

Diagonal Bricklayer Model Operational Description

Model Goal

The Diagonal Bricklayer Model (DBM) has one simple and well defined goal. It will deterministically calculate a structure of bond debt through early redemption in order to obtain a level revenue requirement for all Model Years of a study in accordance with user defined parameters and controls.

Algorithm Description

A single run of the algorithm consists of a number of Study Years (SY) which defines the debt structure and starting point for processing debt. Each Study year is further defined by a set of Model Years (MY) from the first Fiscal Year to the last Fiscal Year of the model.

The debt is split into four categories: Historical, Projected, Replacements and Forecasts. Historical bonds represent debt currently on the books and are present for all Study Years in the model. Projected bonds represent debt that BPA has under consideration for issue. Replacement bonds represent a form of projected debt that is issued for all remaining Model Years of each study. Forecast Bonds are another form of projected debt that exists for all Study Years of a model in the same way as Historical Bonds. The following diagram illustrates a 20 year run with debt as it comes into the model (Forecasts are not represented since they are like Historical bonds in this regard).

		Model Years																				
		2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028
Study Years	1	1																				
	2	1	2																			
	3	1	2	3																		
	4	1	2	3	4																	
	5	1	2	3	4	5																
	6	1	2	3	4	5	6															
	7	1	2	3	4	5	6	7														
	8	1	2	3	4	5	6	7	8													
	9	1	2	3	4	5	6	7	8	9												
	10	1	2	3	4	5	6	7	8	9	10											
	11	1	2	3	4	5	6	7	8	9	10	11										
	12	1	2	3	4	5	6	7	8	9	10	11	12									
	13	1	2	3	4	5	6	7	8	9	10	11	12	13								
	14	1	2	3	4	5	6	7	8	9	10	11	12	13	14							
	15	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15						
	16	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16					
	17	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17				
	18	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18			
	19	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19		
	20	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	

The numbers along the left side are the Study Years and the Year Numbers along the top are Model Years. The diagonal represented by the border between the two colors represents the region where the DBM processes debt and also where Replacement Bonds (Cyan) and Projections (Grey) come into the model. To be explicit, the Years are processed in the following way according to the chart above:

- SY 1, MYs 2008 – 2028
- SY 2, MYs 2009 – 2028
- SY 3, MYs 2010 – 2028
-
- SY 20, MYs 2027 – 2028

Thus, when we talk about processing along the diagonal, this is what we mean. Also, it should be evident by the diagram above that Replacements are completely new bond sets every Study Year, where Projections are an accumulation of bonds from all prior years to the current Study Year. This is represented in the table above by the numbers in each cell. Again, to avoid ambiguity, Projections and Replacements are processed by the model as follows:

- SY 1, MYs 2008 – 2028, Projections issued: 2009, Replacements Issued from 2010 - 2029
- SY 2, MYs 2009 – 2028, Projections issued: 2009 & 2010, Replacements Issued from 2011 - 2030
- SY 3, MYs 2010 – 2028, Projections issued: 2009 - 2011, Replacements Issued from 2012 - 2031
-
- SY 20, Model Years 2027 – 2028, Projections issued: 2009-2028, Replacements Issued: 2029 - 2049

As you can see, it is possible to have replacements which extend well beyond the end of the diagonal period and beyond model processing in general. This processing methodology represents a desired behavior for the model; that it does not re-process years for which it has already achieved a solution.

As each Study Year is completed, results are stored and processing begins the next study year with all decisions made up to the diagonal, but never beyond it. In addition, all bonds have a simple and restricted set of actions that can be performed on them in the DBM in order to accomplish the overall goal. They can be Called (redeemed early), or Rolled (refinanced). Any calls which happen to the right of the diagonal are cleared when the next study year starts. Those calls which happened to the left of the diagonal are retained for future results and analysis.

Bond debt is represented generally in the model by objects containing properties. The minimal set of properties for any debt is an initial principal, issue date, maturity date and interest rate. These characteristics are unchanging for all debt objects processed through the model.

In order to simplify the algorithm and provide an additional measure of control over the outcome, rolling has been defined as a strictly user controlled action in the DBM. Users can set specific roll dates on debt or allow the system to roll the debt to its maximum life or not to roll any bonds, but only prior to processing. Roll decisions can be applied to the debt directly (by altering the Final Maturity Date) or through the Global Value Adjustment Context statements. Regardless of choice, the decision is fixed for the life of the bond and will be the same for all Study Years in the model.

The model starts out by calculating the Total Debt Service for all Model Years in the first Study Year. The total debt service is defined as (MY represents the Model Year):

Total Debt Service (MY) = Sum (Bond P & I payments (MY)) + Sum (Bond Call Costs (MY)) + Sum (Expenses (MY)) + Floating Interest Income (MY)

Since Floating Interest Income is negative, it will tend to reduce Total Debt Service for the model year. Likewise, Expenses tend to be negative also and will reduce Total Debt Service at such times. All other values are determined by the bonds in question. Negative Bonds will be entirely negative for P & I, while all other bonds are positive. Bond call costs tend to be positive, increasing Total Debt Service. Make Whole Call discounts are handled by increasing the amount of call, so they do not directly reduce Total Debt Service in the years where they occur.

The initial revenue required to cover the Total Debt Service is zero, which places the model into a shortfall situation at the start. This provides the basis for the initial calculation of revenue requirement since no calls are possible when all years are in shortfall. Since the Revenue must be the same (level) for all Model Years of a study, only a single revenue number is calculated. The calculation for increasing revenue is:

Revenue += (1/3) (First Shortfall Amount / (Max (1, First Shortfall Year – Start Model Year)))

The revenue is adjusted upward by one third of the amount of shortfall divided by the number of years from the shortfall to the start of the model for the current study year. This has the effect of spreading the shortfall across the available years so that the increase in revenue can be distributed into smaller amounts. At the start of the model the First Shortfall Year and Start Model Year are the same and would equal zero. In this case, we assume 1 year and take the entire shortfall as the amount needed for adjustment. Then we increase revenue by one third of that amount until we have resolved the revenue to be within a set volatility number. If the volatility is 5%, then the revenue will end up within 5% of resolving the shortfall in the first year before the model will continue processing other years. The default volatility number is 0 (zero). If the first Model Year for the Study Year had been a surplus instead of a shortfall, revenue will be set to the Total Debt Service amount for that year and then begin the process of adjusting upward as needed. This action is necessary in order to achieve the lowest possible levelized revenue required to satisfy the total debt service for all Model Years in a study. The revenue can only be reduced when the first Model Year of a study has a surplus when compared against the final revenue

from the previous Study Year. Since the first Study year does not have a prior Revenue defined, this is not possible. However, it has been noticed that the end result would be the same either way.

The first year with a surplus, meaning that based upon the current revenue level we have excess money available, will be used in an attempt to redeem debt early in order to remove or reduce the surplus. Calls in the DBM are made on bonds in a specific ranking order. The preferred order sorts bonds using the following criteria:

Final Maturity Date (ascending), Interest Rate - Call Premium Rate (descending), Issue Date (ascending)

The outcome of this ranking is that bonds which are maturing soon with respect to the Year of Call (Action Year) will be called before bonds that are maturing later. This provides minimal movement for debt but also allows us to make room to call debt later which may be involved in causing a shortfall. If the Final Maturity Date is the same, we call the bonds with the highest interest rate first, taking into account any call premium cost, so that we save the most money in later years (by removing the interest payments). This tends to have a snowball effect on the surplus available in later years as the savings begin to build up. Visually, it would appear as a block or hole of surplus in the yearly Total Debt Service which moves toward the shortfall and reduces it if possible. An example for clarity:

Bond Properties

Principal: \$10,000,000

Issue: 2008

Maturity: 2028

Interest Rate: 5%

Total Interest Payments: \$500,000 a year for 20 Years = \$10,000,000

If the debt is left to mature, total cost will be \$20M. If the bond matures in a Shortfall Year, the impact from this bond is only \$10.5M. Calling the bond 1 year early will move the \$10.5 to the previous year and save \$500K in interest payments, assuming it is called at par. That creates reduction of almost \$11M in Total Debt Service during the Shortfall Year. If the bond we just created happened to be ranked highest at an earlier point in its life, then the amount of money saved in subsequent years will be greater and increase the ability of the model to call other debt earlier as well, saving even more money. When all surpluses in the earliest year are consumed, the next earliest year is used. This continues until there are no more surpluses between the starting Model Year and the first Shortfall Year at which point revenue must again be raised. The other possibility is that the shortfall is resolved and the model looks for another shortfall to deal with.

The process described above happens for one bond at a time. After each call on a single bond, the Total Debt Service is recalculated and the evaluation for Surplus and Shortfall begins again.

The process does not end for the Study Year until one of the following conditions is met:

- 1) All years are within a user specified Volatility of the revenue amount and within the Volatility for year to year variance between Total Debt Service amounts.
- 2) All available callable debt in every shortfall year is called.
- 3) All Model Years have been skipped because there are no callable bonds available.

After each study has finished processing, the results are saved to the database in order to reduce potential database locking. The model will continue processing additional study data while database saves occur.

User Controls in the DBM

The user has been given a tremendous amount of control over model behavior at various levels. While they cannot directly influence the algorithm behavior in terms of its overall processing methodology, they do have the ability to alter just about everything else. All controls in the model affect some portion of the model data, directly or indirectly. Each of these controls is listed in the order of most impact on the model to least impact. The lowest impact controls perform a form of fine tuning for specific outcomes.

The first and most coarse control in the model is one of Time Intervals. Time Intervals allow the user to specify the start and end year for a study period and the number of Study Years to jump between subsequent Study Years processed. Going back to the diagonal diagram for a moment:

		Model Years																				
		2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028
Study Years	1	1																				
	2		1																			
	3			1																		
	4				1																	
	5					1																
	6						1															
	7							1														
	8								1													
	9									1												
	10										1											
	11											1										
	12												1									
	13													1								
	14														1							
	15															1						
	16																1					
	17																	1				
	18																		1			
	19																			1		
	20																				1	

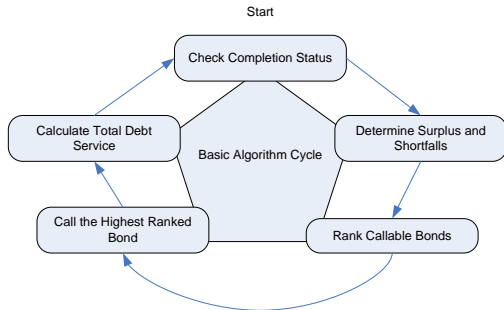
All Study Years exist on the diagonal line. However, we have the opportunity adjust how the model processes debt along the diagonal. The default behavior is to process all 20 Study Years one at a time. However, it is sometimes advantageous to travel the diagonal like skipping a stone on a calm lake. In order to make use of this ability, the model has been provided with a simple mechanism for controlling the behavior and it works like this:

- Period 1: 2008 – 2010, Interval 1 (Process every Study Year for the first 3 years)
- Period 2: 2011 – 2016, Interval 2 (Process every other Study Year for the next 4 years and the fifth Study Year as well)
- Period 3: 2016 – 2028, Interval 12 (Jump to the last Study Year and finish processing)

From the perspective of the model, decisions about which bonds to call from study to study exist as they did from the last Study Year that was processed. In Period 1, 2009 is initialized with all historical and projected debt retaining any calls that happened in 2008. Likewise, 2010 is initialized with all historical and projected debt retaining any calls that happened in 2008 and 2009. However, Period 2 begins processing at 2012, completely skipping over 2011. This means that in 2012, Historical and Projected debt will have calls from 2008, 2009 and 2010 only. It also means that 2012 is the first year that 2011 and 2012 projected bonds have been processed through the model. When the model begins processing it will attempt to first call debt into 2011, but from the perspective of the debt that exists in 2012. The reason this is an important concept is that bond ranking is changed by the introduction of additional debt and an increased number of years to process (2011 in the case of SY 2012) also introduces additional flexibility to the model, which can reduce the overall revenue required to satisfy the Total Debt Service. For completeness, after Period 2 finishes processing the model will process Period 3 (SY 2028) with all projected debt loaded from 2008 – 2028 and 2028 replacements. The last set of calls on debt will be from 2016 and the model will attempt to start calling debt in 2017 to resolve shortfalls from the perspective of 2028.

The second and arguably most important control is related to what we call Debt Classes. A Debt Class is a grouping of debts which have similar behavior in terms of Interest Calculation Method, Day Count Method, GAAP or Cash Redemption and other attributes. All Debt Classes in the system are user defined and they must have a value for all attributes (outlined in other documents). Typical Debt Classes are Treasury, Appropriation, Irrigation, Taxable and Non-Taxable bonds, CIBs, and Short and Long Term Leases. Users have the ability to adjust Debt Classes for a specific study instance or the administrator can adjust them at the default level inherited for all newly created studies. Obligations map to Debt Classes and the class directly affects the behavior of the obligation when the model is running.

The third and possibly equally important control is the ranking algorithm. As implemented in the model, the ranking algorithm is a set of sorting criteria which is applied to bonds callable in the Action Year only. The bond at the top of the list is called once per cycle during the calling phase. In general terms, the cycle looks like:



All Ranking criteria in the model are controllable by the user. They are specified as defaults for studies to inherit as well as changeable for a study instance. Ranking has a huge impact on the outcome of the model. The wrong criteria will cause the model to reach a solution that is higher in overall debt service than is really required. The correct criteria will allow the debt to settle into a low state where all shortfalls are resolved if possible.

The fourth mechanism that users have to determine the outcome of a study is through constraints on the data. Users have the ability to select sets of bond debt to allow them to be called or not. This is like a switch which is independent of Ranking. Ranking is dependent on the bonds that are callable during each Action Year. Altering the callable bonds will alter Ranking and provides a fine tuning control for modifying model behavior. The results of altering the callability of debt will probably not directly affect the overall results of the model, but it will determine which obligations can be redeemed early, thus impacting some flexibility to reduce shortfalls.

As an additional piece of control it is possible to augment the base ranking by changing the priority on each specific class of bonds. This can be used when a subset of bonds is desired for earliest or latest redemption during runtime.

As an example of priority and delay, consider the following:

<u>Class</u>	<u>Priority</u>
Appropriation	2
Treasury	2
Irrigation	3
Non-Federal	4

The Ranking that comes out of these Priority settings will be as follows:

- 1) Treasuries and Appropriations in Ranking order
- 2) Callable Irrigation in Ranking Order
- 3) Non-Feds in Ranking Order

This provides a good degree of control over the bonds that will be called through the system.

In addition to this, users have the ability to select which bonds are included in the model for processing. Bonds can be Allowed to process or Excluded from processing based on their Debt Class and optionally by matching criteria to bond properties. This can be used to answer specific questions like:

- 1) What would the debt look like if we didn't have some specific class or instance of a bond?
- 2) What would our payment schedule be if we included some debt not normally considered?
- 3) How would our payment schedule be affected if we replaced a set of obligations with some other obligations?

This is an attempt to allow modeling flexibility for answering questions that are not routine or might not have been anticipated during the design of the DBM.

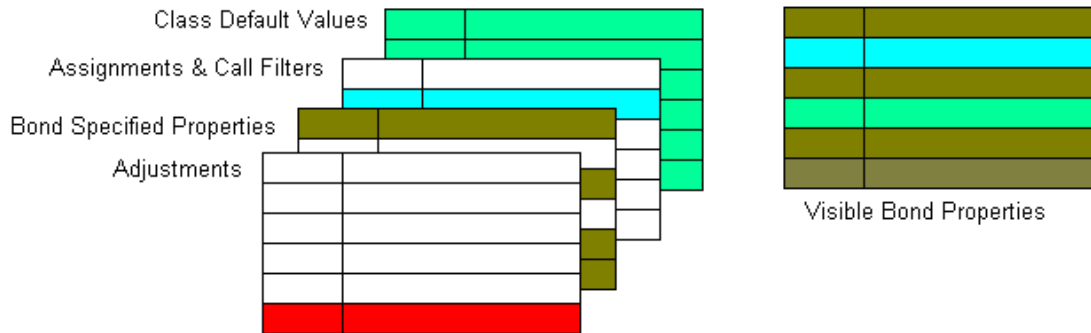
The next mechanism is the ability to alter bond properties at the start of a Study Run. The system provides the user with the ability to change a specific bond's properties so that, for instance, it is callable, rollable, the interest rate is some specific value, the first call date is adjusted to delay or promote calling or the Final Maturity date is set to some specific value to facilitate a specific refinance. These changes can happen to an entire Debt Class, a subset of a Debt Class, a specific bond series or a specific bond instance by stating the exact matching properties for the criteria.

At this point, it should be stated that the system provides a mechanism to view the selected bond debt when match criteria is entered into the system, regardless of where in the interface that criteria and constraint is entered. This provides the ability to double check so that the results match expectations based upon valid assumptions. An error in specifying criteria may not have a large impact on the outcome of the model, but it may result in a payment schedule for specific obligations that is not desired.

The final properties of each bond in the system follow a strict convention for precedence. Listing the precedence, highest first, for property source results in the following list:

- 1) Bond Specified Properties
- 2) Constraint Values – Assignments and Call Filtering
- 3) Class Default Values.

What this means is that any property which has a definite value on the bond is preferred over all others and the bond will have that value when it is processed by the system. If the property is null or empty on the bond and a value has been specified for it (through matching criteria or Debt Class) with a Constraint Value Assignment, then the bond will obtain that value. If the property does not exist on the bond and does not exist as an assignment, then the property from the bond's Debt Class will be used. After determining the value of the property, if an adjustment has been specified for that bond, it will be applied resulting in the final value of that property for the bond. The following diagram should help illustrate the point:



All debt objects require the following properties to be valid when processing:

- 1) Issue Name
- 2) Series Name
- 3) Component Name
- 4) Issue Date
- 5) Initial Principal
- 6) Maturity Date

As long as these properties are present directly on the bond, all remaining properties can be determined from the Debt Class. Additionally, none of these properties can be altered by a Constraint Value assignment statement. You may notice that Interest Rate is not a required property on the debt. When the Interest Rate is missing the value is obtained, as described previously, by first looking at an Assignment

and then by looking up the value from the Rate Forward and Rate Spread tables that are present in the Debt Class associated with this debt. The same activity occurs when a bond is rolled.

The last mechanism is a form of constraint call a Limit. A Limit is an upper boundary of some kind specified to avoid a very specific situation. One example of a limit is Borrowing Authority (BA). BA is a hard limit on the amount of outstanding Treasury principal that is allowed at any one time. In order to allow the maintenance and control of BA, the user will specify a constraint on Treasury Outstanding Principal of some value which will cause the model to pre-determine a minimum payment schedule for those bonds such that the limit is not violated. The main result of a Limit is to create what is called a Minimum Call Schedule for some obligations such that the bond will not contribute to violation of the Limit.

Limits can be placed on many result properties, all of them are totals for the debt class. Specifically, Limits can be placed on Outstanding Principal, Interest Expense, Call Amounts, Call Expense and Amortization Amounts. They can be further specified such that they apply only in a set of Study Years (Start Year, End Year). Some Limits do not pre-process debt, but apply during runtime, an example of that is Amortization Amounts, Call Amounts and Call Expense. When these Limits run, they directly influence the callable set of bonds so that Calls on a class of debt cannot exceed the Limit specified. Amortization is the sum of called amounts and maturing debts. If the Call Expense Limit is introduced and achieved during runtime, then the next bonds called from that Debt Class must be "At Par" or the debts in that Debt Class will be removed during the Call Filtering context.

NOTE: At this time, the Limit functionality has not been debugged and is not available for use.

Though not directly related to the model, there is one additional control which should be mentioned. The user also has the ability to place a Payment Plan on any specific obligation, not at the series or component level, but on the maturity itself. The purpose of the Payment Plan is to allow the user control over when a specific obligation will be called. Call schedules on a bond have precedence. Listed in order of importance:

- 1) Payment Plan (User Defined Calls)
- 2) Model Defined Calls (through model processing)
- 3) Limit Defined Calls (Minimum Call Schedule)

When the payment schedule of a debt is calculated, the Payment Plan (if it exists) is applied first. In the situation that a bond is not called in full at the end of the Payment Plan, the model is allowed to apply calls to the bond. However, all Model Defined calls must occur after the last year of the Payment Plan. Occasionally, a bond will also have a Limit Defined Call Schedule which will prevent the bond from reaching Final Maturity. Limit Defined Calls act as a catch-all. If the bond is partially called during processing, the Limit will full call the bond by the end of the Limit Defined Call Schedule. The model is not able to place Model Defined Calls after the first date in the Limit Defined Call Schedule. To be specific, the three call schedules have the following date relationship.

User Defined Call Date < Model Defined Call Dates < Limit Defined Call Date

Though the dates do not overlap, it is possible for a bond to be Full Called before the Model and Limit Defined Call Schedules are processed. Further, User Defined Calls and Limit Defined Calls are fixed (unchanging) during model processing.

Model Parameters and Contexts

The model has specific parameters which must be set in order to process a debt schedule. They include:

- 1) start and end fiscal years
- 2) the number of Study Years to process and intervals if desired.
- 3) datasets to use when running the model
- 4) cash flow table values (AC Credits, Hard Wired Amortization amounts)
- 5) floating income rate
- 6) fiscal year end date (for GAAP processing)

All of these parameters are required for a model to work correctly. Everything else is either a Constraint in some Context of the model processing (Call Ranking, Call Filtering, Assignments, Adjustments, and Exception) or Data. The relationships between Parameters, Constraints and Data are:

- 1) Parameters control the behavior of the model only and may specify data to process.
- 2) Constraints alter the view of the data in some way (ex: remove debt, add debt, make debt callable or not, change debt properties)
- 3) Data is the information which is processed and transformed through the DBM in order to achieve the overall goal.

Parameters exist at the highest level in the Context tree, before the model even starts running. They are used to initialize the model. Constraints exist at several levels in the model but are divided so that they affect very specific portions of the model during runtime. Each of the contexts mentioned previously has a specific function:

- 1) Exceptions – allows the user to add or remove debt from processing
- 2) Assignments, Adjustments– allows the user to alter bond properties with values inherited from the defaults on the bond's Debt Class.
- 3) Call Filtering – allows the user to add or remove a bond from call consideration, assuming that the bond does not have the callable flag set on it directly.
- 4) Call Ranking – allows the user to alter the priority and sort consideration for use when choosing the best bond to call.

All of the controls discussed previously fall into these contexts when used and the user interface divides them clearly so that they can be utilized with full knowledge of where they apply in the model.

GAAP Calculations

GAAP calculations are used to determine the amount of money from interest, expense and some principals which is allocated into a prior fiscal year. The equation used to determine this GAAP fraction is:

Day Count = Days from First Maturity Date to First Fiscal Year End Date

Days per Year = the number of days per year in the day count method

Days per Month = the number of days per month in the day count method.

Year Fraction = the average number of days in each month of the year.

R = a rounding factor and has a value of 0.5

$$\text{GAAP Fraction} = (\text{Floor}((R + (\text{Day Count} / \text{Days per Month}))) / \text{Year Fraction})$$

Prior Fiscal Year Allocation = Payment Amount * GAAP Fraction

Current Fiscal Year Allocation = Payment Amount * (1 - GAAP Fraction)

This equation will calculate the fraction of a year represented by the amount of time elapsed from the maturity date for any obligation to BPA's fiscal end date in the same year. This only needs to be calculated for the first year of a bond's life and doesn't change for any subsequent year. It is recognized that different day count methods will use a different number of days per month and days in the year, however the differences are fairly small. To avoid complexity and because the system is for simulation rather than accounting, the decision was made to approximate the days per month, days per year and year fraction values from the above equation. The following table represents the values as designated by the different day count methods:

Day Count Method	Average Days Per Month	Days Per Year	1 Month Fraction of the Year
30 / 360	30	360	0.08333
Actual / 360	30.16666 – 30.66666	360	0.08379 – 0.08518
Actual / 365	30.16666 – 30.66666	365	0.08265 – 0.08402
Actual / Actual	30.16666 – 30.66666	365	0.08265 – 0.08402

These numbers would be slightly different during a leap year as shown in the following table:

Day Count Method	Average Days Per Month	Days Per Year	1 Month Fraction of the Year
30 / 360	30	360	0.08333
Actual / 360	30.33333 – 30.66666	360	0.08426 – 0.08518
Actual / 365	30.33333 – 30.66666	366	0.08288 – 0.08379
Actual / Actual	30.33333 – 30.66666	366	0.08288 – 0.08379

While we approximate these 3 constants, it is important to realize that the actual count of days between dates is accurate for the day count method. The following table contains examples of each day count method:

Day Counts by Method					
Day Count Method	Start Date	End Date	Days in the Year	Year Fraction	Day Count Result
30 / 360	1/1/2008	6/1/2008	360	0.41666667	150
Actual / 360	1/1/2008	6/1/2008	360	0.41530055	152
Actual / 365	1/1/2008	6/1/2008	365	0.42222222	152
Actual / Actual	1/1/2008	6/1/2008	366	0.41643836	152
30 / 360	1/1/2009	6/1/2009	360	0.58333333	150
Actual / 360	1/1/2009	6/1/2009	360	0.58630137	151.4137
Actual / 365	1/1/2009	6/1/2009	365	0.59444444	151
Actual / Actual	1/1/2009	6/1/2009	365	0.58630137	151

The light green color is for values associated with a leap year. The Turquoise color is for non-leap year dates. These results are from actual day count execution and should represent the true number of days between the two dates. When performing GAAP calculations, a true day count is used as shown in the formula above. However, a small error is introduced in the GAAP calculation in the form of the “constant” days per month, days per year and year fraction values for each method.

The following values were used as the constants for each day count method:

Day Count Method	Average Days Per Month	Days Per Year	1 Month Fraction of the Year
30 / 360	30	360	0.08333
Actual / 360	30.41666	360	0.08449
Actual / 365	30.41666	365	0.08334
Actual / Actual	30.41666	365	0.08334

These values were picked because they represent the midpoint between the previous range values and in the general case, are very close to correct most of the time.

Architecture and Design Considerations

The design of the model attempts to acknowledge the complexity of the domain problem as well as the potential number of combinations for properties that may exist. In order to enhance future maintenance and reduce the need for intensive testing every time the system is updated, it was designed using a plug-in based architecture. Everything from the Algorithm to the Interest Payment Calculations is implemented as a single, loosely coupled, highly re-usable component following our custom plug-in interface guidelines.

Each Interface (discussed elsewhere in greater detail) is specifically designed to function in one and only one context. Every plug-in registers with a parent controller which stores information about the Context, name, location, version and commands recognized by the plug-in so that they can be dynamically used in the interface as needed. When new modules need to be created, they can be built and tested individually for correct functioning and then installed and registered for use in the model. This greatly enhances the ability of the model to outlast its current implementation and grow in capability as the user's needs change.

The entire system is built as a web delivered, multi-process, multi-threaded, modeling environment for debt repayment studies. The main engine will run for long periods, which is not really the intention of web based applications. The web is build around the idea that transient, stateless connections will be used. If that were the case here, then the model would be forced to run outside of the web environment. The engine thread has the effect of keeping a process running long after it would normally time-out at the web server. We were forced to increase the process timeout to more than 120 minutes so that longer running models would have time to finish processing. In addition, the session timeout needed increasing to the same amount so that the user's session was valid until the process ends.