Using
Dynamic Aggregator

applied FINANCIAL

July 2005

Contents

Introduction

This guide introduces ***Dynamic Aggregator***™, a software application from Applied Financial Technologies (AFT). AFT provides an industry leading suite of end-user applications and developer's tools used to model, characterize, and project mortgage security prepayments.

This document is written for an end-user who is new to Dynamic Aggregator. A simple scenario is used to illustrate the application's main features and some basic techniques for using them.

Throughout the guide, instructions you can perform are shown in a gray shaded box such as the following.

> Select an item from the *Report Type* drop-down list.

Feel free to use the instructions as a model, changing them as your interest and needs dictate. Early experimentation is a key to successful use of Dynamic Aggregator.

By the end of the guide, you should be able to navigate the application and exercise its basic functions well enough to begin your own analytical project.

Overview

Dynamic Aggregator is a software program used to analyze mortgage loan portfolios and pools with respect to their propensity to prepay.

The Valuation Challenge

All participants in the mortgage markets – investors, traders, portfolio managers, risk managers, or loan servicers – grapple with the same fundamental challenge: accurate valuation of mortgage portfolios and mortgage-backed assets.

The presence of an implied option in mortgage based instruments – resulting from the borrower's right to prepay the mortgage – complicates the valuation procedure. If poorly implemented errors in valuation, risk assessment, and pricing can result.

Most market participants measure and analyze prepayment risk at the pool level with a de facto set of elements that include *weighted average coupon (WAC)*, *weighted average maturity (WAM), loan type*, and *original term*.

While these attributes are widely available, analysis based solely on them masks the fact that loans/pools having the same WAC, WAM, loan-type, and original term can differ markedly in their respective propensities to prepay. This implies that they have different economic values, different risks, and should command different prices.

A deeper understanding of a portfolio's prepayment behavior requires analysis based on a broader set of loan characteristics. Attributes such as original loan size, loan-to-value ratio (LTV), property location, borrower's income and credit score are good examples.

Dynamic Aggregator allows you to define, control, and conduct this form of extended analysis on your proprietary mortgage data in your own computing environment. Going far beyond published prepay speeds and agency provided data, Dynamic Aggregator gives you a custom tool to meet the valuation challenge.

Data Warehousing

Over the past few years the widespread availability of powerful relational databases (Oracle, MS SQL Server, Sybase, etc.) and industry standard desktop PCs helped spark an IT industry trend known as *data warehousing*. In a sense, Dynamic Aggregator is part of this trend.

Let's start with a definition. In brief, a data warehouse is:

> A collection of data gathered and organized specifically for the purposes of analysis, synthesis, and in-depth understanding of relationships between data elements.

Dynamic Aggregator fits the definition of a single-subject data warehouse (sometimes referred to as a *datamart*) in that it contains information only about mortgage loans. Dynamic Aggregator's architecture reflects several aspects common to data warehouses including:

Single Fact Table – fact tables are the central point of focus in a data warehouse. They contain the measures and primary data that are of interest to the business.

In Dynamic Aggregator, this table is known as the *Single Loan Table*. The single loan table is created from the data your organization uses to trade, manage, and/or service mortgage assets. Your data – formatted to Dynamic Aggregator's specs – is loaded into the product to provide the basis for analysis.

Aggregation – a fundamental part of the application – is used in two ways.

Summarization

First, *summarization* (referred to in Dynamic Aggregator as 'pre-aggregation') is used to intelligently reduce a large set of discrete data points (loans in Dynamic Aggregator) to a manageable number without sacrificing precision. This is necessary because most firms have enough loans to result in a single loan table that is far too large to use as a primary source for analysis.

Here's how it works. Given a portfolio containing more than 2 million loans, it's not hard to imagine that there will be several – perhaps hundreds of thousands – that will be alike with respect to certain elements such as WAC, WAM, original term, loan type, and so forth. While the loans will necessarily differ with respect to other attributes – current balance outstanding, borrower income, and so on – aggregation can be used to summarize similar loans to reduce the overall size of the data set that supports the analysis. The result is a significant reduction in the time it takes to perform various analytics while preserving the integrity of the original data.

Dimensions

In addition, Dynamic Aggregator aggregates data on *dimensions*. In everyday use, we think of dimensions as referring to physical properties of an object - such as length, width, and height. In data warehousing terms, dimensions describe the properties of the data that drive the analysis.

In Dynamic Aggregator, the data dimensions – known as *pool sets* - are completely user determined. You define which data attributes are used to form the dimensions as well as the grain of the dimension. (*Grain* refers to the level of detail at which the data is aggregated. Thus, the grain also dictates the lowest level of analysis that the result set can support.)

For example, you can choose the data attribute *Current Weighted Average Coupon (CurWAC)* to serve as a dimension. In so doing, you will also determine a set of ranges (referred to as *bins* in Dynamic Aggregator) that stratify the loan data according to the values found for current weighted average coupon. A sample implementation of this is shown in the following table:

Dimension (Element)	*Ranges (Bins)*
Current WAC	0.0% - 4.0%
	4.0% - 5.0%
	5.0% - 6.0%
	6.0% - 20.0 %

Table 1 – Sample Implementation of Current WAC as a Data Dimension

Aggregation performed using the implementation shown above would result in the loans being 'separated' according to the specified ranges. This means that you could separately analyze loans and pools with a current WAC less than 4%, between 4 and 5%, between 5 and 6%, and greater than 6%.

Note that for the example above you couldn't separately analyze pools whose current WAC was between 7% and 8%. Nor could you separately analyze pools whose current WAC fell between 5.5% and 5.75%. The grain of the dimension (that is, the design of the bin ranges) doesn't support those distinctions. (See the section on the *Create Pre-Agg* window for more information.)

Multiple Views – a key criteria of data warehouses is the ability to inspect the data from a variety of viewpoints.

Dynamic Aggregator provides a series of output choices that examine the aggregated data from several viewpoints relevant to portfolios of loans and pools. In addition, Dynamic Aggregator's *sub-pooling* concept provides a mechanism for implementing basic hierarchical relationships between primary data elements.

Many users find value in creating multiple pool sets – each with a slightly different set of dimensions – that provide even greater insight into their loan portfolios.

Graphic Output – most data warehouses are coupled with a GUI-based query tool that enables graphic visualization of the results.

Dynamic Aggregator includes an integrated report and graph generator that is cognizant of the data structures from the outset – even while allowing total freedom in defining and ranging the dimensions. Analytical results can be viewed online, printed as a text report, or saved as an Adobe PDF (Portable Document Format) file.

Dynamic Aggregator brings the best of the broader IT trend of data warehousing and integrates it with an advanced prepayment modeling engine to create a single package that is both powerful and flexible.

Dynamic Aggregator Process

At the outset, it's tempting to simply "dive into" Dynamic Aggregator and begin to conduct analysis. While this will undoubtedly work for some people, AFT has found that using Dynamic Aggregator effectively is a function of knowing beforehand "where you want go". In other words, a little bit of planning ahead can save a lot of time and effort.

Having successfully implemented Dynamic Aggregator across a range of clients, AFT has observed that the best results come from following a simple set of "best practices". They are described below:

1. **Visioning** – developing a set of questions regarding the loan portfolio and its prepayment behaviors and exposures. This step usually occurs 'offline', and many times crystallizes the need for the analytical outputs that Dynamic Aggregator provides.
2. **Data Translation** – refers to the activities of collecting your portfolio data, and the critical step of translating the fields, codes, status flags, and other indicators used in your in-house or outsourced loan portfolio management systems to the Dynamic Aggregator master database table. (See *Appendix 1* for more information.)
3. **Ranging/Scoping** – inspecting the single loan database using data elements that are candidates for acting as data dimensions. This step is facilitated by Dynamic Aggregator's *Explore DB* window and results in a validated list of dimension items as well as maximum, minimum, and range (bin width) values. (Frequently, users of Dynamic Aggregator are familiar enough with the loan portfolio that this step can be done informally.)
4. **Pre-Aggregation** – summarization of the single loan database using data reduction techniques. This step results in a manageable subset of the single loan table that retains its data characteristics. It also starts the definition of the dimension elements.
5. **Pool Creation** – the implementation of a defined set of dimensions on a pre-aggregated table. This step results in the formation of a dataset (known as a 'pool set' in Dynamic Aggregator) that supports analytical review. Pool creation differs from pre-aggregation in that a hierarchy of elements is supported.
6. **Analysis** – prepayment modeling applied to pool sets with results depicted via a variety of output reports and filters. This step results in a deeper understanding of the expected prepayment behavior of the loan portfolio and can validate or repudiate theories formed in the *Visioning* step.

The amount of time and effort applied to these steps will vary depending on your specific needs and your familiarity with the loan portfolio and its characteristics.

In the next chapter, we'll start exploring the features of Dynamic Aggregator and how to use them.

Using Dynamic Aggregator

Preliminaries

In this section we'll explore the basic features of Dynamic Aggregator through the use of a simple scenario. We'll step through the Dynamic Aggregator process introduced in the previous section to provide some structure to the discussion.

This approach isn't meant to be exhaustive – rather just the opposite. In understanding a basic use of the application and how to apply the general process, you will gain a solid working knowledge applicable to customizing your use of Dynamic Aggregator.

Scenario Context

Here's the scenario: a financial institution - such as a bank - holding a large portfolio of mortgage loans is called on to demonstrate the comprehensiveness of its risk management practices to internal auditors or outside regulators.

At the outset, we might ask: *How large is a 'large portfolio'*? Typically, institutions holding in excess of one million loans benefit the most from Dynamic Aggregator, but don't worry if your holdings are smaller. The application delivers 100% of its functionality regardless of total portfolio size.

For the remaining pages of this guide, let's imagine that we are part of the team putting together a response to the auditors/regulators and that our responsibility is to validate the firm's internally developed prepayment model by comparing its outputs to those acquired from Dynamic Aggregator.

Visioning

In this scenario, the visioning step is straightforward. The primary objective is to produce from Dynamic Aggregator a set of analytic outputs that can be compared to similar outputs from the internal system. Let's assume that the questions Dynamic Aggregator must answer are:

- What are the historical prepayment speeds for the loan portfolio?
- What are the likely future prepayment speeds?
- What effect will market conditions have on future speeds?

So – assuming the internal system can provide similar results – we are set to proceed.

Data Translation

Before Dynamic Aggregator can be used, original and current loan data must be extracted from your internal systems, formatted, and loaded into the product's relational database. It's tempting to relegate this as a back-office or IT type of

activity – and there's no doubt those groups are usually involved. However, AFT has found that it's **critical** for a 'front office' business expert be available for what might best be called *semantic interpretation*.

> An example may help. All too often key facts about your loans – facts absolutely essential to accurate prepayment modeling – are represented in fields with generic names such as *Loan Code* or *Status Flag*. It's unlikely that a back-office or IT staffer will fully grasp the content of such fields.
>
> Usually, an actual user of the system – someone from the portfolio group, for example – has the requisite knowledge of *what the fields mean*. That knowledge is needed to properly map the data to the Dynamic Aggregator format.
>
> We know it's not glamorous work, but doing a good job practically guarantees that Dynamic Aggregator will bring value to your situation. Cutting corners will almost always guarantee the opposite.

Because data translation occurs largely offline, for our purposes we'll assume that Dynamic Aggregator is onsite, available to you, and already loaded with data extracted from your company's internal portfolio systems.

(To learn more about the fields that are commonly loaded and made available in Dynamic Aggregator, see *Appendix 1*.)

That completes the preliminaries necessary to use Dynamic Aggregator. In the following section, we'll begin to actually use the product to build our analysis.

Logging In

Most Dynamic Aggregator users invoke the application via an icon, shortcut, menu choice, or other means to execute an application on their MS Windows based computer. When the program loads, a window similar to the following is displayed:

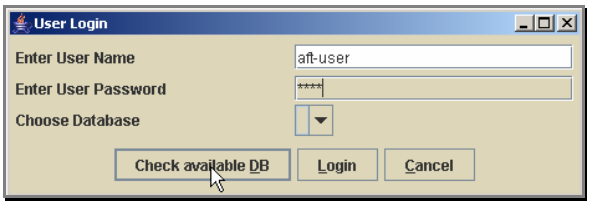

Figure 1: User Login Window

After supplying a valid *User Name* and *Password*, click *Check available DB*. This populates the drop-down list to the right of the *Choose Database* label.

Dynamic Aggregator supports multiple databases and is ideal for use in situations where there are several portfolios or pools of loans that must be separately analyzed. However, you may access only one database at a time.

To proceed, choose a database by clicking on its name. In the figure below, the database "AFT" has been selected. (Your database(s) may have other names.)

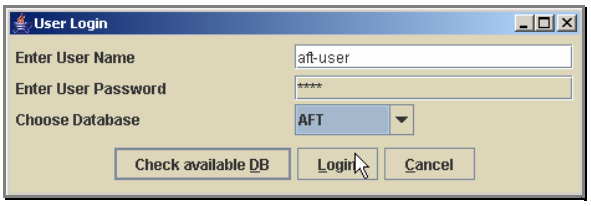

Figure 2: User Login Window

To continue and gain access to Dynamic Aggregator's windows and features, click *Login*.

Exiting Dynamic Aggregator

Ending a Dynamic Aggregator session is simple. Click the *window close control* -  - at the top right corner of the window.

Explore DB Window

In this chapter, the *Explore DB* window is introduced as a tool you can use to quickly get a handle on the characteristics of the loan database you are most interested in.

Ranging / Scoping

In all likelihood, you already have a handle on the characteristics of your loan portfolio. However, it is useful to know that Dynamic Aggregator provides an online tool – in the form of a window named *Explore DB* – that can visually represent those characteristics and validate your knowledge.

The objective of ranging the data is to arrive at a set of data attributes to use in pre-aggregation. The *Explore DB* window lets you work in an iterative manner – more or less proceeding from intuition to objective results.

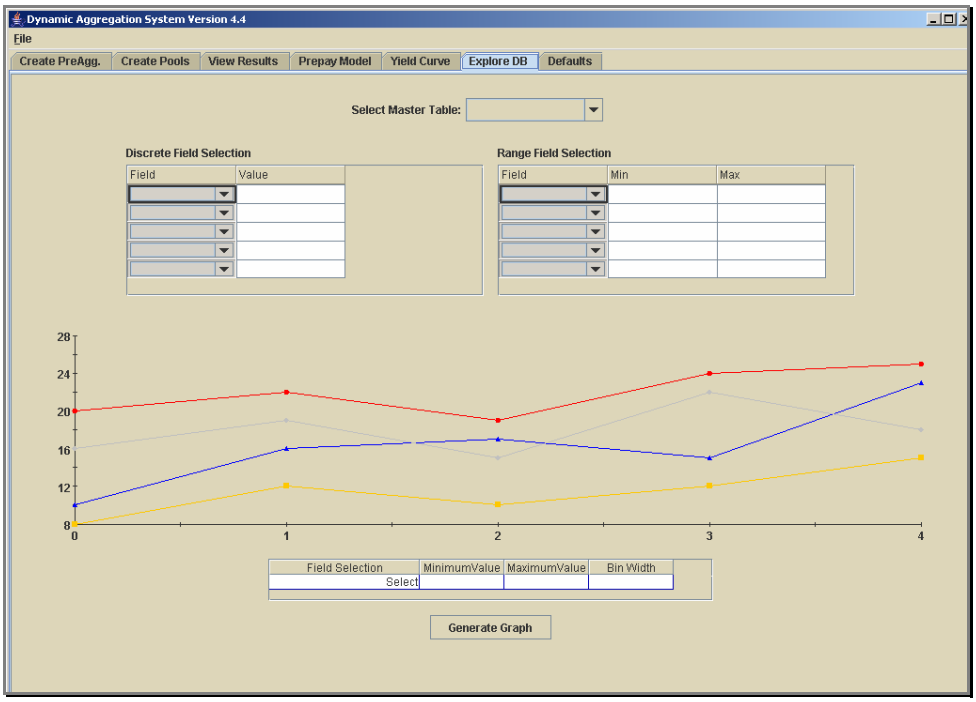

Figure 3: Explore DB Window

The *Explore DB* window works like a query engine to let you "range" any item from a large number of data elements

Ranging Original Loan Amount

We'll start with a very simple assumption that the loan characteristic *Original Loan Amount* is a good candidate for a data dimension. This section provides a procedure to validate that.

Figure 4: Selecting the Single Loan Table as the Master Table

To begin, from the *Select Master Table* drop-down list, choose "Single Loan Table" as shown above.

This directs Dynamic Aggregator to perform queries on the raw, unaggregated data loaded from your internal systems.

*There is only one **single loan table** in your Dynamic Aggregator database. All of the other entries that may appear in the* Select Master Table *drop-down are aggregations and are not useful to us at this stage.*

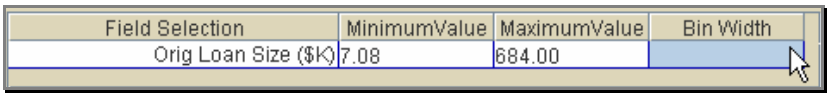

Figure 5: Selecting the data item Original Loan Size

To continue, from the *Field Selection* drop-down list (just above the *Generate Graph* button), select *Original Loan Size ($K)* as shown above.

Original Loan Size – along with many other data fields available in Dynamic Aggregator – is referred to as a **range field**. This means that the values stored in the data item are (or can be) continuous. (This is in contrast to a **discrete field** whose values are confined to a fixed list. We'll discuss an example of that later on.)

When you select a range field, Dynamic Aggregator automatically populates the minimum and maximum values that exist in the selected master table. These values set the widest possible range of values for the selected item.

Because in our example the single loan database has been designated as the master table, we know that there is at least one loan that had a loan amount at origination of $7,080.00 and at least one loan that had an amount of $684,000.

But at this point we know nothing about the distribution of original loan amounts for all of the other loans whose values fall between these two. The maximum and minimum values may represent wild, 'corner case' values or they may represent a significant number of single loans. To find out, we must query the data using **bins**.

In Dynamic Aggregator, a **bin** specifies the width of a particular stratum into