumed to be 4.00 percent for every period. Enter the following formula in cell AH7:

   ```
   = IF(Swap_Active="No",0,OFFSET(Vectors!$D$6,Vectors!A7,
     MATCH(Swap_Out,lstInterestRates,0)))
   ```

   Copy and paste this formula over the range AH7:AH366.



	D	E	F	G	H	I	J	K
1								
2								
3		*Interest Rates*						
4		1-Month LIBOR	3-Month LIBOR	6-Month LIBOR	Prime	Custom 1	Custom 2	Custom 3
5								
6								
7		2.60%				4.00%		
8		2.81%				4.00%		
9		2.87%				4.00%		
10		2.94%				4.00%		
11		3.00%				4.00%		
12		3.06%				4.00%		

FIGURE 7.6 Make sure the Vector's sheet is updated so Cash Flow sheet calculations work.

8. The calculation for the swap flow out is identical to the swap flow in, with the exception of the referenced rate. Enter the following formula into cell AI7:

   =AE7*AH7*C7

   Copy and paste this formula over the range AI7:AI366.
9. To determine the net amount paid or earned from the swap, subtract the swap flow out from the swap flow in. This is done with the following formula in cell AJ7:

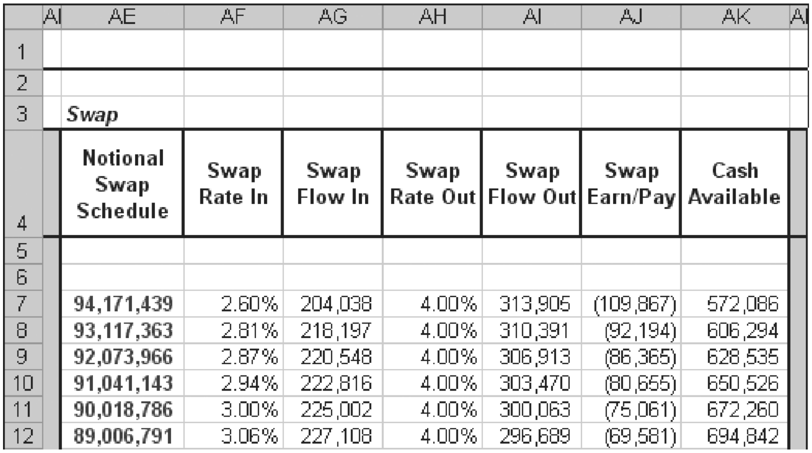
   =AG7−AI7

   Notice there is no MIN here because the value can be negative depending on the interest rate assumptions. Copy and paste this formula over the range AJ7:AJ366.
10. The swap section is completed by tracking the cash available after giving affect to swap payments. Enter the following formula in cell AK7:

    =X7+AJ7

    Copy and paste this formula over the range AK7:AK366. By now the swap section should look like Figure 7.7.
11. With the introduction of this advanced structure, a minor modification needs to be made to an existing formula so cash continues to flow through the waterfall. Change the formula in cell AP7 to:

    =AK7−AN7



	AE	AF	AG	AH	AI	AJ	AK
1							
2							
3	*Swap*						
4	**Notional Swap Schedule**	**Swap Rate In**	**Swap Flow In**	**Swap Rate Out**	**Swap Flow Out**	**Swap Earn/Pay**	**Cash Available**
5							
6							
7	**94,171,439**	2.60%	204,038	4.00%	313,905	(109,867)	572,086
8	**93,117,363**	2.81%	218,197	4.00%	310,391	(92,194)	606,294
9	**92,073,966**	2.87%	220,548	4.00%	306,913	(86,365)	628,535
10	**91,041,143**	2.94%	222,816	4.00%	303,470	(80,655)	650,526
11	**90,018,786**	3.00%	225,002	4.00%	300,063	(75,061)	672,260
12	**89,006,791**	3.06%	227,108	4.00%	296,689	(69,581)	694,842

FIGURE 7.7 The Swap section on the Cash Flow sheet is complete.

Without this change the swap calculation will have no effect on the rest of the waterfall. Make sure to copy this change down to cell AP366.

FINAL NOTES ON SWAPS

A swap can introduce complex changes to the cash flow depending on the interest rates assumed. If the rates are assumed to be extremely volatile then the swap earn/pay amounts can be very large. Keep in mind that this section has not assumed any cost for the swap. The more beneficial a swap is to a deal, the more expensive it will probably be. Swap expenses should be quoted from a swap provider who can provide up-to-the-minute market prices.

RESERVE ACCOUNTS

Reserve accounts are the most tangible and easiest form of credit enhancement to understand. They are accounts set aside exclusively for a transaction in case there are problems making payments to certain liabilities. If the liability cannot be met through normal cash flow and the deal documentation allows, a reserve account can be used to make up payment shortfall. Reserve accounts are either cash funded from the start of the transaction or they can be designed to grow by trapping excess cash in a transaction. Conversely, as deals amortize the reserve account can also amortize or stay at a fixed amount.

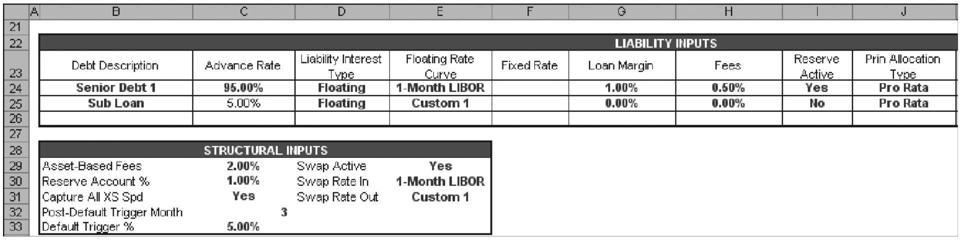
Issuers tend not to want cash-funded reserve accounts because the money being reserved is untouchable and not earning a high return. It is important that the money can only be accessed for certain obligations; otherwise there is little value in assuming the reserve amount.

Another important feature of reserve accounts is that they are typically reimbursed if there is enough cash in the transaction. A minimum reserve amount is often required and when the reserve balance goes below the minimum, reimbursements are necessary. This is important for the methodology being implemented in Project Model Builder because the placement of the reserve account calculations depends on where reimbursements are written into the priority of payments.

MODEL BUILDER 7.3: INCORPORATING A CASH-FUNDED RESERVE ACCOUNT

1. A cash-funded reserve account is assumed in Project Model Builder. This type of reserve account has cash funded by the asset issuer, which is typically a percent of the assets. In the deal documentation there will be language that designates what liabilities the reserve account covers. Project Model Builder assumes that only the senior liabilities have access to the reserve account. Typically fees and top-level items on the waterfall have access to the reserve account, but modeling this is not necessary because very few entities would do a deal that is risky enough where the top of the waterfall has the possibility of drawing from a reserve account.

   To start modeling the reserve account go to the Inputs sheet and enter the following label in cell I23, **Reserve Active**. Make cells I24 and I25 data validation lists using lstYesNo as the range. Name cell I24 **LiabReserveOnOff1** and I25 **LiabReserveOnOff2**. Enter the label **Reserve Account %** in B30 and the value **1.00%** in C30. Name cell C30 **RsrvPercent**. The Inputs sheet should look like Figure 7.8.
2. Go to the Cash Flow sheet and enter the following labels:
   BG4: **Reserve Account Minimum**
   BH4: **Reserve Account Beginning Balance**
   BI4: **Withdrawals**



LIABILITY INPUTS

Debt Description	Advance Rate	Liability Interest Type	Floating Rate Curve	Fixed Rate	Loan Margin	Fees	Reserve Active	Prin Allocation Type
Senior Debt 1	**95.00%**	**Floating**	**1-Month LIBOR**		**1.00%**	**0.50%**	**Yes**	**Pro Rata**
Sub Loan	5.00%	**Floating**	**Custom 1**		**0.00%**	**0.00%**	**No**	**Pro Rata**

STRUCTURAL INPUTS

Asset-Based Fees	**2.00%**	Swap Active	**Yes**
Reserve Account %	**1.00%**	Swap Rate In	**1-Month LIBOR**
Capture All XS Spd	**Yes**	Swap Rate Out	**Custom 1**
Post-Default Trigger Month	**3**		
Default Trigger %	**5.00%**		

FIGURE 7.8 The Reserve Account additions are in multiple sections of the Inputs sheet.

BJ4: **Reimbursements**
BK4: **Reserve Account Ending Balance**
BL4: **Cash Remaining**

3. The reserve account minimum is the first concept in this section. It is the amount that should be maintained in the reserve account each period. If the deal has an amortizing reserve, this amount will decrease as the asset pool balance decreases. However, in Project Model Builder the reserve is a fixed amount. Enter the following formula in cell BG7:

   =AssetCurBal1*RsrvPercent

   This formula multiplies the reserve percent from the Inputs sheet by the asset pool's beginning balance. Copy and paste this formula over the range BG7:BG366.
4. To calculate a reserve section the beginning of period and end of period balance should be split into separate columns. In this example, the beginning of period reserve balance is always the end of period balance from the prior period. Enter the following formula in cell BH7:

   BK6

   Copy and paste the formula over the range BH7:BH366.
5. In order to populate values while working with the reserve account section, skip over to cell BK6. Enter the following formula:

   =IF(A6=0,AssetCurBal1*RsrvPercent,BH6−BI6+BJ6)

   This formula begins by checking to see if the current period is the start of the transaction (period 0). If it is then the reserve account is assumed to start with a funded balance of the reserve percent multiplied by the asset pool balance. Otherwise the ending balance will be the beginning balance minus withdrawals plus reimbursements.
6. The next focus of this section is withdrawals from the reserve account. Withdrawals should only be made if the deal documentation allows. In this model assume that only senior interest and principal are covered by the reserve account. Go to cell AV4 and enter the label **Unpaid Covered by Reserve**. Next go to cell AW4 and enter the label **Unpaid**. In cell AV7 enter:

   =IF(LiabReserveOnOff1="No",0,MIN(AU7,BH7))

   This formula first checks to see if the reserve is active. If it is not active then there is no coverage of unpaid amounts. If the reserve is active then the lesser of "What You Have and What You Need" is applied. The needed part is the unpaid interest that is calculated in cell AU7, while the "have" part is the beginning balance of the reserve account for the period in cell BH7. With this set up the amount covered by the reserve will never be more than the reserve account balance. Copy and paste this formula over the range AV7:AV366.

7. If there is not enough cash in the reserve account the unpaid amount needs to be carried over. Label cell AW4 **Unpaid** and enter the following formula in AW7:

   =AU7−AV7

   This subtracts the amount covered by the reserve from the unpaid amount. Copy and paste this formula over the range AW7:AW366.
8. A similar process now needs to take place for the senior principal. Label cell BC4 **Unpaid Covered By Reserve**. In BC7 enter the following formula:

   =IF(LiabReserveOnOff1="No",0,MIN(BB7,BH7−AV7))

   The major difference in this formula is that the amount used from the reserve (cell AV7) is subtracted from the reserve account balance (cell BH7). This is a logical assumption if one assumes that the cash flow waterfall has a time element from left to right. First, any unpaid interest would be paid from the reserve account and then unpaid principal would be covered only if there is money available left in the reserve account. Copy and paste this formula over the range BC7:BC366.
9. The formula for tracking the carried over unpaid amounts also needs to be created. Label cell BD4 **Unpaid** and in BD7 enter:

   =BB7−BC7

   Copy and paste this formula over the range BD7:BD366.
10. Go back over to the reserve account section to cell BI7. The amount withdrawn is the sum of the fields that calculate amounts covered by the reserve account. Enter the following formula in cell BI7:

    =AV7+BC7

    Copy and paste this formula over the range BI7:BI366.
11. Calculating reimbursements is the next and perhaps the most complicated part of modeling reserve accounts. The amount to be reimbursed should always be the reserve minimum less the reserve account beginning of period balance. Reimbursements are not necessary when the debt that is being covered by the reserve is paid off and the reserve account is liquidated, or when the reserve balance is at or above the reserve minimum. Enter the following formula in cell BJ7 to accomplish all of this:

    =IF(CB6<1,0,MAX(MIN(BG7−BH7,BE7),0))

This formula first checks the beginning balance of the debt being covered by the reserve (in this case the senior debt) and makes reimbursements zero if the debt is paid off. If the debt still exists then the formula performs a lesser of "What You Have and What You Need." In this calculation, the reserve account balance subtracted from the reserve account minimum is needed, while the cash remaining in cell BE7 is available.

However, occasionally the reserve account balance might be higher than the minimum. If that is the case then this formula will produce a negative result. The MAX formula insures that there are no negative reimbursements in such cases. Copy and paste the formula in BJ7 over the range BJ7:BJ366.

12. The reserve account section is completed by creating a cash remaining calculation by entering the following formula in cell BL7:

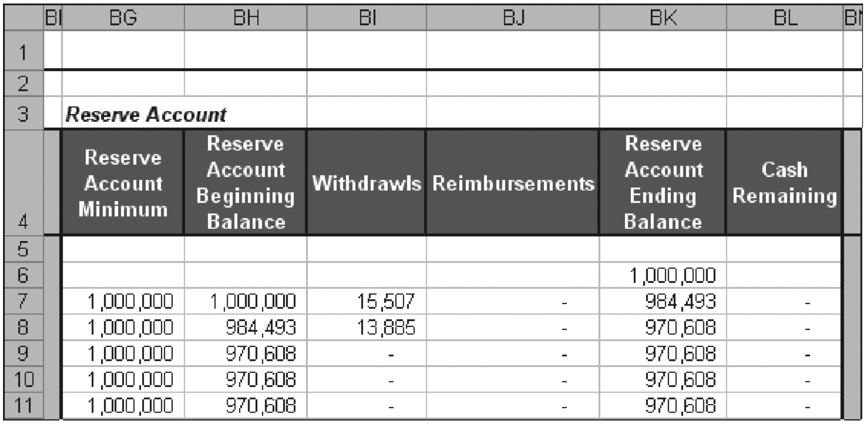
=BE7−BJ7

This formula is important because it shows that the reimbursements are removed from the cash available at this point in the cash flow waterfall. If reimbursements were anywhere else in the documentation, then this reference would have to be moved accordingly. Copy and paste this formula over the range BL7:BL366. The entire reserve account section should look like Figure 7.9.

13. A few modifications need to be made to existing formulas to make the reserve account work correctly. First, cell BR7 needs to be modified to:

=BL7−BP7

This will use the cash remaining from the reserve account section rather than skipping over all of the reserve account calculations. Make sure to copy the formula down to BR366.



	BI	BG	BH	BI	BJ	BK	BL	BI
1								
2								
3		***Reserve Account***						
4		**Reserve Account Minimum**	**Reserve Account Beginning Balance**	**Withdrawls**	**Reimbursements**	**Reserve Account Ending Balance**	**Cash Remaining**	
5								
6						1,000,000		
7		1,000,000	1,000,000	15,507	-	984,493	-	
8		1,000,000	984,493	13,885	-	970,608	-	
9		1,000,000	970,608	-	-	970,608	-	
10		1,000,000	970,608	-	-	970,608	-	
11		1,000,000	970,608	-	-	970,608	-	

FIGURE 7.9 The Reserve Account section on the Cash Flow sheet.

14. Also, the senior interest and principal aggregations in the balance section need to include amounts covered by the reserve account. Change CC7 to:

    =AT7+AV7

    And CD7 to:

    =BA7+BC7+BY7

    Make sure to copy the changes down to row 366 for each column.
15. The final step to finish the operating part of the cash flow waterfall is tracking excess cash. In cell BZ4 enter the label **Excess Released**. In cell BZ7, enter the following formula:

    =BW7−BY7

    This formula subtracts any excess cash that was used to pay down principal from the cash remaining after sub loan principal is paid (essentially the end of the waterfall). The amounts in this column will be released from the transaction to whoever holds the rights to the excess. Copy and paste this formula over the range BZ7:BZ366.
16. One final modification needs to be made to the Principal Due calculation in column AZ. Modify AZ7 with the following shown in bold:

    = IF(OR(Z7, AB7,AC7),MIN(AX7,**AX7+BH7−AV7,**CB6),
    IF(LiabPrinType1="Sequential",MIN((N7+Q7+R7),CB6),
    MIN((N7+Q7+R7)*LiabAdvRate1,CB6)))

    What this slight change does is require the cash reserve to be used in the case of a trigger breach. Most transactions will use the cash reserve in such a manner, but each deal's documentation should be checked to see how the components operate. Copy and paste cell AZ over the range AZ7:AZ366.

CONCLUSION OF THE CASH FLOW WATERFALL

The cash flow waterfall is now completely operational. However, it should be checked and formatted. All cash should flow through from left to right and down. Make sure to check the model under construction with the completed model so that the calculations are the same. This can be done by taking the sum of many of the individual columns in row 5 and checking to see if the sums are the same as those in row 5 of the completed model.

Also notice that in row 3 of the completed model there are names for the different sections. While these have no calculation value, they are helpful for jumping between sections in the waterfall by using CTRL + arrow keyboard commands.

Finally, as mentioned earlier, the color system used is useful for grouping similar concepts. Using the same color scheme for concepts in every model built allows a model operator to quickly identify sections of the model. Also, the grey separation lines are useful to break up concepts and create a “modular” type system. The breaks are particularly useful when sections of the waterfall need to be added or removed.

The remainder of the book focuses on additions to the model that ensure it is operating properly, efficient methods and tools for extracting and manipulating data, analysis of the data produced by the model, and in general understanding the model that has been created.

TOOLBOX

AND and OR

AND and OR are two important functions used in this chapter. Understanding the difference between them is simple but extremely important. An AND function returns a TRUE value if all the conditions in the formula are true. If just one of the conditions set up in the AND formula are false, then the return value is false. An OR function returns a TRUE value if any of the conditions in the formula are true. It does not matter how many conditions are false.

These two functions are very effective at translating written tests into a computer model. They are perfect for sections in a term sheet, where certain conditions must be met for an event to take place.

CHAPTER 8

Analytics and Output Reporting

Sophisticated financial models are virtually useless if the results are inaccurate or difficult to understand. The best financial models have internal tests that track computational and conceptual accuracy. While the purpose of these tests is to make sure that the model calculation is valid, the interpretation of the calculations needs to be created and presented in a comprehensive and simple-to-understand format. The goal is to have a single output sheet that summarizes the model calculations and provides metrics for financial performance.

INTERNAL TESTING

Internal tests are the first step in reporting results. They make sure that as a model is modified and changed, the core logic still functions as originally intended. The tests also allow a user to troubleshoot problems faster if the need arises. Tests mainly focus on cash flow calculations, but allow a user to interpret conceptual validity based on the test results. This will become clearer as internal tests are implemented in Project Model Builder.

A number of tests will be set up in Project Model Builder to make sure that the model is functioning correctly. Each one is discussed as it is implemented.

Cash In versus Cash Out

One of the most fundamental tests is to make sure that whatever cash has gone into the deal has come out. This can be reworded in a more important manner: to make sure that whatever cash is used in the deal was funded by cash coming into the transaction. Essentially, a cash flow model has a finite amount of money from the assets. The liabilities can only draw from this finite amount.

A critical error that sometimes occurs is when a model is modified and cells are not linked correctly cash is created or used from a nonexistent source. To make sure this does not occur the cash that comes into a deal should be tracked and compared against the cash that flows out of the deal. Except for extraordinarily rare circumstances there should never be a difference.

MODEL BUILDER 8.1: CASH IN VERSUS CASH OUT TEST

1. Most tests will be tracked on the Cash Flow sheet, but reported on the Inputs sheet and eventually the Output sheet when it is created. For cash in versus cash out, columns will need to be created on the Cash Flow sheet.
   Go to column CJ of the Cash Flow sheet and enter these labels in the following cells:
   CJ4: **Cash In**
   CK4: **Cash Out**
   CL4: **Difference**
2. The Cash In is all of the cash that is available to pay liabilities. At first this is all of the cash that the assets generate. While that is a large part of the Cash In each period, two of the advanced features of the structure provide cash: the swap and the reserve account. In cell CJ7 enter:

   = Q7 + R7 + T7 + U7 + AG7 + BI7

   Tracing each one of these back, the cash flow coming into the transaction consists of voluntary prepayments, scheduled asset amortization, asset yield, recoveries from defaulted assets, swap flows in, and any amount coming in from the reserve. Copy this formula and paste it over the range CJ7:CJ366.
3. The Cash Out has more references since there many points that cash comes out of the transaction. Enter the following formula in cell CK7:

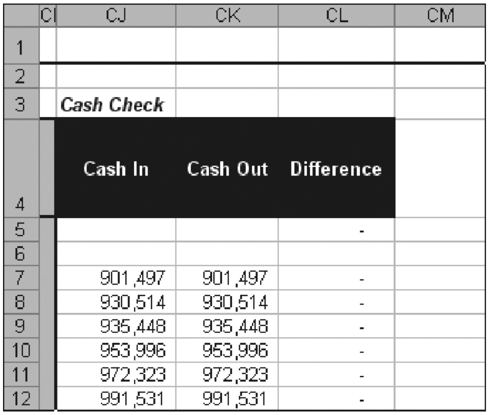
   = AI7+AN7+AT7+AV7+BA7+BC7+BJ7+BP7+BU7+BY7+BZ7

   The formula is self-explanatory for simple liabilities such as fees paid (cell AN7), but notice some of the less obvious references such as cells AV7, BC7, BJ7, BY7, and BZ7. Swap payments sometimes go out so these must be deducted. Remember, too, that reserve account withdrawals were considered to be Cash In, so the actual use of that cash to cover liabilities is Cash Out. Also, if the reserve is reimbursed cash leaves the transaction. Finally, if there is excess cash at the end of the waterfall it is used by either applying the cash to senior principal or releasing it. Copy and paste cell CK7 over the range CK7:CK366.
4. The real test now is to see if the Cash In minus the Cash Out is equal to zero. Enter the following formula in cell CL7:

   = CJ7−CK7

   Copy and paste this formula over the range CL7:CL366. See Figure 8.1 for the new section on the Cash Flow sheet.
5. Checking the entire "Difference" column each run would be tedious, so some quick links need to be set up. First sum up range CL7:CL366 in cell CL5 by using:

   = SUM(CL7:CL366)



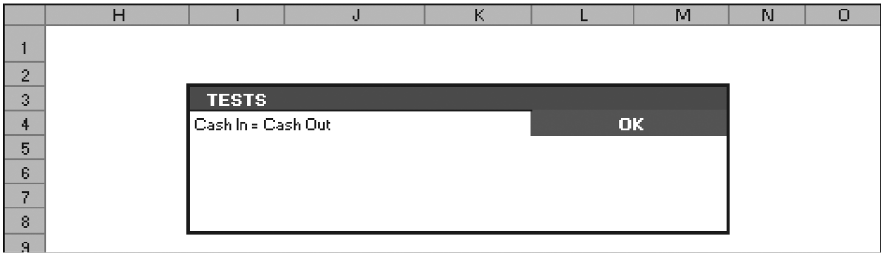
	CI	CJ	CK	CL	CM
1					
2					
3		*Cash Check*			
4		Cash In	Cash Out	Difference	
5				-	
6					
7		901,497	901,497	-	
8		930,514	930,514	-	
9		935,448	935,448	-	
10		953,996	953,996	-	
11		972,323	972,323	-	
12		991,531	991,531	-	

FIGURE 8.1 Cash In versus Cash Out helps prevent funds from being "created" in the model.

6. Next go to the Inputs sheet. Since most of the model is controlled from this sheet it is useful to make sure the model is running correctly as assumptions are changed. Label cell I3 **TESTS** and I4 **Cash In = Cash Out**. Go to cell L4 and enter the following formula:

   = IF('Cash Flow'!CL5=0,"OK","ERROR")

   This formula checks the sum of the cash in versus cash out differences to see if it is zero. If it is then the model is working correctly, otherwise there is an error. Many of the tests will be set up using this IF statement format with either an OK or ERROR return. Conditional formatting is particularly useful here so that it is very obvious when there is an error. If unfamiliar with conditional formatting, see the ToolBox section of this chapter. Otherwise make the cell shading green and font bold white for when the cell is OK and red shaded with bold white font when there is an ERROR.



	H	I	J	K	L	M	N	O
1								
2								
3		TESTS						
4		Cash In = Cash Out			OK			
5								
6								
7								
8								
9								

FIGURE 8.2 The TESTS section is on the Inputs sheet for quick reference.

Balances at Maturity

Two very important tests check the asset and liability balance at maturity. The more important of the two is the debt balance at maturity. If a principal amount remains unpaid at maturity, most likely the debt balance will incur a loss. These tests are very quick to implement.

MODEL BUILDER 8.2: BALANCES AT MATURITY TESTS

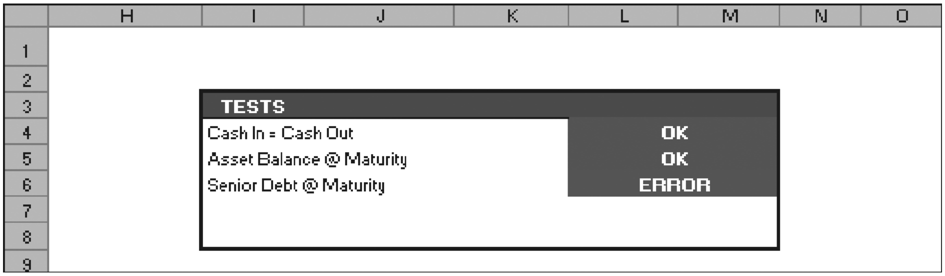
1. Go to the Inputs sheet and enter the following labels:
   I5: **Asset Balance @ Maturity**
   I6: **Senior Debt @ Maturity**
2. In cell L5 enter the following formula:

   = IF('Cash Flow'!V366<1,"OK","ERROR")

   This checks the final possible period for the assets. If there is a balance greater than $1, than there is a problem. Use the same conditional formatting on cell L5 as seen in the first test for this chapter.
3. The senior debt balance will be looked at with more scrutiny since it is the focus of further analysis. To make automating the model easier in Chapter 10, go to the Cash Flow sheet and name cell CB366: **FinalLoanBal**.
4. In cell L6 enter the following formula:

   = IF(FinalLoanBal<1,"OK","ERROR")

   If the senior debt is not paid off by the final period, then an error will show. Use the same conditional formatting as seen in the first test for this chapter. Note that this test can be done for the subordinate debt if needed, but is only done for the senior debt in Project Model Builder. See Figure 8.3 for detail.



TESTS	
Cash In = Cash Out	OK
Asset Balance @ Maturity	OK
Senior Debt @ Maturity	ERROR

FIGURE 8.3 The Balance tests are integral to the model and are readily seen on the Inputs sheet.

Asset Principal Check

In a system where there is a finite amount of cash coming from the assets, it is imperative that the cash inflow is correct. As asset amortization goes beyond simple amortization and exotic products with unusual loss curves or payment methods are incorporated, it is important to make sure the individual components of asset amortization equal the original amount. In other words, does the sum of all the prepayments, scheduled amortization, and defaulted amounts equal the beginning balance of the assets?

Remember that prepayments, scheduled amortization, and defaults reduce the asset balance each month. If this is the case, then summing up all of those reductions should equal the beginning balance. An error often occurs here when there are data problems with the representative lines or in a loan level model, with the individual loans. Also a loss or prepay curve could be off or a custom payment might be calculated incorrectly.

MODEL BUILDER 8.3: ASSET PRINCIPAL CHECK TEST

1. Go to the Cash Flow sheet and make sure that there are sums of the voluntary prepay, actual amort, and new default columns. This is done by the following formulas:

   N5: = SUM(N7:N366)

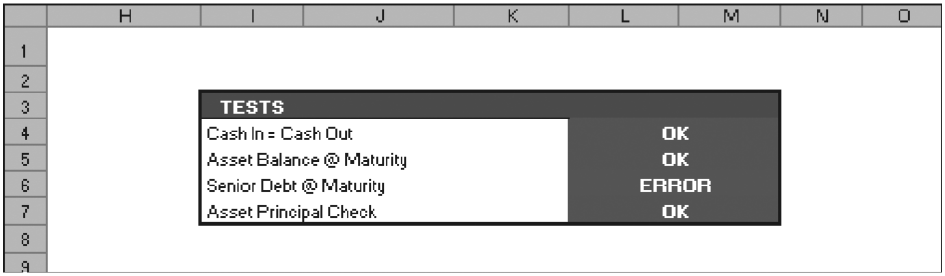
   Q5: = SUM(Q7:Q366)

   R5: = SUM(R7:R366)

2. Go back to the Inputs sheet and enter the following label in cell I7: **Asset Principal Check.** In cell L7 enter the following formula:

   = IF(ROUND(('Cash Flow'!N5+'Cash Flow'!Q5+'Cash Flow'!R5),0)
   = ROUND('Cash Flow'!V6,0),"OK","ERROR")

   This formula adds up all the sums of the asset reduction and compares that sum to the original asset balance. The ROUND function is used because as assets are calculated there can be very minor differences in the decimal positions that will falsely trip this test. Also use the same conditional formatting on cell L7 as seen in the first test of this chapter.



TESTS	
Cash In = Cash Out	OK
Asset Balance @ Maturity	OK
Senior Debt @ Maturity	ERROR
Asset Principal Check	OK

FIGURE 8.4 The Tests section of the Inputs sheet is complete.

PERFORMANCE ANALYTICS

The Cash Flow sheet is impractical to quickly garner information from unless additional calculations are performed. The calculations should explain relevant characteristics for financial analysis. All parties to a transaction are primarily concerned with yield, loss, and the timing of cash flows. These concepts are captured by monthly yield, bond equivalent yield, duration, and weighted average life calculations.

Monthly Yield

Monthly yield is more importantly used to calculate bond equivalent yield, rather than as a metric on its own. The monthly yield of an asset or debt is the discount rate that makes the present value of all of the cash flow from the asset or to the debt equivalent to the initial principal balance. For assets, the cash flow that is counted is the yield, scheduled amortization, voluntary prepayments, and the recovered principal. For debt, the cash flow that is counted is the interest and principal.

MODEL BUILDER 8.4: CALCULATING MONTHLY YIELD

1. Many of the performance analytics require a stream of cash flows for the assets and debt that is discounted. Instead of doing this on the cash flow sheet, create a new sheet named **Analytics**.
2. Since most of the model is complete, there are many sections that can be referenced instead of recreating formulas. In cell A13 enter:

   = 'Cash Flow'!A4

   Drag cell A13 over the range A13:C375. This will reference the dates and timing section from the Cash Flow sheet. The range A14:C15 on the Analytics sheet will have zero value references, so clear the contents of these cells.

3. To the right of the dates and timing will be the discounted cash flows for the assets and the debt. Label the following cells by entering these cell references:
   E13: **= AssetDes1**
   F13: **= LiabDes1**
   G13: **= LiabDes2**
   Column D will be left blank for a space between the dates and timing, and the discounted cash flows.
4. Before calculating the discounted cash flows, a few more items need to be set up. Enter the following references to populate labels:
   E3: **= AssetDes1**
   F3: **= LiabDes1**
   G3: **= LiabDes2**
5. In cells B4 and B5, enter the following labels respectively: **Initial Principal** and **Monthly Yield**.
6. The initial principal amounts are needed for the calculations and should be referenced as follows:
   E4: = **AssetCurBal1**
   F4: **= 'Cash Flow'!CB6**
   G4: **= 'Cash Flow'!CF6**
7. For now enter a starting monthly yield of **1.0%** in cells E5, F5, and G5. Also, for automation purposes later name the range E5:G5, **rngYieldChange**.
8. Next the discounted cash flows need to be calculated. For the assets enter the following formula in cell E16:

   = ('Cash Flow'!Q7+'Cash Flow'!R7+'Cash Flow'!T7
   +'Cash Flow'!U7)/(1 + $E$5)^$A16

   This formula adds the voluntary prepayments, amortization, interest, and recoveries as cash flow from the assets. That sum is then discounted by the assumed monthly yield. Copy this formula and paste it over the range E16:E375.
9. The discounted cash flow for the debt is similar, but the cash flows associated with them are interest and principal. Enter the following formulas in cells F16 and G16:

   F16: = ('Cash Flow'!CC7 + 'Cash Flow'!CD7)/(1 + $F$5)^$A16
   G16: = ('Cash Flow'!CG7 + 'Cash Flow'!CH7)/(1 + $G$5)^$A16

   Copy and paste these formulas over their respective ranges (F16:F375 and G16:G375).
10. Go back up to cell B6 and enter the label **PV Difference**. The present value difference is the result of subtracting the sum of the present valued cash flow

from the initial balance. For the assets and debt enter the following formulas:

E6: = SUM(E16:E375)-E4

F6: = SUM(F16:F375)-F4

G6: = SUM(G16:G375)-G4

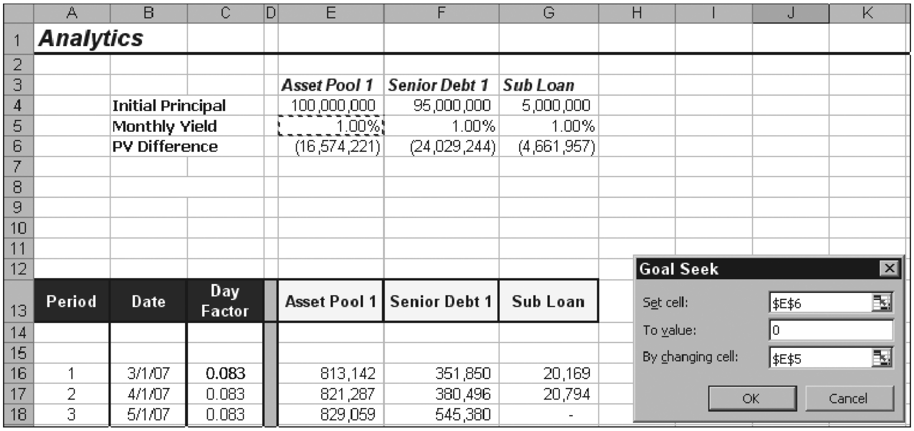
Name the range E6:G6, **rngYieldTarget**.

Calculating the Monthly Yield

The yield calculations are in place, but to really understand what is happening look at the PV difference. For the assets and debt, there should be a negative PV difference. This means that the discount rate, which is the assumed monthly yield, is too high. Try reducing the assumed monthly yield to .50 percent for each. Now there is a positive PV difference, which means that the assumed monthly yield is too low.

Trying higher and lower values is an inefficient means of what is technically known as a *divide-and-conquer algorithm*. Excel has a built-in tool called *Goal Seek* that can perform the operation to find the exact monthly yield necessary. If Goal Seek is unfamiliar, see the Toolbox section of this chapter. Otherwise, try goal seeking the asset monthly yield as an example.

Open the Goal Seek tool and make the "Set Cell" referenced to the PV difference in cell E6. The "To value" should be set to zero because the correct monthly yield will make the present value of the cash flows equal to the initial balance. Subtracting the same amount from one another should produce zero. Finally, the "By changing cell" is the monthly yield because that is the value that needs to iterate until the PV difference is zero. Once these parameters are entered as seen in Figure 8.5, run Goal



	A	B	C	D	E	F	G
1	*Analytics*						
2							
3					*Asset Pool 1*	*Senior Debt 1*	*Sub Loan*
4		Initial Principal			100,000,000	95,000,000	5,000,000
5		Monthly Yield			1.00%	1.00%	1.00%
6		PV Difference			(16,574,221)	(24,029,244)	(4,661,957)
7							
8							
9							
10							
11							
12							
13	Period	Date	Day Factor		Asset Pool 1	Senior Debt 1	Sub Loan
14							
15							
16	1	3/1/07	0.083		813,142	351,850	20,169
17	2	4/1/07	0.083		821,287	380,496	20,794
18	3	5/1/07	0.083		829,059	545,380	-

Goal Seek

Set cell: $E$6

To value: 0

By changing cell: $E$5

FIGURE 8.5 Excel's Goal Seek tool is used to find the monthly yield.

Seek. The monthly yield should change until the PV difference is zero. Note that there could be errors in trying to calculate the yield of an asset or liability that is taking a complete loss and that the yield should just be assumed to be zero.

While this goal seek process should be repeated for both debt tranches, it is very tedious and has to be redone every time a change to the cash flow is made. Later in the text, VBA automation is used to control this process and allows a user to click a single button to calculate the yield for the assets and all debt tranches.

Bond-Equivalent Yield

A standard convention in the capital markets is to report yield in terms of the *bond-equivalent yield* (BEY), which can be thought of as an annual yield. This is achieved by multiplying the semiannual yield by two. Debate exists about the best annual yield measure, but since convention uses BEY, it should be incorporated into the model.

MODEL BUILDER 8.5: CALCULATING BOND-EQUIVALENT YIELD

1. On the Analytics sheet, in cell B7 enter the label **BEY**. To calculate the BEY, the semiannual yield needs to be calculated from the monthly yield. Do this by entering the following formula in cell E7:

   =((1+E5)^6 – 1)

2. As stated earlier, the BEY is just the semiannual yield doubled. Modify the formula so it reads:

   =2*((1+E5)^6 – 1)

3. Copy and paste this formula over the range E7:G7. The asset and debt BEYs should be very similar to the average annual interest rates of the assets and debt tranches.

MODIFIED DURATION

Duration measures a bond value's sensitivity to rate changes. Fabozzi officially defines it as "the approximate percentage change in value for a 100 basis point change in rates."[1] The formula for modified duration is:

$$\frac{1}{(1+yield/k)}\left[\frac{1^{*}PVCF_{1}+2^{*}PVCF_{2}+\cdots+n^{*}PCVCF_{n}}{k^{*}\mathrm{Pr}ice}\right]$$

[1] Frank J. Fabozzi, *Fixed Income Analysis*. (New Hope, PA: Frank J. Fabozzi Associates, 2000), 255.

where

k = periodicity of the payments

n = total periods

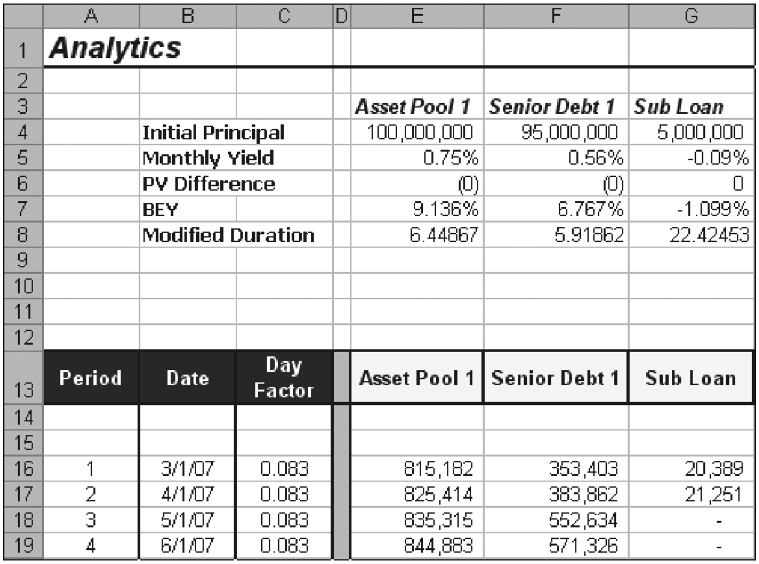
$PVCF_t$ = present value of the cash flow in period t discounted by the yield

MODEL BUILDER 8.6: CALCULATING MODIFIED DURATION

1. On the Analytics Sheet, label cell B8: **Modified Duration**. The formula for modified duration translates easily into Excel. In cell E8 enter:

   = 1/((1+E5/12))*(SUMPRODUCT(E16:E375,$A$16:$A$375)/(12*E4))

2. Most of the formula is self-explanatory with the written equation presented earlier. The one item to notice is that the SUMPRODUCT function is used on the present values of the cash flows for more efficient calculation. Copy and paste this formula over the range E7:G7. The final Analytics section should look like Figure 8.6.



	A	B	C	D	E	F	G
1	*Analytics*						
2							
3					*Asset Pool 1*	*Senior Debt 1*	*Sub Loan*
4		Initial Principal			100,000,000	95,000,000	5,000,000
5		Monthly Yield			0.75%	0.56%	-0.09%
6		PV Difference			(0)	(0)	0
7		BEY			9.136%	6.767%	-1.099%
8		Modified Duration			6.44867	5.91862	22.42453
9							
10							
11							
12							
13	Period	Date	Day Factor		Asset Pool 1	Senior Debt 1	Sub Loan
14							
15							
16	1	3/1/07	0.083		815,182	353,403	20,389
17	2	4/1/07	0.083		825,414	383,862	21,251
18	3	5/1/07	0.083		835,315	552,634	-
19	4	6/1/07	0.083		844,883	571,326	-

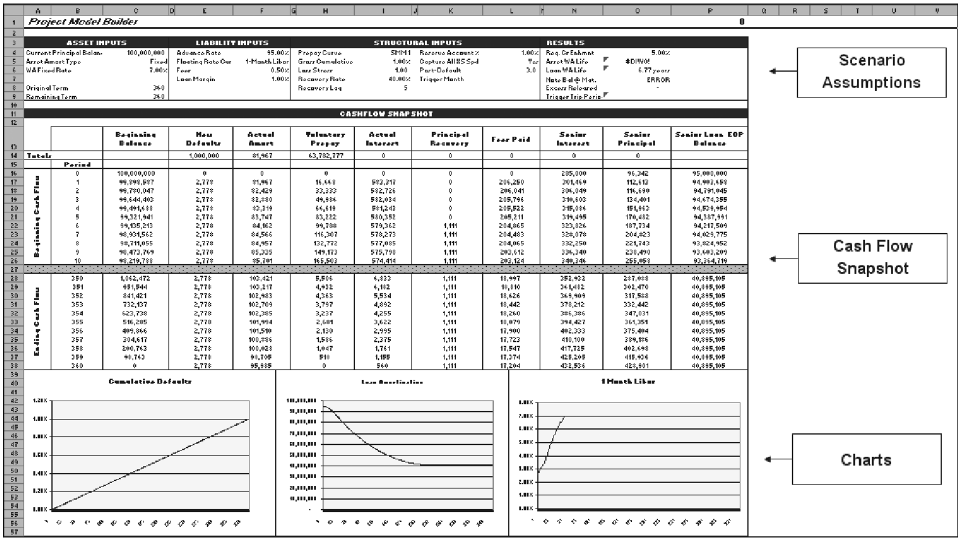
FIGURE 8.6 The analytics sheet is now complete.

OUTPUT REPORTING

Most people do not have the time to sift through the minutiae of a complex financial model. Users of financial models generally want an understanding of results pertinent to their decision-making process. A majority of these results can be summarized in one page.

This single page, known as an *output report*, most efficiently conveys a cash flow model when it is split into three separate sections: scenario assumptions, cash flow snapshot, and charts. Figure 8.7 provides a general overview of an output sheet. These three sections make it easy for a person to quickly see what assumptions were put into the model, what the results are for important periods in the cash flow, and how certain data intensive assumptions/results look graphically in charts.

An important part of the output report is that it will be created as its own worksheet in the model and should contain as few entered values as possible. This means that nearly all of the sheet should reference parts of the model. The reasoning behind this is that as assumptions and results change per scenario, the output report should automatically update. Errors usually occur when a scenario is run, but the output sheet is not updated. By using as many references as possible there will be fewer possibilities for an error and an increase in model operation efficiency.



Project Model Builder 0

ASSET INPUTS		LIABILITY INPUTS		STRUCTURAL INPUTS				RESULTS	
Current Principal Balan	100,000,000	Advance Rate	95.00%	Prepay Curve	SMM1	Reserve Account %	1.00%	Req. Cr Enhmnt	5.00%
Asset Amort Type	Fixed	Floating Rate Cur	1-Month Libor	Gross Cumulative	1.00%	Capture All XS Spd	Yes	Asset WA Life	#DIV/0!
WA Fixed Rate	7.00%	Fees	0.50%	Loss Stress	1.00	Port-Default	3.0	Loan WA Life	6.77 years
		Loan Margin	1.00%	Recovery Rate	40.00%	Trigger Month		Note Bal @ Mat.	ERROR
Original Term	360			Recovery Lag	5			Excess Released	-
Remaining Term	360							Trigger Trip Perio	

CASHFLOW SNAPSHOT

	Period	Beginning Balance	New Defaults	Actual Amort	Voluntary Prepay	Actual Interest	Principal Recovery	Fees Paid	Senior Interest	Senior Principal	Senior Loan EOP Balance
Totals			1,000,000	81,967	63,782,777	0	0	0	0	0	
Beginning Cash Flow	0	100,000,000	0	0	0	0	0	0	285,000	96,342	95,000,000
	1	99,898,587	2,778	81,967	16,668	583,317	0	206,250	301,469	112,613	94,903,658
	2	99,780,047	2,778	82,429	33,333	582,726	0	206,041	306,049	116,690	94,791,045
	3	99,644,403	2,778	82,880	49,986	582,034	0	205,796	310,603	134,401	94,674,355
	4	99,491,688	2,778	83,319	66,619	581,243	0	205,522	315,086	151,963	94,539,454
	5	99,321,941	2,778	83,747	83,222	580,352	0	205,211	319,495	170,482	94,387,991
	6	99,135,213	2,778	84,162	99,788	579,362	1,111	204,865	323,826	187,734	94,217,509
	7	98,931,562	2,778	84,566	116,307	578,273	1,111	204,483	328,078	204,823	94,029,775
	8	98,711,055	2,778	84,957	132,772	577,085	1,111	204,065	332,250	221,743	93,824,952
	9	98,473,769	2,778	85,335	149,173	575,798	1,111	203,612	336,340	238,490	93,603,209
	10	98,219,788	2,778	85,701	165,503	574,414	1,111	203,124	340,346	255,058	93,364,719
Ending Cash Flow	350	1,062,472	2,778	103,421	5,506	6,833	1,111	18,997	352,932	287,088	40,895,105
	351	951,544	2,778	103,217	4,932	6,182	1,111	18,810	361,482	302,470	40,895,105
	352	841,421	2,778	102,983	4,363	5,534	1,111	18,626	369,909	317,588	40,895,105
	353	732,137	2,778	102,709	3,797	4,892	1,111	18,442	378,212	332,442	40,895,105
	354	623,738	2,778	102,385	3,237	4,255	1,111	18,260	386,386	347,031	40,895,105
	355	516,285	2,778	101,994	2,681	3,622	1,111	18,079	394,427	361,351	40,895,105
	356	409,866	2,778	101,510	2,130	2,995	1,111	17,900	402,333	375,404	40,895,105
	357	304,617	2,778	100,886	1,586	2,375	1,111	17,723	410,100	389,186	40,895,105
	358	200,763	2,778	100,028	1,047	1,761	1,111	17,547	417,725	402,668	40,895,105
	359	98,763	2,778	98,705	518	1,155	1,111	17,374	425,205	415,936	40,895,105
	360	0	2,778	95,985	0	560	1,111	17,204	432,536	428,901	40,895,105

FIGURE 8.7 The output sheet has three distinct sections: Scenario Assumptions, Cash Flow snapshot, and Charts.

MODEL BUILDER 8.7: CREATING THE OUTPUT REPORT

1. The first step to creating the Output Report is to make a section on the Inputs sheet for reporting parameters. On the Inputs sheet, enter these labels in the following cells:
   G28: **Reporting Parameters**
   G29: **Scenario Name**
   G30: **Beginning Snapshot Start**
   G31: **Ending Snapshot Start**
2. For now, enter **Base Case** in cell I29 and name it **ScenName**. Enter **0** in cell I30 and name it **SnapshotStart**. Finally, enter **350** in cell I31 and name it **SnapshotEnd**. These cells are used later to control certain parts of the output report.
3. Create a new worksheet and name it **Output**. Since this sheet is primarily referenced to other parts of the model, the step by step will work differently than the other Model Builder sections. Rather than explain each cell, when most of them are references to concepts and calculations already explained, this section will rely heavily on the already completed version of file *MB8-7.xls* in the Ch08 folder on the CD-ROM.
4. In the completed version of Model Builder 8.7, notice that the first nine rows consist primarily of assumptions. A reader instantly knows which transaction the report has been created for by looking at the top left in bold letters. At the top right, the scenario version is readily visible. Below that the first few boxes a reader sees are inputs relating to the assets, liabilities, and structure. Almost all of these are references that should be linked up individually for the model under construction. Create all of the references for the first nine rows exactly the same as in the completed version. There are a few cells with formulas that require further explanation.
5. A unique cell on the Output sheet is I5, the gross cumulative loss. Instead of a cell reference to the default rate, this is a calculation of the gross cumulative dollar defaults by the original asset balance. In this case the gross cumulative loss percent is .97 percent, while the default rate is 1.0 percent. The difference can be occurring due to seasoning, amortization timing, and/or loss timing. Understanding this nuance will be discussed further in Chapter 9, Understanding the Model. For now it is important to realize that the gross cumulative loss percent should not be a direct reference.
6. Another unique cell is O4, which is labeled Req. Cr Enhmnt. In a senior subordinated structure with one senior tranche, this is the amount of credit enhancement that is necessary to absorb the expected losses. For now enter:

   $$= 1-\text{F4}$$

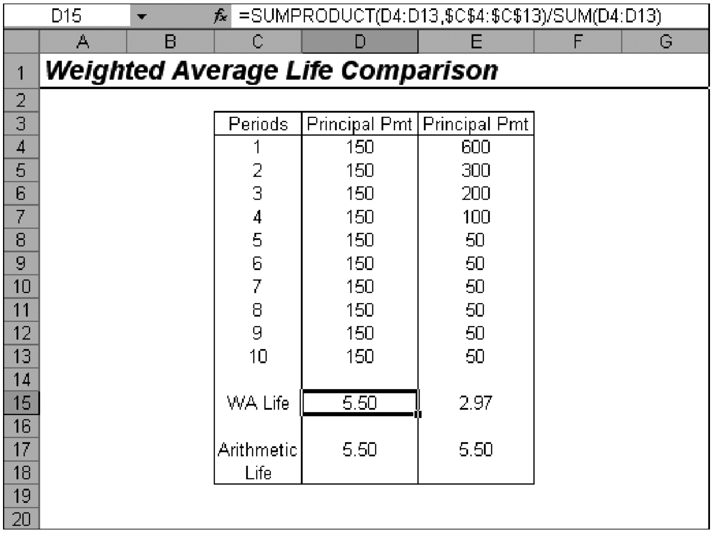
   This simply subtracts the senior advance rate from 100 percent and suggests that anything below the senior advance rate is the amount necessary to cover the expected losses. This is precisely how the senior tranche should be sized. The next chapter will discuss sizing the senior tranche in more detail.

7. Cells O5 and O6 contain formulas for weighted average life ("WAL"), a metric that is used frequently. This calculation was left off of the Analytics sheet because it does not rely on discounted cash flows. Instead WAL is a calculation based on the amortization of the assets and liabilities over time. It expresses the number of periods it takes to amortize an asset or liability, weighted by periodic reductions of balance.

   Imagine an asset that has equal periodic reductions in balance of 150 and pays off in 10 periods. In this case a total of 1,500 has been paid down equally over 10 periods. The WAL would be 5.5 periods, which is just the same as taking the average of the number of periods. However, if the assets paid down with 600 in the first period, followed by 300 in the second, 200 in the third, 100 in the fourth, and 50 for the remaining six periods, the WAL would be 2.97 periods. In such a case, the balance was reduced by a large amount earlier and has "died off" faster. See Figure 8.8 and the *WALifeComparison.xls* file in Ch08's Additional Files subfolder on the CD-ROM for details.

   Prior to inputting the formulas in cells O4 and O5, a column tracking all of the reductions in asset balance is necessary on the Cash Flow sheet. Go to the Cash Flow sheet to column CN. Enter **Asset Amort Total** in cell CN4. Then in cell CN7 enter:

$$= N7 + Q7 + R7$$



D15 =SUMPRODUCT(D4:D13,$C$4:$C$13)/SUM(D4:D13)

Weighted Average Life Comparison

Periods	Principal Pmt	Principal Pmt
1	150	600
2	150	300
3	150	200
4	150	100
5	150	50
6	150	50
7	150	50
8	150	50
9	150	50
10	150	50
WA Life	5.50	2.97
Arithmetic Life	5.50	5.50

FIGURE 8.8 The WAL is a measure of how long an assets balance is outstanding.

This formula adds up any amount that reduces the asset balance: defaults, scheduled principal, and voluntary prepayments. Copy and paste this formula over the range CN7:CN366.

8. Calculating WAL is a straightforward weighted average formula that uses the SUMPRODUCT-SUM combination seen in previous chapters. Back on the Output sheet, in cell O5, enter:

   = SUMPRODUCT('Cash Flow'!CN7:CN366,'Cash Flow'!A7:A366)/
   SUM('Cash Flow'!CN7:CN366)/12

   This weights the periods by the asset reduction each period and then provides an average. Also notice that there is a 12 as a divisor since the WAL is typically presented in years and this model is being calculated on a monthly basis.

9. The WAL should also be calculated for the debt. On this output sheet only the senior debt has the WAL calculated. Since the only item that reduces debt balance is principal, an additional column on the Cash Flow sheet is not necessary. In cell O6 enter:

   = SUMPRODUCT('Cash Flow'!CD7:CD366,'Cash Flow'!A7:A366)/
   SUM('Cash Flow'!CD7:CD366)/12

10. The last unique formula for the upper portion of the Output sheet tracks when any trigger has been tripped in the model. To accomplish this functionality, a single column on the Cash Flow sheet that tracks whether or not any trigger has been tripped in any period needs to be created.
    Go to the Cash Flow sheet to column CO. Label cell CO4 **Combined Trigger Tracking**. Then in cell CO7 enter:

    = OR(Z7:AB7)

    This will return a TRUE or FALSE depending on whether any of the triggers in Z, AA, or AB have been tripped. Copy and paste cell CO7 over the range CO7:CO366.

11. Once the periodic tracking has been set up, a formula needs to return the first period that the trigger tracking becomes TRUE. This can be done using an array formula. If array formulas are new, see the Toolbox section at the end of this chapter. The array formula used in cell O9 should be:

    {= MIN(IF('Cash Flow'!CO7:CO366,'Cash Flow'!A7:A366,1000))}

    The inside part of this formula is an IF statement that checks each cell in the range CO7:CO366. If any of those are true the associated period in column A is returned, otherwise a very large value that exceeds the maximum number of periods is returned (1,000). With a string of periods where the trigger is tripped the earliest one can be found using the MIN function. Remember to press **Ctrl + Shift + Enter** when entering an array function.

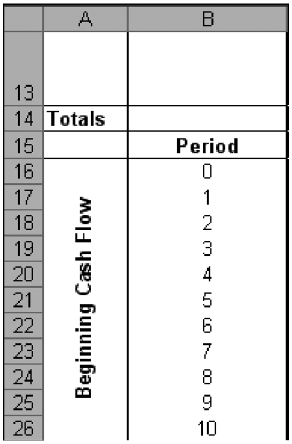
12. The middle part of the output report provides a snapshot of the cash flows. Reams of complete cash flows are rarely looked through, but specific periods such as the beginning and end are often examined. To do this a system needs to be set up that allows a user to select the periods of the cash flows to view.

Earlier, a Reporting Parameters section was added to the Inputs sheet, where cells I30 and I31 contain controls to change the first period for the beginning and ending views of the cash flows. Go to the Outputs sheet to cell B16. This will be a direct reference to cell I30 on the Inputs sheet (named SnapshotStart). On the Outputs sheet, cell B17 is the next period, which is the previous period plus one. Enter the following formula in cell B17:

= B16+1

Make sure not to use the named range because when cell B17 is copied and pasted over the range B16:B26 the reference needs to change. By now B17:B26 should look like Figure 8.9.

13. Still on the Outputs sheet, go to C16. The completed version shows the beginning balance for this column. It should be noted that any of the cash flow columns can be shown on the Outputs sheet and that the ones used in the completed version are just being used because they are the most frequently used. To get the



	A	B
13		
14	**Totals**	
15		**Period**
16	**Beginning Cash Flow**	0
17		1
18		2
19		3
20		4
21		5
22		6
23		7
24		8
25		9
26		10

FIGURE 8.9 The Cash Flow snapshot relies on a select group of periods determined by the model operator.

beginning balances of the assets for the periods used in the snapshot viewer, use an OFFSET function. Enter the following formula in cell C16:

= OFFSET('Cash Flow'!$V$6,B16,0)

This formula offsets the cell above the first period of the asset balance on the Cash Flow sheet by the period used in the snapshot viewer (cell B16). As a user changes the starting period for the snapshot beginning, the asset balance will be offset commensurately. Copy and paste cell C16 over the range C16:C26.

This can be repeated for as many columns as needed. Look to the completed Output sheet to see some other popular columns that are reported. Also notice that there are sums for each column that are references to the Cash Flow sheet directly. This is because the sums should be for all of the periods and should not just sum the snapshot view.

14. Complete the same process for the ending snapshot view in rows 28 to 38 on the Output sheet. Leave row 27 blank so it is clear where the beginning snapshot ends and where the ending snapshot begins.
15. The final section of the output report is the bottom third, which should contain charts of data over time. Examples of such data include: interest rates, prepayment rates, default rates, balances, excess spread, and so on. Some of the charts may not have source data readily available and will require more work on other sheets. For example, in the completed version of the output report there is a chart that tracks cumulative defaults. The source data for this chart is from column CP on the Cash Flow sheet.

The Importance of Testing and Output

While it may seem tedious to implement all of the tests and the output report, the time saved in the long run is much greater. Whenever building a model keep the mindset that it will be used for multiple scenarios. Each time the model assumptions are changed an operator wants to be sure that the results are accurate, logical, and easy to share. Tests ensure accuracy and logic, and should be viewed easily. Similarly, the output report should be comprehensive enough so someone who has not seen the model can understand the intent, execution, and results of each scenario the model produces.

TOOLBOX

Conditional Formatting

Conditional formatting automatically changes the formatting of cells depending on parameters that a user sets up. This can be especially useful to quickly identify important changes that occur during model operation.

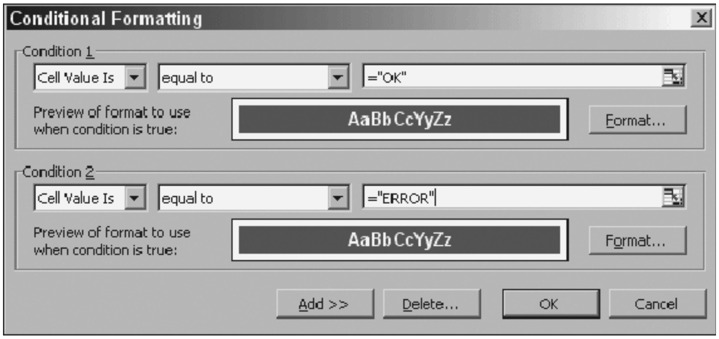
To use conditional formatting, select the desired cell, click **Format** on the menu bar and then click **Conditional Formatting**. The Conditional Formatting dialog box appears as shown in Figure 8.10.

The Conditional Formatting dialog box allows for a number of parameters. First, the format can depend on the value of the cell. This is set up by selecting **Cell Value Is** in the first Condition 1 list box and then a series of corresponding parameter options in the other list boxes. The other option is to select **Formula Is** in the first Condition 1 list box and select a formula. Multiple formats can also be set up for a single cell. This is particularly useful in conjunction with IF statements to highlight a change in true or false values.

Goal Seek

Excel provides the Goal Seek tool that allows a user to perform single parameter optimization operations. Goal Seek works by taking in three assumptions: a formula that needs to be optimized, the value that the result of the formula should be optimized to, and a variable that changes which allows the original formula to be optimized.

The Goal Seek dialog box shown in Figure 8.11 is opened by clicking **Tools** in the menu bar and then **Goal Seek**.

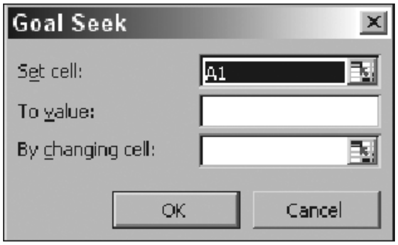


Condition 1: Cell Value Is | equal to | ="OK"

Condition 2: Cell Value Is | equal to | ="ERROR"

Conditional Formatting
Preview of format to use when condition is true:
AaBbCcYyZz
Format...
Preview of format to use when condition is true:
AaBbCcYyZz
Format...
Add >>
Delete...
OK
Cancel

FIGURE 8.10 Conditional Formatting dialog box.

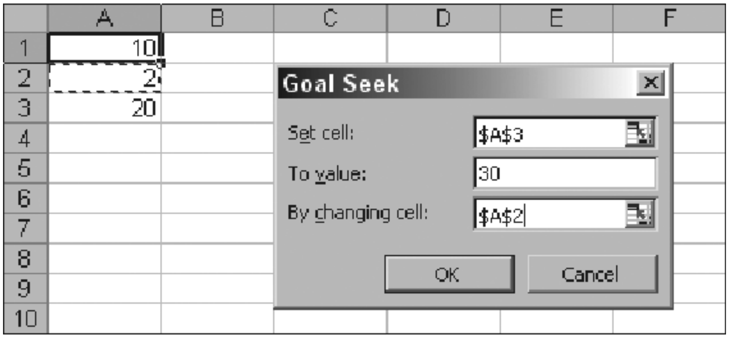


Set cell: A1

To value:

By changing cell:

FIGURE 8.11 Goal Seek dialog box.



	A
1	10
2	2
3	20

Goal Seek

Set cell: $A$3

To value: 30

By changing cell: $A$2

FIGURE 8.12 Enter the cell references in the Goal Seek dialog box.

As an example, if a sheet had 10 in cell A1, 2 in cell A2, and the formula A1*A2 in A3, Goal Seek can be used to figure out an optimal solution. With the current setup, cell A3 is equal to 20. If one wanted cell A3 to equal 30 by changing the value in cell A2, Goal Seek could be used. In that case, the references and values in Figure 8.12 would be entered in the dialog box. Also refer to *GoalSeekExample.xls* in Ch08's Additional Files subfolder on the CD-ROM.

As will be seen later in this text, Goal Seek can be used on an entire model with many formulas connected to each other. Also, Excel contains a more advanced optimization tool called *Solver* when multiple parameters need to be entered.

Array Formulas

Array formulas are one of the more tricky aspects for Excel users to learn. The main reason people have difficulty with them is because they reference ranges of cells, which can be challenging to visualize. An array function allows a user to perform custom operations on arrays of equal lengths.

Deconstructing the array function used in this chapter is an excellent method to demonstrate how they work. In this chapter the following array function is used:

{= MIN(IF('Cash Flow'!CO7:CO366,'Cash Flow'!A7:A366,1000))}

The array function here first performs an IF statement on the range CO7:CO366. The values for column CO are true or false values. In a regular IF statement, a single cell value is tested to return one of two answers. In the array function version, every cell in the array is tested against a parameter or range of parameters and returns an equal length result array. If cell CO7 is false, for example, then 1,000 is returned as the first value of the result array. If cell CO8 is true, then A8 is returned as the second value of the result array. However, if cell CO9 is false, then cell A9 will be returned as the third value of the result array. The first three values of the result array can be visualized as 1,000, 1, 2. The next part of this array function takes the MIN of that array, which is 1.

There are two important rules to remember when using an array function. Any arrays referenced within the formula must be of equal length; otherwise an error will be returned. Also, array functions must be entered by pressing **Ctrl + Shift + Enter** instead of simply Enter. An array function that is successfully entered can be identified with the { } symbols surrounding it.

CHAPTER 9

Understanding the Model

Each individual chapter has focused on explaining one concept of a structured finance model and the mechanics behind implementing that concept in Excel. At this point, the model is mechanically complete and many individual sections have been covered. However, the ability to integrate assumptions into interconnected concepts and produce interpretable results is where the true value of a financial model resides.

So, what happens when asset losses increase? Will the senior bonds receive all of the scheduled interest and principal? What stress scenarios can the transaction handle? All of these questions require an understanding of how each individual component of the model works and how all of those elements work together as a whole.

The best method of understanding a model is by changing individual assumptions one by one and evaluating the results. This allows a model operator to witness the cause and effect of each assumption. It is particularly valuable to set the assumptions to reasonable extremes so that result differences are more evident. This section first reviews the model as a whole and then walks through the results of changing each of the major assumptions.

THE COMPLETE MODEL IN REVIEW

A top-down approach separates the model into two distinct sections: assets and liabilities. On the asset side, cash is generated over time by yield, scheduled amortization, voluntary prepayments, and default recoveries. All of these methods of cash flow generation have assumptions that can vary greatly. However, for every set of assumptions for a scenario there is only a finite amount of cash.

The cash is used to pay for liabilities that set up the deal and fund the assets. Typical liabilities include fees, debt interest, and debt principal, which get paid in very specific orders and amounts depending on the cash flow waterfall. During each period and at the end of the legal final maturity date of the transaction, the finite amount of cash needs to be sufficient to cover all of the liabilities, otherwise parties in the transaction can suffer a loss. The basic model overview is presented in Figure 9.1.

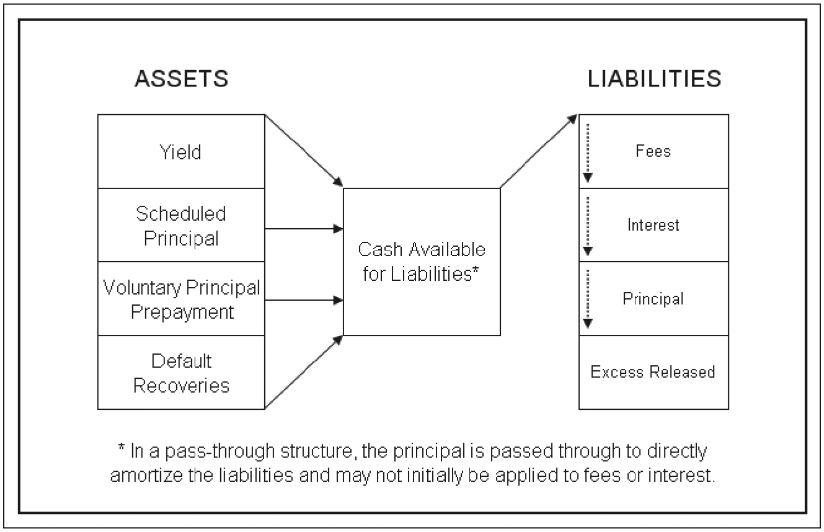

ASSETS
LIABILITIES
Yield
Scheduled Principal
Voluntary Principal Prepayment
Default Recoveries
Cash Available for Liabilities*
Fees
Interest
Principal
Excess Released
* In a pass-through structure, the principal is passed through to directly amortize the liabilities and may not initially be applied to fees or interest.

FIGURE 9.1 In structured transactions cash is typically created and used in a standard order.

Loss is protected against in a number of ways, predominantly through the use of credit enhancement. The top forms of credit enhancement include:

- Excess spread
- Reserve accounts
- Financial guaranties
- Overcollateralization

Excess spread is generated by the assets yielding more than the cost of the liabilities. This excess is typically the first form of enhancement used to cover loss. This coverage works in the model by reducing the asset balance by losses and attempting to reduce the debt by a commensurate amount. Since no cash is generated by defaults, the excess is used to maintain the commensurate debt reduction.

If excess spread is not sufficient to cover the liabilities, a *reserve account* can be drawn upon if set up at deal inception. Reserve accounts are either prefunded or grow to certain levels with excess cash. A prefunded reserve account gives comfort since the cash exists and is set aside, however it is an inefficient use of funds. Whenever a need occurs, the cash is withdrawn and typically reimbursed during better times.

A *financial guaranty* mechanically works the same as a reserve account and covers liability shortfalls. The terms of the guaranty are very important because they detail exactly which liabilities are covered, to what degree they are covered, and the priority of reimbursement to the insurer. Many of the monoline insurers that provide financial guaranties insure timely interest and ultimate principal. This means that if there is ever an interest shortfall during the transaction or a principal shortfall at final maturity then the monoline would cover the difference.

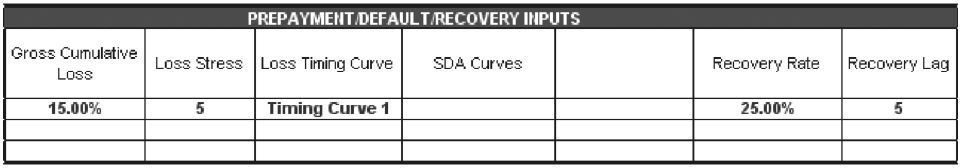
Finally, the concept of *overcollateralization* is the ultimate safeguard to certain investors. Usually the assets are funded by more than one party, one of which assumes a riskier position. The senior party often has structural safeguards that allow for principal repayment priority over the subordinate debt holders. The senior debt holders model their risk by the ability to be repaid by final maturity. They do not care if the subordinate debt holders are repaid. In a stress situation, cash would be directed to the senior holders and typically the subordinate holders would be locked out. This could create a scenario where the senior debt holders are repaid, but the subordinates are not.

Whether or not senior debt holders are repaid largely depends on the percentage of debt relative to the subordinate holder. Any time a senior tranche advances less than 100 percent of the assets, with a subordinate piece making up the difference, the senior tranche is termed *overcollateralized*. In Project Model Builder, the senior debt was initially structured at 95 percent of the assets, with the subordinated debt making up 5 percent. This 95 percent can be considered the amount that the senior debt holders advanced against the assets, otherwise known as the *advance rate*. Because senior debt is often at a lower cost than subordinate debt, many analyses focus on maximizing the senior advance rate. If the senior notes start at $95,000,000, at least that amount must be generated by the assets through scheduled principal, prepayments, and excess spread. If a scenario models losses at a rate where excess spread and reserve accounts are exhausted and the senior debt is still not repaid, the only other option is to rerun the scenario with a lower advance rate. This process becomes iterative until the optimal advance rate is determined.

UNDERSTANDING THE EFFECTS OF INCREASED LOSS

To observe the effect that increased loss has on a transaction, a few changes need to be made to the assumptions in Project Model Builder. Use the model under construction or the file *MB9-1.xls* in the Ch09 folder on the CD-ROM for this chapter. Change the gross cumulative loss rate (pdrCumLoss1) to 15 percent, the loss stress (pdrLossStress1) to 5, the timing scenario (pdrLossTime1) to Timing Curve 1, and the recovery rate (pdrRecovRate1) to 25 percent. The assumptions should be the same as in Figure 9.2. This is a very stressful scenario where a large amount of credit enhancement is necessary.

The major change that has taken place in this scenario is that the assets are assumed to have a base loss rate of 15 percent, which is stressed five times. The loss is also assumed to be distributed evenly over 360 periods based on timing curve 1. This means that a pool of unseasoned assets would expect to lose 75 percent of its principal to defaults over the life of the transaction. However, take a look at cell I5 on the Output Sheet. Notice that the gross cumulative loss is only 47.42 percent. This is because the assets are completely amortized by period 229 (noticeable in column



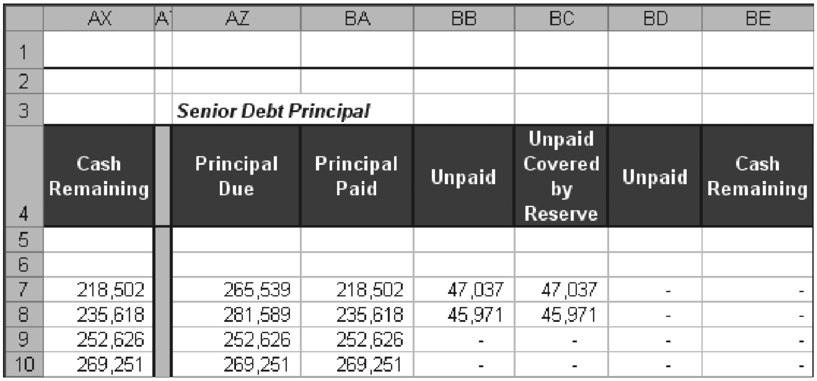
PREPAYMENT/DEFAULT/RECOVERY INPUTS						
Gross Cumulative Loss	Loss Stress	Loss Timing Curve	SDA Curves		Recovery Rate	Recovery Lag
15.00%	**5**	**Timing Curve 1**			**25.00%**	**5**

FIGURE 9.2 Change the assumptions to produce a very stressful loss scenario.

L of the Cash Flow sheet) and avoid a good portion of the loss curve. Always keep in mind how much loss is actually taken since prepayments and defaults themselves alter assets' amortization schedules.

The first source of credit enhancement to protect against this loss is excess spread. Notice that the assets are generating 9.00 percent, while the all-in cost of the liabilities is only 7.50 percent (Swap rate of 4.00% + Liability margin & fees of 1.50% + Asset servicing fees of 2.00%). The excess yield generated can make up for some of the lost principal in the transaction. Go to the Cash Flow sheet to cell AZ7. This is the principal due calculation for the senior debt. With no loss, there would be enough principal flowing through the waterfall to exactly meet this amount. However, with the loss amount assumed there is going to be a major shortfall in the next column when the debt is attempted to be paid. This shortfall is mitigated by the excess yield that is flowing through the waterfall and that is part of the Cash Remaining in cell AX7. See Figure 9.3 for more detail.

The next source of credit enhancement that is available to the senior debt is a reserve account. Check to make sure that the reserve account percentage (RsrvPercent) is set to 1.00 percent. This indicates that a prefunded reserve account was set up at the start of the transaction. Look at cell BC7 on the Cash Flow sheet. There should be amounts that are drawn from the reserve to cover the shortfalls



	AX	A	AZ	BA	BB	BC	BD	BE
1								
2								
3			*Senior Debt Principal*					
4	Cash Remaining		Principal Due	Principal Paid	Unpaid	Unpaid Covered by Reserve	Unpaid	Cash Remaining
5								
6								
7	218,502		265,539	218,502	47,037	47,037	-	-
8	235,618		281,589	235,618	45,971	45,971	-	-
9	252,626		252,626	252,626	-	-	-	-
10	269,251		269,251	269,251	-	-	-	-

FIGURE 9.3 Shortfalls due to loss are first covered by excess spread and then other forms of credit enhancement such as reserve accounts.

created by the loss. Also, go to the reserve account section (columns BG through BL) on the Cash Flow sheet and notice how it is drawn down each period until it is exhausted.

Since there is no financial guaranty modeled in Project Model Builder the only form of credit enhancement left is overcollateralization. The advance rate is set to 95 percent, which means that given the asset size the senior notes initially start at $95,000,000. With $100,000,000 of assets the senior notes can have $5,000,000 lost and still be repaid. The $5,000,000 would be the nonrepayment of the subordinated notes.

However, even with the excess spread, the reserve account, and the overcollateralization, there is still not enough cash in the structure to pay the senior debt by final maturity. Look at cell CB366 on the Cash Flow sheet. This is the last possible period for the senior notes to receive a payment. At this period there is still a balance of ~$34.9 million for the senior notes, which will most likely result in a loss to the senior debt holders. If such a scenario is deemed probable, what assumptions can be changed to prevent it from occurring?

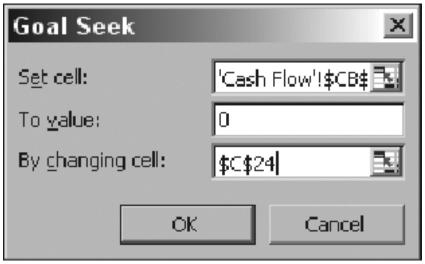
Excess spread is difficult to change. The asset yield can only be increased if the issuer has higher yielding assets that can be added to the pool. However, higher yielding assets are typically more risky and a new static loss analysis should be completed to see if the loss expectation needs to be increased. The liability rates are usually set to market rates which are set by the market. The other fees could possibly be negotiable, but would probably only have a marginal impact.

The reserve account could be increased, but this is an inefficient use of cash. A reserve account locks cash up, typically in *guaranteed investment contracts* (GICs), which earn a low rate of interest. This cash could be better used to create more assets.

In such a loss case, the most often-changed assumption is the overcollateralization amount dictated by the advance rate. Notice the amount due to the senior debt in the final period. Now change the advance rate for the senior debt to 92.5 percent. Because the amount issued by the senior entity is reduced, and the same amount of cash being is still being generated by the assets, the amount due to the senior debt in the final period is reduced. The advance rate can be continuously reduced (86.01 percent in this scenario) until there is no amount due to the senior debt in the final period and the notes are considered to be paid off.

An effective means of finding the optimal advance rate is by using the Goal Seek tool. The "Set cell" should be the final period senior principal balance (cell CB366). It's "To value" should be 0. Finally, the "By changing cell" should be LiabAdvRate1 from the Inputs sheet. When this is run the advance rate will be minimized until the senior debt balance at the final period is zero. Figure 9.4 shows this in detail.

There are instances where the Goal Seek tool finds a solution that does not seem optimal. This may be because the maximum number of iterations is set too low. The option for the number of iterations that Goal Seek runs is controlled with the



Set cell: 'Cash Flow'!CB

To value: 0

By changing cell: $C$24

FIGURE 9.4 The Goal Seek tool can be used to find the optimal advance rate.

following steps. On the Excel menu bar, click **Tools** and **Options**. Then click the **Calculations** tab and select the **Iterations** option and, if necessary, in the Maximum iterations box enter **100**.

VARYING PRINCIPAL ALLOCATION METHODOLOGIES

Recall that there were two options for principal amortization built into the model: sequential and pro rata pay. The previous scenario assumes a pro rata pay scenario where the senior debt receives a pro rata share of the asset principal. This is less advantageous than a sequential pay scenario, because in such a scenario principal is paid to the senior debt first, until the senior debt is amortized. Observe the difference by changing the Principal Allocation Type on the Inputs sheet from pro rata to sequential—and remember to change the advance rate back to 95 percent. Now, instead of the senior debt having a balance of ~$34.9 million, the debt has a balance of ~$34.8 million. This difference is very slight; but this is largely due to the existing triggers that are set up in the model to direct cash to the senior notes. Try taking the triggers out to see how much of an effect the principal allocation method has on the amortization of the senior debt.

The difference is caused by the calculation and release of cash to pay the subordinated debt on a pro rata basis. The senior debt's periodic principal due is less than it would be under a sequential pay structure. Since the senior note pays down slower there is a higher interest cost and a higher balance in the final period. Also, until a trigger is breached, cash flows to pay the subordinated interest and principal.

A trigger is breached in both scenarios. Aside from the trigger that is preset to rapidly amortize the deal in period three, go to column AC on the Cash Flow sheet and look at period 25 (row 31). In that period, the default trigger turns on. This is caused by the cumulative defaults exceeding the amount set up on the Inputs sheet (Trigger_Def). When the trigger is tripped, all cash that remains up to the senior principal payment is used to amortize the senior principal. Notice that cell AZ31 on the Cash Flow sheet calculates the senior principal due amount as everything that is remaining (AX31). If the trigger amount were decreased from 5.00 percent to anything lower, the trigger would trip earlier and direct more cash to the senior debt faster.

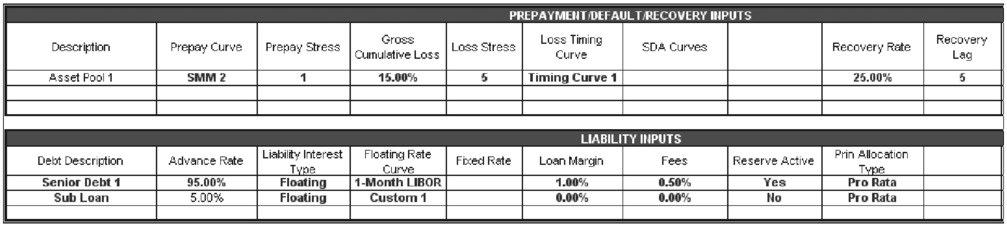
Varying Prepayment Rates

The importance of prepayments is particularly heightened during times of increased loss. Prepayments can have a dual effect in a high stress scenario. On one hand they can help avoid parts of the loss curve by accelerating the amortization of the assets. While on the other hand the loans that are expected to prepay are the better credits who can switch to a lower interest rate, thereby decreasing excess spread and creating a higher concentration of poorer performing loans.

In the high loss example currently set up in Project Model Builder enter a **1%** SMM for all periods in column N on the Vectors sheet. Set the prepayment curve on the Inputs sheet to SMM 2. Make sure the assumptions are the same as in Figure 9.5.

Now check the senior debt's final period balance. Instead of ~$34.9 million at the end there is now only ~$14.4 million. This is primarily caused by avoiding the tail of the projected loss curve. Notice that the sum of all of the new defaults is ~$34.6 million in the high prepay scenario. Previously the sum was ~$47.4 million. Also contributing to the reduction in debt balance is a savings in senior debt interest due to the faster amortization of the debt. Compare the two scenarios total debt interest to notice the difference.

The opposite effect is possible in a more granular analysis. If the model were set up using a loan level analysis, certain obligors would be expected to prepay over others. Obligors with high interest rates and good credit would have an increased propensity to prepay than lower interest rate loans. This causes a decrease in the average yield of the assets termed *weighted average coupon deterioration* (WAC deterioration). With a decrease in asset yield over time, excess spread is reduced and sometimes a greater level of credit enhancement is necessary to make a transaction work.



PREPAYMENT/DEFAULT/RECOVERY INPUTS

Description	Prepay Curve	Prepay Stress	Gross Cumulative Loss	Loss Stress	Loss Timing Curve	SDA Curves		Recovery Rate	Recovery Lag
Asset Pool 1	**SMM 2**	**1**	**15.00%**	**5**	**Timing Curve 1**			**25.00%**	**5**

LIABILITY INPUTS

Debt Description	Advance Rate	Liability Interest Type	Floating Rate Curve	Fixed Rate	Loan Margin	Fees	Reserve Active	Prin Allocation Type	
Senior Debt 1	**95.00%**	**Floating**	**1-Month LIBOR**		**1.00%**	**0.50%**	**Yes**	**Pro Rata**	
Sub Loan	5.00%	**Floating**	**Custom 1**		**0.00%**	**0.00%**	**No**	**Pro Rata**	

FIGURE 9.5 Make sure all of the assumptions on the Model Builder are the same as in the figure.

VARYING LOSS TIMING

Project Model Builder was built with five possible loss timing scenarios. In general, there are three patterns that a transaction should be tested against: base case losses, front-loaded losses, and back-loaded losses. In Project Model Builder, Timing Curve 3 is clearly the front-loaded case and Timing Curve 5 is the back-loaded case. Assume that Timing Curve 1 is the base case.

Continuing with the assumptions from Figure 9.5, change the Loss Timing Curve (pdrLossTime1) to Timing Curve 3. Now take a look at the final period senior debt balance. It jumps to ~$66.9 million! This is because less of the loss curve is being avoided by the faster amortization of the assets. If Timing Curve 1 is used a large part of the default curve is avoided because the assets amortize by period 229. Also, the upfront defaults deteriorate the assets from the start, which prevents excess spread from generating early in the transaction. With unseasoned assets most front-loaded loss curves will be very damaging to a transaction.

Now switch the Loss Timing Curve to Timing Curve 5. The final period senior debt balance falls to zero. A major reason for this is that the assets have amortized prior to taking the most strenuous part of the loss curve. Notice that the assets are amortized by period 265, but according to Timing Curve 5 one would only expect to have taken less than 30 percent of the loss curve through that timeframe. With ~70 percent of the loss curve avoided the gross cumulative loss is greatly reduced.

Loss timing is very important for the periodic flow of cash. Triggers can be tripped in certain timing scenarios, leading to changes in debt amortization methods. Reserve accounts can be emptied earlier or later. Total interest can be higher or lower. All of these effects can alter the weighted average life of the debt, vary the overall debt yield, and possibly cause debt holder loss.

VARYING RECOVERY RATE AND LAG

The most observable mitigant to loss is recovery. For structured finance transactions, recovery is a source of cash that is dependent on rates and time. Put the loss timing back to Timing Scenario 1 and change the recovery rate from 25 percent to 75 percent. It is clear from the TEST section of the Inputs page that the debt is paid off by maturity. (Not that the Inputs cell L6 switches from ERROR to OK) By increasing the recovery rate there is additional cash in the transaction. Recovery cash is purely for the benefit of the transaction and can be thought of as additional yield since it does not reduce the asset balance.

The other component of recovery is the amount of time it takes to receive the cash. This is important because as the recovery lag increases, the recovery benefit to the transaction diminishes. Change the recovery lag from 5 to 40. By doing this, the TEST section shows that the senior debt balance is not paid off by final maturity. With a longer recovery lag the debt is paid down slower and accrues more interest than with a shorter lag.

THE VALUE OF A SWAP

Project Model Builder was created with a swap in place that fixed the liabilities to a 4 percent interest rate. Without such a hedge the interest rate on the notes would be subjected to the floating rate curve, which in the example is much higher than 4 percent. Keeping the 75 percent recovery rate, set the recovery lag back to 5 so that the senior debt pays off and then turn the Swap Active cell (Swap_Active) on the Inputs sheet to No. By doing this, the TEST section shows that the senior debt no longer pays off by maturity because of the increased interest cost without a swap.

ADDITIONAL TESTING

Every assumption that has been created in Project Model Builder has an effect on the results. Try changing each assumption, check the TEST and Output sheet to see the result, and then go over the Cash Flow sheet in detail to see how each section has changed. It is extremely important to understand cause-and-effect relationships because when actual transactions are modeled often times a result is returned that can be unexpected. These unanticipated results need explanations, which can only be done by going backward through the process to understand what assumptions caused the result.

CHAPTER 10

Automation Using Visual Basic Applications (VBA)

A powerful capability of all Microsoft Office applications is the ability to program actions using the Visual Basic Applications (VBA) programming language. For Microsoft Excel, VBA is especially useful for frequently used commands that require multiple procedures, repetitive actions, and in more advanced situations for calculations that exceed the spreadsheet's processing ability. Examples relevant to cash flow modeling include creating buttons that control print commands and goal seek functions, building a quick system to run multiple scenarios, or constructing an amortization engine that can generate and aggregate the cash flow for thousands of loans. Implementing such functionality requires a basic understanding of the VBA language and how the language interacts with Excel.

Most users have unknowingly used VBA by recording a macro to complete simple repetitive tasks. However, few take the step to learn how to write and edit VBA code by hand. The problem most users have with unlocking the full potential of VBA is learning how an object-oriented programming (OOP) language works. While entire books can and have been written on using VBA, this chapter intends to introduce the model operator to the basics of using VBA through additions to Project Model Builder and other relevant examples. Beginners may find additional texts helpful for further explanation, while intermediate to advanced users may want to skip to the specific code examples.

CONVENTIONS OF THIS CHAPTER

One of the most useful means of explaining VBA is through the use of code examples shown in this chapter. Recall that this is also the convention for named cells or ranges. Distinguishing between the two should not be difficult since the VBA code is typically written in blocks of text, while the named cells or ranges are written into the normal text.

THE VISUAL BASIC EDITOR

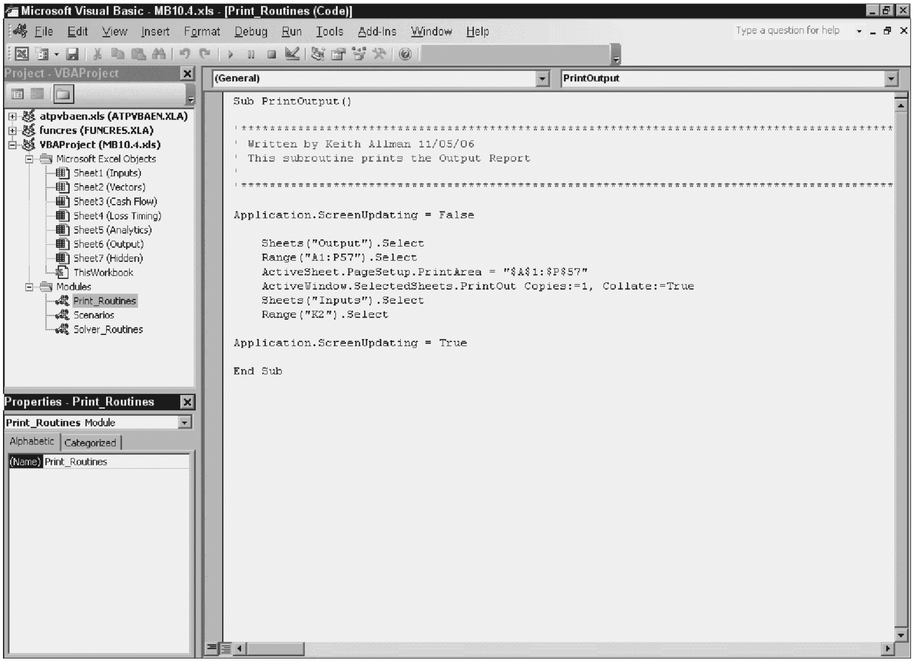
Coding in VBA is simplified by an interface called the Visual Basic Editor (VBE). VBA code can be written, stored, run, and debugged from the VBE. To access the VBE go to Tools, Macro, Visual Basic Editor or use the ALT-F11 keyboard shortcut. The VBE will open in a separate window and should appear as in Figure 10.1.

The Menu Bar

The general menu bar features recognizable commands such as File and Edit; however, most of the options within each command will seem strange to a new user. For now look at the Standard Toolbar that should appear as a default setting.

The Standard Toolbar has a few buttons that will be useful for the basic operation of the VBE. Keep in mind the following buttons, which can be seen in Figure 10.2:

- View Microsoft Excel: Jumps back to the Excel workbook
- Run Sub/UserForm: Runs the code currently selected
- Break: Breaks the code currently being run
- Reset: Resets the code after a break has occurred
- Object Browser: opens the library of VBA objects



```
Sub PrintOutput()

'********************************************************************************************
' Written by Keith Allman 11/05/06
' This subroutine prints the Output Report
'
'********************************************************************************************

Application.ScreenUpdating = False

    Sheets("Output").Select
    Range("A1:P57").Select
    ActiveSheet.PageSetup.PrintArea = "$A$1:$P$57"
    ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True
    Sheets("Inputs").Select
    Range("K2").Select

Application.ScreenUpdating = True

End Sub
```

FIGURE 10.1 The Visual Basic Editor (VBE).

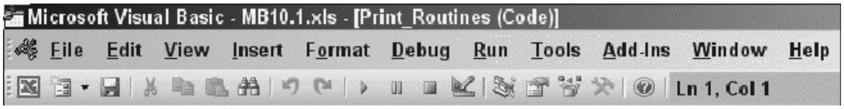


FIGURE 10.2 The Standard Toolbar looks similar to other Office Toolbars, but has unique buttons specifically for creating, editing, and running code.

The Project Explorer and the Properties Window

To the left side of the VBE there are two important windows: the Project Explorer and the Properties Window. The Project Explorer looks a little like Windows Explorer in the way it organizes information. It is set up as a directory tree where more detailed information within a general concept can be expanded or compressed by clicking on "+" and "−" symbols.

The most general category in VBA is a Project, which is essentially the Excel workbook and any associated additions created in the VBE. The first subfolder contains the Excel objects, which are the individual sheets in the workbook. Code can be stored under a sheet or for the workbook in general, but for this book's examples code will be created and stored in a module.

A module is a separate area to enter code. The code is often organized by purpose and functionality into individual sections called subroutines. Basic macros use one subroutine to accomplish a task, while more advanced macros often use multiple subroutines. Related subroutines are stored in the same module. For instance, a module might be named Print_Routines and contain three subroutines that format and print different sections of the Excel workbook. See Figure 10.3 for a detailed look at a module.

VBA CODE

Writing VBA code is like typing out a set of instructions using words and values that are specific to Excel. Trying to run code that Excel does not understand will generate an error and stop the subroutine from running. For a crash course in VBA, the most basic elements that a new programmer should know are objects, methods, and variables. While there are certainly more components to VBA, learning about

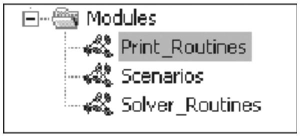


FIGURE 10.3 Modules organize code for subroutines and functions.

the three elements mentioned above will greatly aid a reader in understanding the example code in this chapter.

The first basic element of VBA is an object. In object-oriented programming objects are the building blocks of code. They are items that code performs tasks on and have properties that can be manipulated. Workbooks, worksheets, ranges, cells, and so on are all objects in VBA for Excel.

Objects are worked with primarily through the use of methods. The properties of an object are changed by using different methods or combinations of methods. For example, if a cell is an object and one wanted to change the background of a cell to yellow they would use the .Interior.ColorIndex method. There are hundreds of methods to learn, which are best picked up through examples such as those in this chapter.

Variables are the third basic element of VBA. They are particularly useful for understanding the examples in this chapter because most of the examples make heavy use of variables. A variable is a character or string of characters that are assigned a value. The designers of VBA created specific types of variables in order to save memory and allow a user to create a more efficient macro. For instance, a variable can be declared as a Boolean, meaning that the only acceptable value is "true" or "false." It is important to understand the different types of variables in VBA because if a programmer attempts to assign an inconsistent value to a variable type an error will be generated and the macro will stop running.

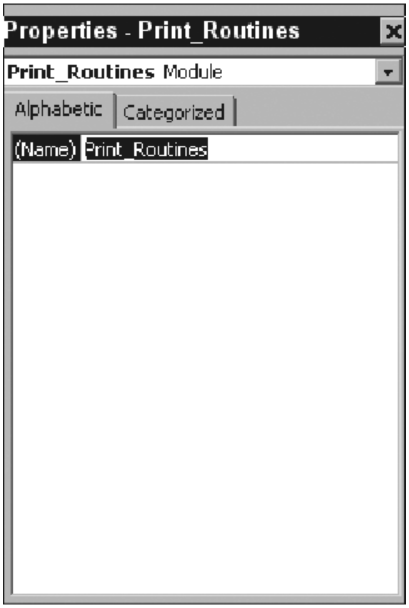
SIMPLE AUTOMATION FOR PRINTING AND GOAL SEEK

Print and Goal Seek are the two most commonly used menu tools while operating a cash flow model. Both take time and involve repetitive tasks. For printing there is always the concern that the print area has changed or that the page set up is different. Goal seek requires clicking through a number of fields and entering references and values. Both tasks can be quickly transformed into a single macro that can be run with the click of a button.

MODEL BUILDER 10.1: AUTOMATING PRINT PROCEDURES

1. Press **Alt + F11** to open the VBE. In the Project Explorer, find the name of the Excel workbook. The name of each workbook should be prefaced by VBAProject. Right-click the name, and then on the menu bar click **Insert**, and then click **Module**. This should create a file folder named Modules, with one module named Module1. Using the Properties window rename Module1 to **Print_Routines**. See Figure 10.4 for detail.
2. Double-click **Module1** in the Project Explorer. Select the main code window so there is a blinking cursor. Type the following code:

```
Sub PrintOutput()
```



(Name) Print_Routines

FIGURE 10.4 The Properties window changes the characteristics of items in the VBE.

Starting a module with Sub indicates the beginning of a subroutine, the name PrintOutput is a user created description of the subroutine, followed by an open and close parenthesis. After entering this code and pressing **Enter**, the VBE will automatically enter End Sub, indicating an end to the subroutine. Press a few hard returns after the Sub PrintOutput() so that there is space to enter the main body of code between that beginning and End Sub. The End Sub code should always be at the end.

3. The next step is to turn off screen updating, which displays the results of the macro as it is running. Screen updating slows down a macro considerably, so in most cases it should be turned off. Under the previous code enter:

   ```
   Application.ScreenUpdating = False
   ```

   This line of code is a perfect example of object/method interaction. The application is the object, which is affected by the screen updating method. Writing the object followed by a period and then the method is a typical object-oriented programming convention.

4. The main part of a print macro is designating the range to be printed, the correct page set up, and ordering the print. Under the previous code enter:

   ```
   Sheets("Output").Select
   Range("A1:P57").Select
   ```

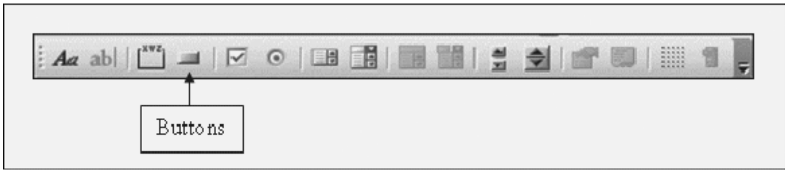


FIGURE 10.5 The Button function on the Form Toolbar is often used to control macros.

```
ActiveSheet.PageSetup.PrintArea = "$A$1:$P$57"
ActiveWindow.SelectedSheets.PrintOut Copies: = 1, Collate: = True
Sheets("Inputs").Select
Range("K2").Select
```

This section of the code selects the Output sheet, selects the entire Output range that needs to be printed, designates that range as the print area, and orders the area to be printed with a few page set up characteristics. Since this macro will be initiated from the Inputs sheet, the last bit of code instructs a cell on the Inputs sheet to be selected when the macro is done. This will prevent the screen from jumping to the Output sheet every time the macro is run.

5. Finally, make sure to activate screen updating before ending the macro by inserting the following under the previous code:

```
Application.ScreenUpdating = True
```

The End Sub that was automatically created should be at the very end.

6. The final step is to create a button on the Inputs page to run this macro quickly. The easiest method for creating macro buttons is using a Form object. Still on the Inputs sheet, make the Forms tool bar visible by selecting View, Toolbars, Forms. This will bring up a series of buttons that looks like Figure 10.5. Click the Button button labeled in Figure 10.5.

   After clicking the Button button, the cursor changes to a crosshair and allows the user to draw a rectangular button by left-clicking (while holding down the click) and dragging until the desired size is created. Make such a button near the G4 area of the Inputs sheet. Immediately upon finishing the click a dialog box will appear that instructs the user to assign a macro. Select the PrintOutputs macro and click **OK**. Finally double click on the name of the button (Button 1) and rename it **Print Output Sheet**.

Model Builder 10.1 Final Code:

```
Sub PrintOutput()
Application.ScreenUpdating = False
   Sheets("Output").Select
   Range("A1:P57").Select
   ActiveSheet.PageSetup.PrintArea = "$A$1 : $P$57"
```

```
    ActiveWindow.SelectedSheets.PrintOut Copies: = 1, Collate: = True
    Sheets("Inputs").Select
    Range("K2").Select

Application.ScreenUpdating = True

End Sub
```

MODEL BUILDER 10.2: AUTOMATING GOAL SEEK TO OPTIMIZE ADVANCE RATES

1. Similar to printing, using goal seek can be automated to work by pushing one button. Goal seek is a bit more complicated to automate because there are a few inputs that need to be entered. Also, certain steps need to be taken to ensure that the goal seek can find an acceptable solution each time. Go to the Project Explorer in VBE and insert another module. Rename the module **Solver_Routines**.
2. Start a new subroutine named **SolveAdvance**. This macro optimizes the senior debt advance rate. It is identical to the goal seek procedure done earlier, where the final senior debt balance is iterated to zero by changing the advance rate.
3. For macros that take a few seconds to run, a useful line of code to insert is one that provides progress information in the Status Bar. The Status Bar in Excel is on the bottom left and typically reads Ready as shown in Figure 10.6.
   The Status Bar can be changed when a macro is running to provide useful information to the model operator. After this line of code and until the Status Bar is changed in the code, the Status Bar will read "Optimize Advance Rate . . .". To do this, enter the following code under the subroutine name:

   ```
   Application.StatusBar = "Optimize Advance Rate . . . "
   ```

   Similar to changing screen updating, the object is the entire application and the method is the StatusBar. Here the message is a constant that is customized. On the line after this, enter the code to turn off screen updating.
4. The next step is to create string constants (constants are objects that are assigned values that do not change throughout the code) that are assigned range names from the Excel sheet. This step does not actually assign a numerical value or reference to the constants, it is assigning the literal text. The purpose of this

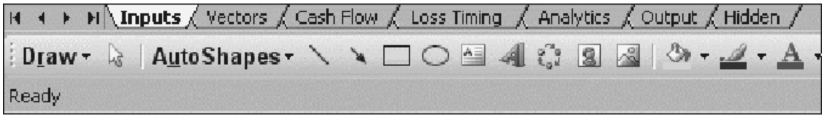


FIGURE 10.6 The Status Bar displays the status of the workbook.

becomes clear when the range names on the Excel sheet are changed. By using constants that are assigned the range name once in the beginning, any changes to the Excel sheet range names only have to be update once in the code. Otherwise any time the Excel sheet range is directly referenced in the code it would have to be changed, which is tedious and prone to errors. Under the previous code enter the following:

```
Const DebtBal As String = "FinalLoanBal"
Const AdvRate1 As String = "LiabAdvRate1"
```

The first line declares DebtBal as a constant. The variable type is a string, which means that any value assigned to DebtBal will be treated as text. Now anytime DebtBal is used in the code it actually has a value of "FinalLoanBal".

5. The next lines of code declare all of the variables:

```
Dim UnpaidLoan As Range
Dim AdvanceRate As Range
```

Variables are declared or dimensioned using the Dim command. A name is created for the variable, followed by the type of variable. In this case, two range variables are declared. It is important to declare variables, otherwise the variable will be declared as a memory intensive variant, which any type of data can be passed through.

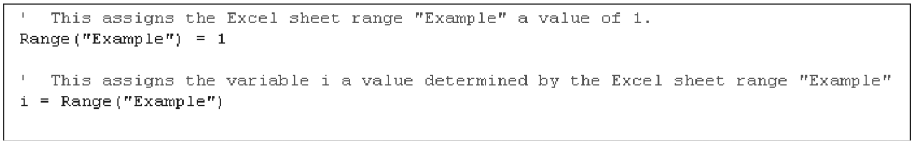
6. For any goal seek, it is important to set a realistic starting point for the cell that will be changing during each iteration. Occasionally goal seek is able to find a solution if the changing cell begins with an illogical value. In the case of optimizing the advance rate, the cell that is changing is the advance rate. Since 100 percent is the maximum that the advance rate can be, it makes sense to optimize down from that value each time. To make sure that the advance rate is always 100 percent at the beginning of each optimization, enter the following code after the declared variables:

```
Range(AdvRate1) = 1
```

Range(AdvRate1) is an object from the Excel sheet. Remember that AdvRate1 is a constant that is the equivalent to “LiabAdvRate1”. The VBE reads Range(AdvRate1) as Range(“LiabAdvRate1”), which is how Excel ranges are referenced in VBA. Recall that LiabAdvRate1 is the named range for the advance rate in Project Model Builder.

Values can be assigned to Excel ranges in this way by stating the range name, an equal sign, and then the value to be assigned. Similarly, a variable can be assigned an Excel value by doing the opposite. See Figure 10.7.

7. Prior to setting up the actual goal seek, code can be used to check to see if an optimal solution already exists, which would save calculation time. If the transaction can be run with a 100 percent advance rate, then that is the optimal solution. Since 100 percent has already been entered in the prior step, all that needs to be done is a check to see if the final senior debt balance is paid. Values



```
'  This assigns the Excel sheet range "Example" a value of 1.
Range("Example") = 1

'  This assigns the variable i a value determined by the Excel sheet range "Example"
i = Range("Example")
```

FIGURE 10.7 Assigning values is an important part of coding for financial applications.

can be checked using an If statement, much like on the Excel sheet. After the previous line of code enter the following:

```
If Range(DebtBal) > 0.01 Then
```

The If-Then construct in VBA works very similar to Excel. First the If statement is declared, followed by the test, and then a Then statement. The major difference is that an End If must be inserted at the end of the code that takes place when the statement is true. (See Model Builder 10.2 Final Code at the end of the exercise for placement.)

8. The next step is assigning values to the ranges that were declared earlier. These ranges will be used for the required range inputs for the goal seek. Below the previous line of code enter:

   ```
   Set UnpaidLoan = Range(DebtBal)
   Set AdvanceRate = Range(AdvRate1)
   ```

   Using the methodology just touched upon in step six, this code assigns values to the VBA ranges from Excel sheet ranges.
9. The actual goal seek command is a single line of code in VBA. Below the previous line of code enter:

   ```
   UnpaidLoan.GoalSeek Goal: = 0.01, ChangingCell: = AdvanceRate
   ```

   The object UnpaidLoan, which was earlier referenced and valued as the Excel range FinalLoanBal, is the goal range. The goal is set to .01 since the purpose of the exercise is to find the advance rate that completely pays off the loan. Zero is not used because occasionally goal seek will have trouble iterating to such a solution. Finally, the last part of the code designates the changing cell, which in this case is the advance rate.
10. Some final lines of code are necessary to set the Excel environment back to normal. After the previous code enter the following:

    ```
    Application.ScreenUpdating = True
    Application.StatusBar = False
    Calculate
    Application.Calculation = xlCalculationAutomatic
    End Sub
    ```

    The screen updating should be turned back on, the Status Bar should be set to false (this will set it back to Ready), a calculate should be performed to update all of the calculations in the model, and finally the model should be set to automatic calculation in case it was turned to manual during the goal seek. Also create a button near D4 on the Excel Inputs sheet named **Optimize Advance Rate** and assign the macro.

Model Builder 10.2 Final Code:

```
Sub SolveAdvance()
Application.StatusBar = "Optimize Advance Rate . . . "
Application.ScreenUpdating = False
Const DebtBal As String = "FinalLoanBal"
Const AdvRate1 As String = "LiabAdvRate1"
Dim UnpaidLoan As Range
Dim AdvanceRate As Range

Range(AdvRate1) = 1

If Range(DebtBal) > 0.01 Then

   Set UnpaidLoan = Range(DebtBal)
   Set AdvanceRate = Range(AdvRate1)
   UnpaidLoan.GoalSeek Goal: = 0.01, ChangingCell: = AdvanceRate
End If

   Application.ScreenUpdating = True
   Application.StatusBar = False
   Calculate
   Application.Calculation = xlCalculationAutomatic
End Sub
```

UNDERSTANDING LOOPING TO AUTOMATE THE ANALYTICS SHEET

Looping is one of the most robust processes that VBA allows. A loop allows code to be repeated for multiple iterations, while giving the option for unique changes during each iteration. Such functionality allows for sensitivity scenarios, loan level amortization, and ultimately stochastic modeling. In the following Model Builder example, a simple goal seek loop is created to perform analytics on the assets and each tranche of debt.

MODEL BUILDER 10.3: AUTOMATING GOAL SEEK TO PERFORM TRANSACTION ANALYTICS

1. Remember that on the Analytics sheet the yield was dependent on making the present value of the cash flows equal to the current values of the different assets

and liabilities. This was done using a goal seek on row 6 of the Analytics sheet (labeled PV Diff). Instead of having to do this by hand for both the assets and liabilities, a macro can be set up that completes all three automatically. The first step is to create a new subroutine in the Solver_Routines module called **SolveYield**.

2. Since this subroutine is essentially a goal seek very similar to the one in Model Builder 10.2, these instructions are condensed. The main point of this exercise is to demonstrate how a basic loop command can be very useful. The beginning of the macro should be familiar and start as follows:

```
Sub SolveYield()
Application.StatusBar = "Solving Analytics . . . "
Application.ScreenUpdating = False
   Const YieldChange As String = "rngYieldChange"
   Const Target As String = "rngYieldTarget"

   Dim YieldRange As Range
   Dim TargetRange As Range
```

3. The first unique part of the code is an additional variable that needs to be declared to assist in looping. This is a declaration of the loop counting variable. Insert this code directly below where the previous code left off:

```
Dim i as Integer
i = 1
```

The variable i is used in the loop construct to track the number of iterations. To ensure that the variable is cleared to its starting value, i is assigned a value of 1.

4. The next step is to activate the Analytics sheet since it contains information necessary to the macro:

```
Worksheets("Analytics").Select
```

5. The core of this macro is the next few lines of code that create a loop. The most common method to create a loop is using For–Next statements. This construct works by opening with the parameters of the loop using the For statement and looping through each parameter with the Next statement. For instance, by writing the code For i = 1 to 10, the parameters of the loop have been set so there can be 10 possible loops. The code below the For statement will continue to run until the variable i equals 10. Since a programmer only wants certain code below the For to be run during each loop, there needs to be a method for moving on to the next iteration. In the above example the code that moves back to the beginning of the For statement is Next i. This instructs the program to jump back to the beginning of the For statement, but with the next value for the variable (2 in this case).

   The example in Project Model Builder is made slightly more complicated because the number of loops depends on the number of cash flows where the present value needs to be optimized. In the current model, there is a set of cash flows for the assets and two tranches of debt, so three loops are necessary. However, if there was an additional tranche of debt, then four loops would be necessary; two more tranches of debt and five loops would be necessary, and so on.

This can be overcome by using the range that was created for the PV differences. Recall that the range rngYieldTarget was defined earlier as E6:G6 on the Analytics sheet. This range should always encompass all streams of cash flow that are to be analyzed. If an additional tranche of debt were in column H, then this range should be extended to column H. Since the range will always contain a number of cells equal to the number of cash flow streams to be optimized, a count method can be integrated into the For statement. Enter the following below the previous line of code:

```
For i = 1 To Range(Target).Cells.Count
```

The code begins like a standard For statement with For i = 1 To, but, instead of providing a numerical value, the reference Range(Target).Cells.Count is used. The range "rngYieldTarget" is being referenced here using the Target variable. Cells is referring to the individual cells within the range and the Count method counts those cells. Since there are three cells in the range, the For statement is effectively For i = 1 to 3.

6. The next few lines of code are mostly familiar, with a few new concepts. Enter this code below the previous line:

```
Set TargetRange = Range(Target).Cells(1, i)
Set YieldRange = Range(YieldChange).Cells(1, i)
TargetRange.GoalSeek Goal: = 0, ChangingCell: = YieldRange
Next i
```

Notice the addition of the Cells property that comes after Range(Target) and Range(YieldChange). Earlier Cells was used to count the number of cells; but it can also be used to reference individual cells within a range. By entering an open parenthesis right after the Cells property, VBA is instructed to look for a reference within the cells of the respective range. VBA follows an RC (rows, columns) convention, so a (1, 1) would be the first row and first column. In the above code the reference is (1, i), which means row 1 and column i. The variable i will take on a numerical variable depending on the loop iteration. This numerical value will also correspond to the column order in the ranges. Specifically during the first loop i will equal 1, making the reference (1, 1). For each range the first row and first column will be referenced. During the next iteration i will equal 2 starting at the beginning of the For statement, making the reference (1, 2), which will refer to the first row and second column. The entire process will carry on until i is equal to the parameter set in the For statement, which is three (the total range count). This is the crux of how looping works in ranges. Finally, add a button labeled **Calculate Analytics** near D7 on the Inputs sheet and assign the macro.

Model Builder 10.3 Final Code:

```
Sub SolveYield()
Application.StatusBar = "Solving Analytics . . . "
Application.ScreenUpdating = False
   Const YieldChange As String = "rngYieldChange"
   Const Target As String = "rngYieldTarget"
   Dim YieldRange As Range
   Dim TargetRange As Range
   Dim i As Integer
   i = 1
   Worksheets("Analytics").Select

For i = 1 To Range(Target).Cells.Count
   Set TargetRange = Range(Target).Cells(1, i)
   Set YieldRange = Range(YieldChange).Cells(1, i)
   TargetRange.GoalSeek Goal: = 0, ChangingCell: = YieldRange
Next i

Application.ScreenUpdating = True
Application.StatusBar = False
Calculate
Application.Calculation = xlCalculationAutomatic

End Sub
```

AUTOMATED SCENARIO GENERATION

A model operator often needs to run multiple scenarios by changing a number of variables for each run. The manual process would be to change each variable on the Inputs sheet by hand and then save or print out the Output Report for each run. This process can become very inconvenient and repetitive. A VBA solution is ideal for such a problem.

Creating a modelwide scenario generator demonstrates intermediate methods and techniques in VBA that can be transferred to other projects. Reading arrays, looping through arrays, and writing out the outputs are practices that allow a model builder tremendous flexibility. Also, additional concepts such as automating the creation and naming of new workbooks saves a model operator from repetitive time-consuming tasks.

MODEL BUILDER 10.4: CREATING A TRANSACTION SCENARIO GENERATOR

1. The first steps take place on the Inputs sheet of the Excel workbook. In cell B35 on the Inputs sheet, type the label **Scenario Generator** and label cell B36

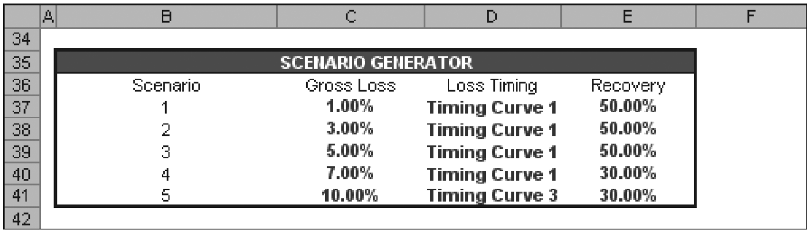
Scenario. Cells C36 through E36 are the labels for assumptions that are to be varied. This is up to the user, but in this example enter the following:
C36: Gross Loss
D36: Loss Timing
E36: Recovery
For now enter scenario number labels starting from 1 for B37:B42. The assumptions to vary can be copied from the Model Builder 10.4 file on the CD-ROM or made up at the user's discretion. It is recommended to copy the data from the CD-ROM so that the screenshots and examples can be tied to the user's model under construction. Also, a few ranges need to be named:
C37:C41: rng_ScenGen1
D37:D41: rng_ScenGen2
E37:E41: rng_ScenGen3
The new area should look like Figure 10.8.

2. Next launch the VBE and create a new module named **Scenarios**. In that module, start a new subroutine called **Scenario_Generator**().
3. The first part of the code is declaring constants and variables. Enter the following below the subroutine name:

```
Const MaxScens = 5
Dim i As Integer, k As Integer
Dim Scen1Array(1 To MaxScens), Scen2Array(1 To MaxScens), Scen3Array (1 To MaxScens)
Dim Scen1Option As String, Scen2Option As String, Scen3Option As String
```

In this exercise, a constant is set up for the maximum number of scenarios, counting variables i and k are created, and array and string variables are declared. Arrays are a new concept for this section. An array is a range of data that can be multidimensional (containing both columns, rows, or multiple combinations). Arrays can be visualized as a range of cells on the worksheet. In this example the arrays will be one-dimensional with a single column of data.

It is important to understand that an array can be visualized as a range of cells in VBA, but it does not entirely act like one. To get data into an array in VBA it must be read in through looping; copying and pasting will not work.



SCENARIO GENERATOR

Scenario	Gross Loss	Loss Timing	Recovery
1	**1.00%**	**Timing Curve 1**	**50.00%**
2	**3.00%**	**Timing Curve 1**	**50.00%**
3	**5.00%**	**Timing Curve 1**	**50.00%**
4	**7.00%**	**Timing Curve 1**	**30.00%**
5	**10.00%**	**Timing Curve 3**	**30.00%**

FIGURE 10.8 The Scenario Generator is controlled on the Inputs sheet, but run using VBA.

Similar loops must be created to get parts of an array out of VBA and onto an Excel sheet, unless the entire array is to be written to the sheet.

4. As in the previous macros, assign string values to certain variables so the code does not have to be changed too drastically if the sheet names change. In this case there are three sheet references that will be varied per scenario. Insert the follow code after the variable declarations (also notice that screen updating is turned off here):

   ```
   Scen1Option = "pdrCumLoss1"
   Scen2Option = "pdrLossTime1"
   Scen3Option = "pdrRecovRate1"
   Application.ScreenUpdating = False
   ```

5. The first loop that needs to be set up reads in the possible values for each scenario and stores those values in VBA arrays. This is done by creating a loop for each possible scenario (5 in this case) and storing a value in a specific, ordinal section of an array. Below the previous code enter the following:

   ```
   For i = 1 To MaxScens
     Scen1Array(i) = Range("rng_ScenGen1").Cells(i, 1)
     Scen2Array(i) = Range("rng_ScenGen2").Cells(i, 1)
     Scen3Array(i) = Range("rng_ScenGen3").Cells(i, 1)
   Next i
   ```

   In this code, the array Scen1Array(i) will store the value in the first cell of the sheet range rng_ScenGen1. The macro understands to do this because the first loop changes the variable i to a value of 1, which means that Scen1Array(1) will be equal to the 1 row, 1 column in rng_ScenGen1 (through the use of the Cells method). Each array will be filled until the maximum number of scenarios is reached (5 in the example model).

6. This section of code is another loop. This time each iteration of the loop will complete all of the tasks to calculate and export the scenario. The first part of this code is transferring the values from the arrays to the sheet for each iteration. Enter the following after the previous code:

   ```
   For k = 1 To MaxScens
     Range(Scen1Option) = Scen1Array(k)
     Range(Scen2Option) = Scen2Array(k)
     Range(Scen3Option) = Scen3Array(k)
     Calculate
   ```

   For each k loop, the respective array value is written back to the Excel sheet in the range determined by the string variables. The command Calculate is included to make sure that all of the formulas in the Excel sheet calculate since changes will have occurred due to changing the three ranges.

7. After the workbook is calculated, the user may want to optimize the advance rate. If this is the case insert the following code:

```
Call SolveAdvance
```

The Call command runs the subroutine that follows it.

8. After each scenario is calculated and the debt optimized, the results should be recorded and stored. The Output sheet contains all of the results that are wanted and is in a constant form. This makes it easy to copy and paste. First the Output sheet needs to be copied at the end of each k loop iteration:

```
Sheets("Output").Select
Range("A1:P38").Copy
```

This code selects the Output sheet and copies the data ranges. Note that the graphs are not copied since they would retain the original links and not have scenario specific results.

9. The copied data should be stored in a separate worksheet. To do this, a new worksheet needs to be inserted, the data pasted, and formatted correctly. Below the previous code enter the following:

```
Sheets.Add
Range("A1").Select
Selection.PasteSpecial Paste: = xlValues
Selection.PasteSpecial Paste: = xlFormats
```

This code adds a sheet to the workbook, selects cell A1, pastes the values of the copied range, and then pastes the formats of the copied range.

10. To save time the sheet should be automatically assigned a name and placed in a consistent place in the workbook. To accomplish this enter the following below the previous code:

```
ActiveSheet.Name = Format("Scen Output" & k)
ActiveSheet.Move After: = Sheets(6 + k)
```

Both the Name method and the Move method affect the ActiveSheet. The Name method changes the worksheet name to begin with Scen Output, followed by whatever number scenario the loop is iterating through. The Move method moves the sheet after a designated number of sheets. In this case the first k loop will move the sheet at the end of the seventh workbook sheet (6 + k on the first loop is seven).

11. Change the Status Bar so that the user knows the progression of the macro based on the loop iteration. Since this is the last line of code for the iteration, end the For Loop with a Next command. Enter the following below the previous code:

```
Application.StatusBar = "Running Scenario: " & Str(k) & " of " & Str(MaxScens)
Next k
```

12. A few lines of code are needed to clean up the macro. Below the last code enter the following:

```
Sheets("Inputs").Select
```

```
Application.ScreenUpdating = True
Application.StatusBar = False
End Sub
```

Instead of the macro ending on a scenario output and the user having to select the Input sheet again, the first line of the previous code automatically selects the Input sheet. Screen updating is turned back on and the Status Bar is reset. Finally, create a button on the Inputs sheet near cell G7 and assign this macro to run when it is pushed. The button should be labeled appropriately, such as **Generate Scenarios**.

13. An additional subroutine that saves time is one that deletes old scenario sheets. This can be added at the user's discretion; but it can be incredibly useful if deleting old scenarios is becoming a repetitive task. It also shows model builders how to identify specific sheets in the workbook automatically and perform operations on them. First, create a new subroutine and declare a worksheet variable:

```
Sub DeleteSheets()
Dim VBAwksht As Worksheet
```

14. Next, a loop should be inserted to delete the old scenarios that are stored in the workbook. This loop will loop through each sheet, check to see if the sheet is a scenario output (remember they all start with the same naming convention of "Scen Output"), and delete the sheet if it is. Enter the following code below the code that turns off screen updating:

```
For Each VBAwksht In ActiveWorkbook.Worksheets
   If Left(VBAwksht.Name, 11) = "Scen Output" Then
      VBAwksht.Delete

End If
Next
```

The For-Next loop is slightly different since it essentially reads "For each object of this type in this object". The If statement uses the Left command, which looks at each worksheet's name and checks to see if the first 11 characters starting from the left read Scen Output. If the statement is true then the worksheet is deleted, otherwise it is skipped and the loop continues until there are no more worksheets in the workbook. When this macro is actually run there will be a prompt to delete each individual sheet as a protection against deleting important data. This macro is contained in the scenarios module in file *MB10-4.xls* in the Ch10 folder on the CD-ROM, but no button has been created in the model.

Model Builder 10.4 Final Code:

```
Sub Scenario_Generator()
Const MaxScens = 5
Dim i As Integer, k As Integer
Dim Scen1Array(1 To MaxScens), Scen2Array(1 To MaxScens), Scen3Array
(1 To MaxScens)
```

```
Dim Scen1Option As String, Scen2Option As String, Scen3Option As String
Dim VBAwksht As Worksheet
Scen1Option = "pdrCumLoss1"
Scen2Option = "pdrLossTime1"
Scen3Option = "pdrRecovRate1"
Application.ScreenUpdating = False
For Each VBAwksht In ActiveWorkbook.Worksheets
  If Left(VBAwksht.Name, 11) = "Scen Output" Then
    VBAwksht.Delete
  End If
Next
For i = 1 To MaxScens
  Scen1Array(i) = Range("rng_ScenGen1").Cells(i, 1)
  Scen2Array(i) = Range("rng_ScenGen2").Cells(i, 1)
  Scen3Array(i) = Range("rng_ScenGen3").Cells(i, 1)
Next i
For k = 1 To MaxScens
  Range(Scen1Option) = Scen1Array(k)
  Range(Scen2Option) = Scen2Array(k)
  Range(Scen3Option) = Scen3Array(k)
  Calculate

Call SolveAdvance

  Sheets("Output").Select
  Range("A1:P38").Copy
  Sheets.Add
  Range("A1").Select
  Selection.PasteSpecial Paste: = xlValues
  Selection.PasteSpecial Paste: = xlFormats
  ActiveSheet.Name = Format("Scen Output" & k)
  ActiveSheet.Move After: = Sheets(6 + k)
  Application.StatusBar = "Running Scenario: " & Str(k) & " of " &
Str(MaxScens)

Next k
Sheets("Inputs").Select
Application.ScreenUpdating = True
Application.StatusBar = False
End Sub
```

Optional Macro

```
Sub DeleteSheets()
Dim VBAwksht As Worksheet
  For Each VBAwksht In ActiveWorkbook.Worksheets
```

```
        If Left(VBAwksht.Name, 11) = "Scen Output" Then
            VBAwksht.Delete
        End If
    Next
End Sub
```

WORKING WITH MACROS IN EXCEL

Now that a number of example macros have been created, it should be stated outright that macros can do quirky things. Personal experience has witnessed a handful of times that an error was generated when everything seemed completely correct. Code was rechecked for the slightest misspelling or typo. Macros were debugged using the watch window and break commands. Hours spent poring over each line of code only to realize that after saving the model, shutting it down, and reopening it, the macro worked perfectly!

Glitches like this are rare; but they can occur. Much of the time an error is due to a misspelled variable or reference. Other typical errors include a data mismatch where the variable was declared as a certain type, but the code is directing a different type of value to be passed through, a value exceeds the constraints of a dimensioned array, or a looping variable has exceeded the parameters of the loop. Learning how to use code breaks and the watch window are invaluable for debugging these and all other errors. Even though the errors can be frustrating and the learning curve high, the benefit of understanding VBA is well worth it.

CHAPTER 11

Conclusion

The best financial modelers understand both technical methodologies and business concepts. One without the other produces an Excel expert or a standard financial analyst. An experienced financial modeler can deconstruct a business concept and transfer the idea into an application or programming code that runs accurately, efficiently, and transparently. The same financial modeler understands how changing assumptions impact the transaction and the implications such adjustments have on the performance of the deal.

So far this text has attempted to achieve both technical and conceptual understanding for structured finance. A model has been constructed in a step-by-step technical manner with business-related theories explained along the way. A section has been dedicated to understanding the model's mechanics, outputs, and the ensuing business interpretations. The only part that lacks discussion is the higher level view—about how professionals in different industries look at the model differently and garner information relevant to their position.

THE INVESTMENT BANKER'S PERSPECTIVE

As some reader's may have noticed, this book is written with a slight bias towards investment banks. Often bankers are the people who construct a transaction model, which is used by most parties involved in a deal. For this reason, a strong focus is put on the bank's perceived risks and protecting against them. Even among banks this risk differs depending on who ends up owning the assets and how the debt is funded.

If the bank is retaining the risk by funding the deal on its balance sheet or through a conduit, then the modeling of the retained debt will be the focus of the bank's analysis. For most sensitivity scenarios, the bank is primarily concerned with a risk rating derived from expected loss and the expectation of loss in general.

However, if the bank is selling the debt into the capital markets, then a greater focus of the modeling will involve debt yield, duration, weighted average life, decrement calculations, and other metrics that investors look for when purchasing securities. A bank selling a transaction into the capital markets is also concerned with loss because no bank wants to have its name associated with a failed transaction.

Overall, since bankers deal with all parties in a transaction, they are concerned with every part of the model. The issuer may change the asset composition or criteria, which would need to be immediately updated in the model to generate accurate cash flows. Investors might demand certain protections that change the waterfall. A surety might be brought in to wrap the transaction, which affects the liability structure. Or a rating agency might ask for certain scenarios that stress the asset and liability assumptions. All of these situations can be handled with the model framework laid out in this book.

THE INVESTOR'S PERSPECTIVE

For the most part, the investor and the bank have aligned interests. Structured transactions are even designed in such a way to align those interests. An investor is concerned about loss to exposure, which can be any level of debt from senior to subordinated. Investors also have timing in mind because many are trying to purchase assets that fit a specific profile that depends on yield, risk rating, and duration.

With such concerns, the investor will want to verify loss and prepayment expectations, interest rate environments, and principal allocation structures. If a deal begins to melt down, the investor also needs to know how much the deal can expect to make or lose and how long it will take to get to such a result. The investor primarily examines the liability side of a model to gather information relevant to their decision.

THE ISSUER'S PERSPECTIVE

Similar to bankers, issuers have a very complex role in structured transactions. Issuers are more familiar with the asset side because it is an integral part of their business. In the early stage of a transaction, the issuer spends a large amount of time examining which assets to include in the transaction pool. Building the pool requires constant analysis of the pool characteristics as loans are added or taken away. Ultimately, the issuer wants to create a pool that will sustain the transaction over time.

At the same time, the issuer wants to get the best funding rate possible. This means that certain risk ratings are desired, which can be achieved by varying amounts of credit enhancement. Beside the asset pool, issuers need to completely understand the liability waterfall, forms of credit enhancement, and the affects of varying assumptions. Only then can they know if they are getting the best possible arrangement.

THE FINANCIAL GUARANTOR'S PERSPECTIVE

A market participant that is also concerned with structured finance modeling is a financial guarantor. This could be a monoline insurance company—or a government entity offering a debt wrap or credit guarantee. Since both of these entities offer pledges to pay interest and principal, they are both highly concerned about cash flow and stress scenarios.

A financial guarantor would want to run extreme cases to see if the transaction withstands a certain risk threshold. It would want to know which scenarios cause loss and the probability of such scenarios. This requires a thorough understanding of the assets and their expected performance. Also, importance would be placed on the guarantor's place in the waterfall in respect to what is wrapped and where reimbursements are allocated.

THE BIG PICTURE PERSPECTIVE

No matter what role or interest a person has in a structured finance model, he or she should never lose sight of what is trying to be analyzed. What might have worked for one transaction may no longer apply for a new transaction, even though they may at first appear similar. Every time a transaction needs modeling the person responsible should determine what needs to be measured and the best method of getting to the correct result. This could be as simple as copying an old model and changing a few assumptions or as complex as building a new one entirely from scratch.

Experienced modelers slowly build a set of models from which they can take pieces to quickly construct new and unique models. The basic model created in Project Model Builder, for example, can easily be adapted to a project finance model by changing how the cash flows are generated. Imagine that a toll road was being financed through senior and subordinate debt. The only difference then would be to change the asset side of the model so cash flows are produced off of traffic estimates and specialized consultant data.

Also, the example model created in this text is somewhat rudimentary. More powerful models have the ability to generate cash from multiple representative lines or unlimited loans. Asset-specific models go into details that are pertinent to industries such as mortgages, autos, leases, and so on. All of these advanced additions require industry-specific knowledge and analytical standards knowledge.

As modeling experience grows, the challenge is to prevent tunnel vision based on past successes. New asset and esoteric classes that are developing in the structured finance industry require modelers to be open to new techniques and different approaches. The measure of this book's success is its ability to teach a person a technical skill while simultaneously developing business understanding. The ultimate objective is to ascertain those skills and take the next step to develop new and more powerful models.

APPENDIX

Using This Book with Excel 2007

The upcoming release of Microsoft Excel 2007 has a number of improvements that make financial models more powerful and capable of handling larger amounts of data. The most obvious changes are graphical, namely a noticeable update to the toolbar command system, which allows faster access to controls for formatting and manipulating spreadsheet data. There is also a vast expansion of the maximum number of rows and columns. Figure A.1 shows the new look of Excel 2007's default workbook window. A very important new feature is Excel's improved use of memory. Formula calculation methodologies have also been made more efficient. Both speed up total workbook calculation time. While it may take a few years before companies switch over to Excel 2007, small businesses and tech-savvy individuals will want to run Excel 2007 now—and they will need to know how it affects this book.

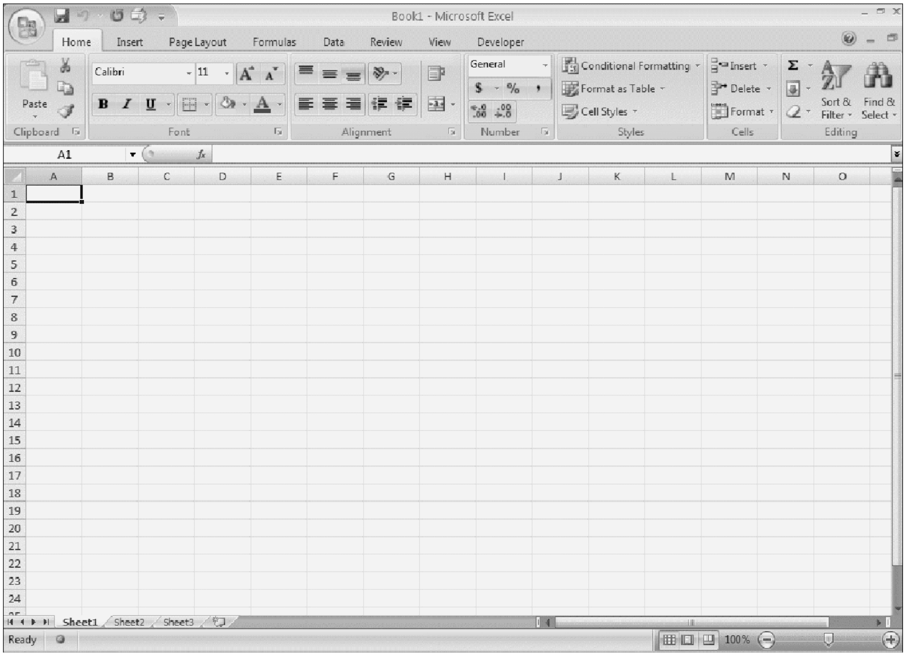


FIGURE A.1 Microsoft Excel 2007's default workbook window.

Overall, the difference between Excel 2007 and earlier versions will have no impact on the functionality of the Project Model. However, a few of the step-by-step instructions will need to be performed differently due to the new tabbed Ribbon system that Excel 2007 uses. Instead of the menu bar selections commonly found on Microsoft Office 2000, 2002, and 2003 applications—File, Tools, Format, and the like—Excel 2007 uses a Ribbon and various tabs that reorganize and access commonly used controls. Accessing those operations is more graphical and icon-based rather than menu driven. This affects the creation of data validation lists, pasting formats, inserting names, adding worksheets, using Goal Seek, and opening the Visual Basic Editor (VBE). Instead of using the menu commands to access these operations, the Ribbon system is used by clicking on the appropriate button. (The keyboard shortcuts still remain and can be used to quickly use many of those operations.) Using operations and tools relevant to this book in Excel 2007 have been detailed in this appendix.

The other obvious change is the virtual elimination of row and column constraints, which changes one important concept in this book. Early in the book, it is stated that for Excel 2003 and prior versions there are row and column constraints that dictate the vertical or horizontal time progression of the model. Given that Excel 2007 offers 16,000 columns and over 1 million rows per sheet, the decision to have time progress vertically or horizontally is irrelevant. However, most structured models are formatted with period progression going vertically; to maintain convention and the ease of use for other structured modelers—this method should be used. There are other areas in financial modeling where the elimination of the row and column constraints are particularly useful, such as working with loan level data tapes. But keep in mind that you are still constrained by the computer's memory and Excel's allocation of that memory. That said, Excel 2007 has relatively little impact on this book because the real value lies in the theory, and the formula and functions that translate that theory into a model. Aside from a handful of functions that have undergone some extremely minor alterations, all of the functions are the same. This means that using the functions in the formulas as the book instructs will produce a consistent result regardless of the Excel version being used.

GUIDE TO OPERATIONS AND FUNCTIONS IN EXCEL 2007 RELEVANT TO THIS BOOK

Data Validation

Data validation is used throughout this book in order to create drop-down lists in cells. The list values are stored on the Hidden sheet, named as ranges, and are used on the inputs sheet through the data validation tool. To use this tool in Excel 2007:

1. Make the cell that requires the data validation list the active cell.
2. Click the **Data tab** and then click **Data Validation** in the Data Tools group. See Figure A.2.

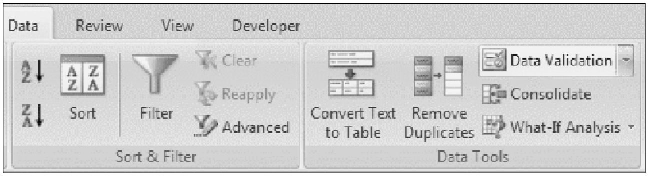


FIGURE A.2 Location of Data Validation on the Data tab.

3. On the Settings tab and under Validation criteria, click **List** in the Allow drop-down list.
4. Enter the named range of the list data in the Source box.
5. Click **OK**.

Cell Formatting

Most formatting in Excel 2007 is found on the Home tab shown in Figure A.3. From there the format can be set using options in the Font, Alignment, Number, and Styles boxes. In particular Excel 2007 has a number of new features that make cell formatting easier to use and more robust.

Still, one of the most useful features is the Format Painter, now located in Excel 2007 in the Clipboard group on the Home tab. This feature works by clicking on a cell with a desired format, clicking the Format Painter button, and then clicking on a different cell where the format should be applied. For cells that are not contiguous, double-clicking the Format Painter keeps it selected.

There are preset cell styles that makes customizing a worksheet faster. These can be found in the Styles group on the Home tab by clicking the Cell Styles button. Custom styles can be selected on the Cell Styles palette—and be created here and saved on this palette for quick access.

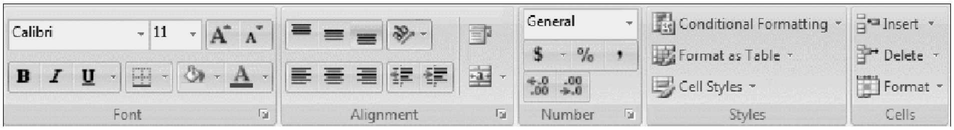


FIGURE A.3 Common formatting tools are found on the Home tab.

Conditional Formatting

In Chapter 8, conditional formatting is used for cells with logic tests to make it clear to the user that the model is either working properly or not. Conditional formatting is located in the Styles group on the Home tab box shown in Figure A.4.

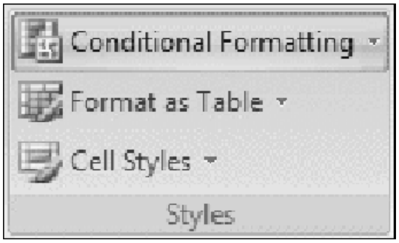


FIGURE A.4 Click Conditional Formatting button to access option menus.

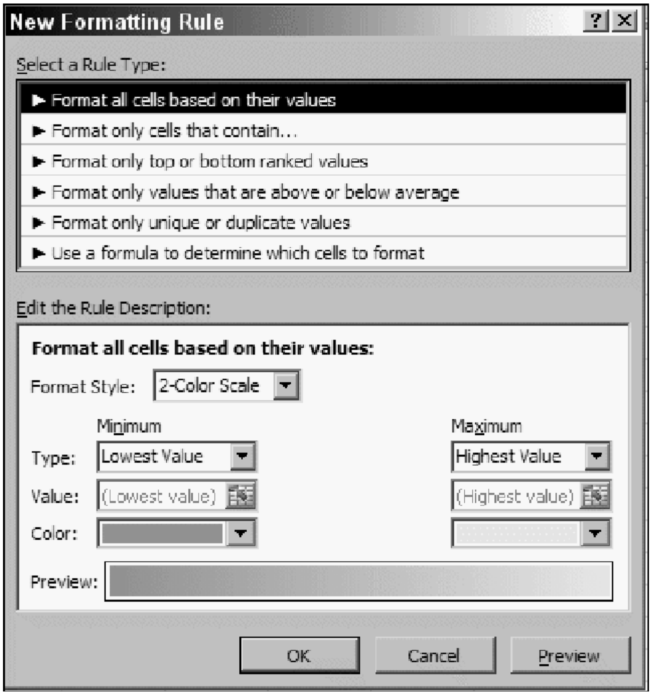
Excel 2007 has greatly expanded the conditional formatting options, which is evident when the Conditional Formatting button is clicked. Instead of opening a dialog box, as was the case in Excel 2003 and earlier, a menu opens that allows you to select the desired conditional formatting from its corresponding submenu.

To create the conditional formatting used in this model:

1. Click the **Conditional Formatting** button in the Styles group on the Home tab.
2. Click **Highlight Cells Rules** to open its menu.
   The logic tests in Chapter 8 worked off of either TRUE or FALSE return values. This requires making the cell formatting conditional on whether its value is equal to either TRUE or FALSE. The correct option from Highlight Cells Rules menu is Equal To. . .
3. Click **Equal To** . . . The Equal to dialog box appears. In the Format cells that are EQUAL TO: box enter **TRUE**.
4. In the drop-down list to the right, select the desired conditional formatting formatting that should be applied when the rule is valid. Click **Green Fill with Dark Green Text.** This is an excellent option if everything is working properly. Otherwise a custom format can be selected as the last option in the drop-down list.
5. Repeat steps 1 through 4 to set the conditional formatting for the FALSE value. Select **Red Fit with Dark Red Text** as the cell formatting to indicate errors.

Clicking the Conditional Formatting button also gives you access to such conditional formatting options as data bars, color scales, and icon sets. These are particularly useful features in financial modeling to quickly see the differences in values or trends over time.

To create your own custom conditional formats, click the Conditional Formatting button and select New Rule from the menu. The New Formatting Rule dialog box opens (shown in Figure A.5). This allows you to select your own options for creating custom conditional formatting rules.



New Formatting Rule

Select a Rule Type:

- ► Format all cells based on their values
- ► Format only cells that contain...
- ► Format only top or bottom ranked values
- ► Format only values that are above or below average
- ► Format only unique or duplicate values
- ► Use a formula to determine which cells to format

Edit the Rule Description:

Format all cells based on their values:

Format Style: 2-Color Scale

	Minimum	Maximum
Type:	Lowest Value	Highest Value
Value:	(Lowest value)	(Highest value)
Color:		

Preview:

FIGURE A.5 The New Formatting Rule dialog box allows you to create custom conditional cell formats.

Naming Ranges

Ranges can still be named quickly in Excel 2007 by clicking Name Box list in the top left corner of the worksheet.

Excel 2007 now has its powerful naming features on the Formulas tab. Here you will find the Defined Names group, which contains such controls as the Define Name and Name Manager buttons (see Figure A.6). Ranges are named using the Define Name button. Clicking it opens the New Name dialog. Here the range is named, its parameters defined for either the entire workbook or a specific sheet, comments can be entered, and the reference selected. For preexisting named ranges, clicking the Name Manager allows you to access the desired range to make edits. The Name Manager dialog box also shows the value of the named cell, what the cell's reference is, whether it is set to the entire workbook or a specific sheet, and any comments. New names can also be created from this dialog box.

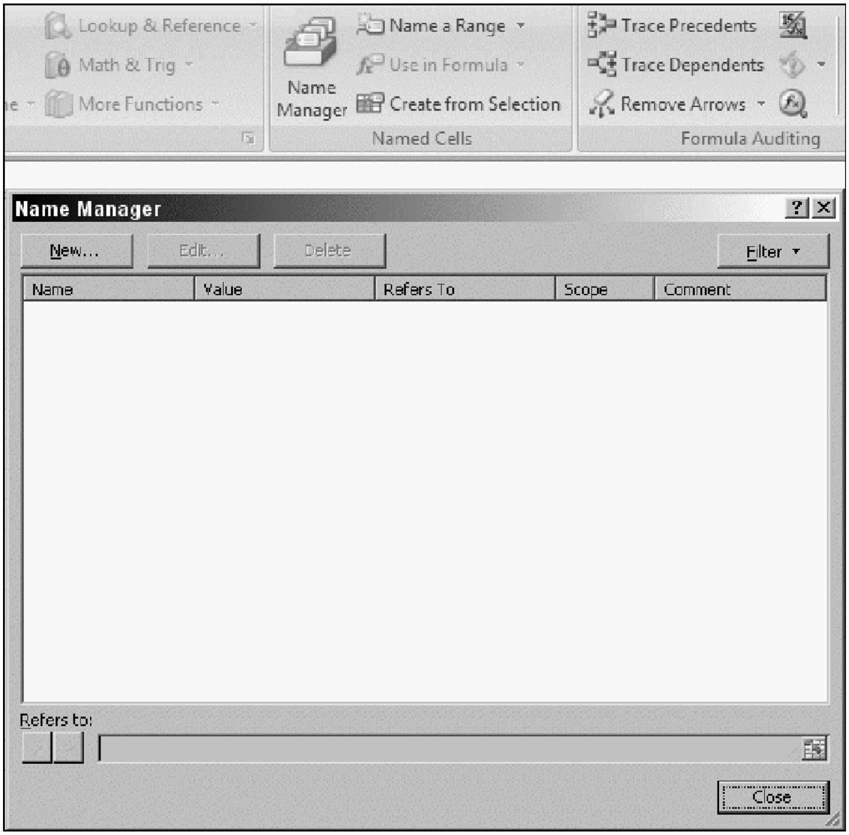


FIGURE A.6 The Defined Names group controls on the Formula tab.

Adding Worksheets

Excel 2007 has a very convenient shortcut for adding new worksheets. At the bottom of the workbook, where the sheet tabs are located, click the Insert Worksheet button tab, indicated by a worksheet icon with an asterisk shown in Figure A.7. If you don't see the Insert Worksheet button, which can happen in multisheet workbooks, click or scroll to the rightmost worksheet tab.

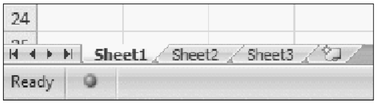


FIGURE A.7 Click the Insert Worksheet button to create a new worksheet.

Goal Seek Tool

To use the Goal Seek function in Excel 2007, click the Data tab and then click the What-If Analysis button in the Data Tools group. Select Goal Seek from the drop-down list. This opens the Goal Seek dialog box as in previous versions of Excel (see Figure A.8).

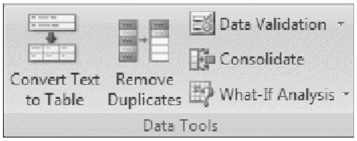


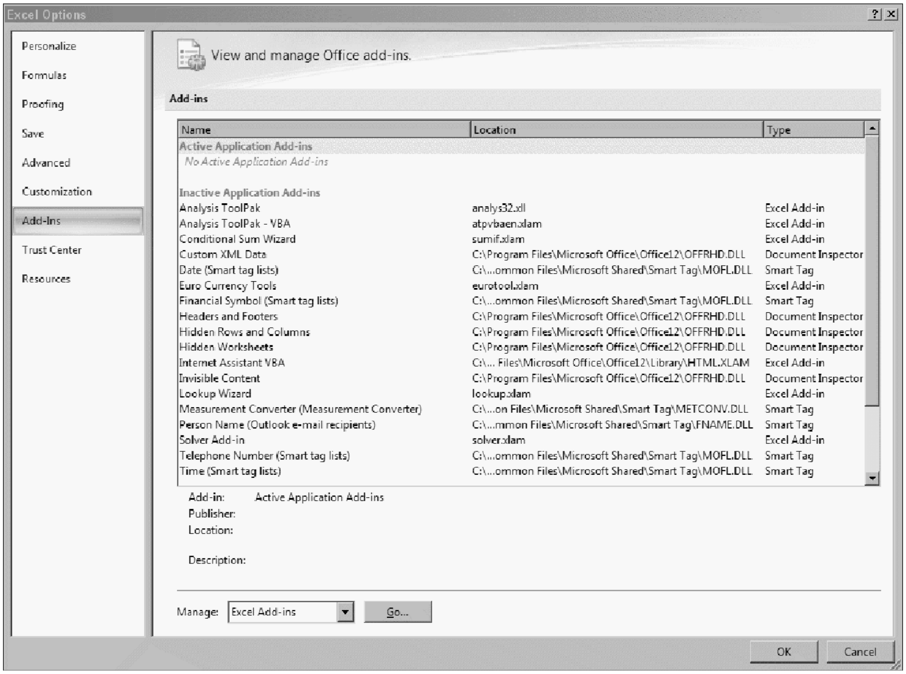
FIGURE A.8 Click the What-If Analysis button in the Data Tools group to select Goal Seek.

Add-Ins

The Analysis ToolPak and the Analysis ToolPak—VBA are Excel add-ins that provide Excel with more functionality such as the EDATE function used in this book. To access Excel add-ins and activate either or both Analysis ToolPaks, use the following steps:

1. Click the **Microsoft Office** button in the top-left corner of the Excel window.
2. On the bottom of the menu, click **Excel Options.** The Excel Options dialog box opens (see Figure A.9).
3. Click **Add-Ins** in the options category list.

   The Add-ins pane opens. Note that it is divided into two lists: Active Application Add-ins and Inactive Application Add-ins.
4. Select the desired add-in from the Inactive Application Add-ins list. In this case, you would select either **Analysis ToolPak** or **Analysis ToolPak—VBA**.
5. Click the **Go** button.
6. In the Add-Ins dialog box, select the desired options. If you selected Analysis ToolPak, you can also include the VBA option.
7. Click **OK**.



View and manage Office add-ins.

Add-ins

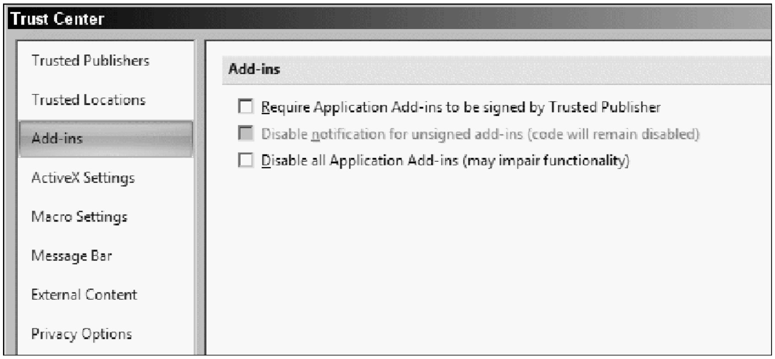
Name	Location	Type
Active Application Add-ins		
No Active Application Add-ins		
Inactive Application Add-ins		
Analysis ToolPak	analys32.xll	Excel Add-in
Analysis ToolPak - VBA	atpvbaen.xlam	Excel Add-in
Conditional Sum Wizard	sumif.xlam	Excel Add-in
Custom XML Data	C:\Program Files\Microsoft Office\Office12\OFFRHD.DLL	Document Inspector
Date (Smart tag lists)	C:\...ommon Files\Microsoft Shared\Smart Tag\MOFL.DLL	Smart Tag
Euro Currency Tools	eurotool.xlam	Excel Add-in
Financial Symbol (Smart tag lists)	C:\...ommon Files\Microsoft Shared\Smart Tag\MOFL.DLL	Smart Tag
Headers and Footers	C:\Program Files\Microsoft Office\Office12\OFFRHD.DLL	Document Inspector
Hidden Rows and Columns	C:\Program Files\Microsoft Office\Office12\OFFRHD.DLL	Document Inspector
Hidden Worksheets	C:\Program Files\Microsoft Office\Office12\OFFRHD.DLL	Document Inspector
Internet Assistant VBA	C:\... Files\Microsoft Office\Office12\Library\HTML.XLAM	Excel Add-in
Invisible Content	C:\Program Files\Microsoft Office\Office12\OFFRHD.DLL	Document Inspector
Lookup Wizard	lookup.xlam	Excel Add-in
Measurement Converter (Measurement Converter)	C:\...on Files\Microsoft Shared\Smart Tag\METCONV.DLL	Smart Tag
Person Name (Outlook e-mail recipients)	C:\...mmon Files\Microsoft Shared\Smart Tag\FNAME.DLL	Smart Tag
Solver Add-in	solver.xlam	Excel Add-in
Telephone Number (Smart tag lists)	C:\...ommon Files\Microsoft Shared\Smart Tag\MOFL.DLL	Smart Tag
Time (Smart tag lists)	C:\...ommon Files\Microsoft Shared\Smart Tag\MOFL.DLL	Smart Tag

Add-in: Active Application Add-ins
Publisher:
Location:

Description:

Manage: Excel Add-ins ▾ Go...

FIGURE A.9 Excel Options dialog box allows you access to Excel Add-Ins and other Excel application options.



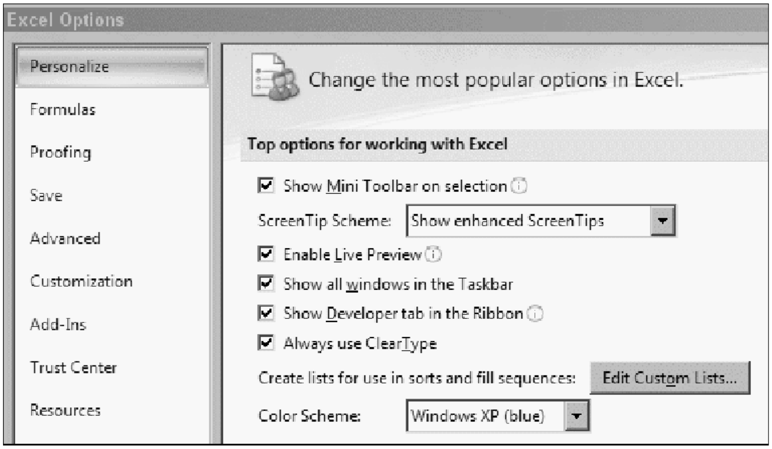
Add-ins

□ Require Application Add-ins to be signed by Trusted Publisher
□ Disable notification for unsigned add-ins (code will remain disabled)
□ Disable all Application Add-ins (may impair functionality)

FIGURE A.10 Add-ins dialog box options for Analysis ToolPak.

Visual Basic Editor

The Visual Basic Editor is still accessed by pressing Alt + F11.To have macro and Visual Basic options available as a tab, you must activate the Developer tab in Excel Options.



☑ Show Mini Toolbar on selection

ScreenTip Scheme: Show enhanced ScreenTips

☑ Enable Live Preview

☑ Show all windows in the Taskbar

☑ Show Developer tab in the Ribbon

☑ Always use ClearType

Create lists for use in sorts and fill sequences:

Color Scheme: Windows XP (blue)

FIGURE A.11 Click in the Show Developer tab in the Ribbon checkbox select this option and access Visual Basic options in the Excel 2007 window.

1. Click the **Microsoft Office** button in the top-left corner of the Excel window.
2. On the bottom of the menu, click **Excel Options.** The Excel Options dialog box opens
3. Click **Popular** in the options category list.
4. In the Popular pane, select the **Show Developer tab in the Ribbon** checkbox (see Figure A.11).
5. Click **OK.**

These steps will create the Developer tab in the ribbon and allow for easy access to code, VBA control, and XML options.

Please note that this appendix was created using a Beta version of Excel 2007. Command processes and graphics may have changed in the final release version.

About the CD-ROM

INTRODUCTION

This appendix provides you with information on the contents of the CD that accompanies this book. For the latest and greatest information, please refer to the ReadMe file located at the root of the CD.

SYSTEM REQUIREMENTS

For best results, use Microsoft Excel 2000 or better that use the appropriate system requirements for your particular version and type. The following are the minimum system requirements for Microsoft Office 2003 and Microsoft Office 2007.

MICROSOFT OFFICE 2003	
Computer and processor	233-MHz or faster processor
Memory	128 MB RAM
Hard disk	400 MB of available hard-disk space; optional installation files cache (recommended); requires an additional 200 MB of available hard-disk space.
Drive	CD-ROM or DVD drive
Display	Super VGA (800 × 600) or higher-resolution monitor
Operating system	Microsoft Windows® 2000 with Service Pack 3 (SP3), Windows XP, or later

MICROSOFT OFFICE 2007	
Computer and processor	500 megahertz (MHz) processor or higher
Memory	256 megabyte (MB) RAM or higher
Hard disk	1.5 gigabyte (GB); a portion of this disk space will be freed after installation if the original download package is removed from the hard drive.
Drive	CD-ROM or DVD drive
Display	1024 × 768 or higher resolution monitor
Operating system	Microsoft Windows® XP with Service Pack (SP) 2, Windows Server® 2003 with SP1, or later operating system

USING THE CD WITH WINDOWS

To install the items from the CD to your hard drive, follow these steps:

1. Insert the CD into your computer's CD-ROM drive.
2. The CD-ROM interface will appear. The interface provides a simple point-and-click way to explore the contents of the CD.

If the opening screen of the CD-ROM does not appear automatically, follow these steps to access the CD:

1. Click the Start button on the left end of the taskbar and then choose Run from the menu that pops up.
2. In the dialog box that appears, type ***d***:\start.exe. (If your CD-ROM drive is not drive d, fill in the appropriate letter in place of ***d***.) This brings up the CD Interface described in the preceding set of steps.

WHAT'S ON THE CD-ROM

The following sections provide a summary of the software and other materials you'll find on the CD.

Model Builder Files

Each chapter in the text includes Model Builder sections that instruct the readers on how to build the model step by step. Included on the CD-ROM are Excel workbooks organized by chapter named folders that contain the completed Model Builder exercise. The Model Builder exercises are named and organized according to the corresponding chapter/subsection. For example, in the text the Model Builder exercises use a "Model Builder: Chapter.Subsection" convention, while the CDROM files are entitled MB "Chapter.Subsection."

Additional Files

Within each Model Builder chapter folder in the CD-ROM there may be a folder entitled, "Additional Files." These files are supplementary analysis/examples that are relevant to the chapter.

CUSTOMER CARE

If you have trouble with the CD-ROM, please call the Wiley Product Technical Support phone number at (800) 762–2974. Outside the United States, call (317) 572–3994. You can also contact Wiley Product Technical Support at http://support.wiley.com. John Wiley & Sons will provide technical support only for installation and other general quality control items. For technical support on the applications themselves, consult the program's vendor or author.

To place additional orders or to request information about other Wiley products, please call (877) 762–2974.

Index

For more information regarding the CD, see the About the CD-ROM section on page 189.

John Wiley & Sons, Inc.