BI Blend Design and Reference Guide

7.2.4 Release

Table of Contents

BI Blend Overview

BI Blend is a "read-only" aggregate storage model designed to support reporting on large volumes of data that is not appropriate to store in a traditional OneStream Cube. BI Blend data is large in volume and most often transactional in nature. As an example, to analyze data by invoice, a standard cube would require metadata to store the data records. In a short period of time, most all the invoice metadata would be unneeded because of the transactional nature of the data. Therefore, storage in a Cube design is not a best practice solution for transactional data.

A key challenge to report on Transactional data is to present it in a uniform format supporting standardized reporting yet be flexible enough to support ever changing records and reporting requirements. The overall large size of the data sets requires a model suitable for responsive reporting and analysis.

BI Blend is a solution that approaches these challenges in a unique and innovate way. It is a solution that rationalizes the source data for uniform and standardized reporting, much like the Standard OneStream Cube models, but stores the data in a new relational column store table for responsive reporting.

The BI Blend solution is intended to support analytics on large volumes of highly changing data, such as ERP system transaction data, which typically would not reside in a OneStream Cube. The processing is unencumbered from the intensive audit controls within a traditional Consolidation Cube, such as managing Calculation Status.

Key Elements of BI Blend

- Flexible for change

- Fast Aggregation (through data as Stored Relational Aggregation)

- Single Reporting Currency translation

- Leveraged OneStream Metadata, Reporting and Integration tools

- Non-Cube, executed to a relational table optimized for reporting on large data sets by storing results in a column store index
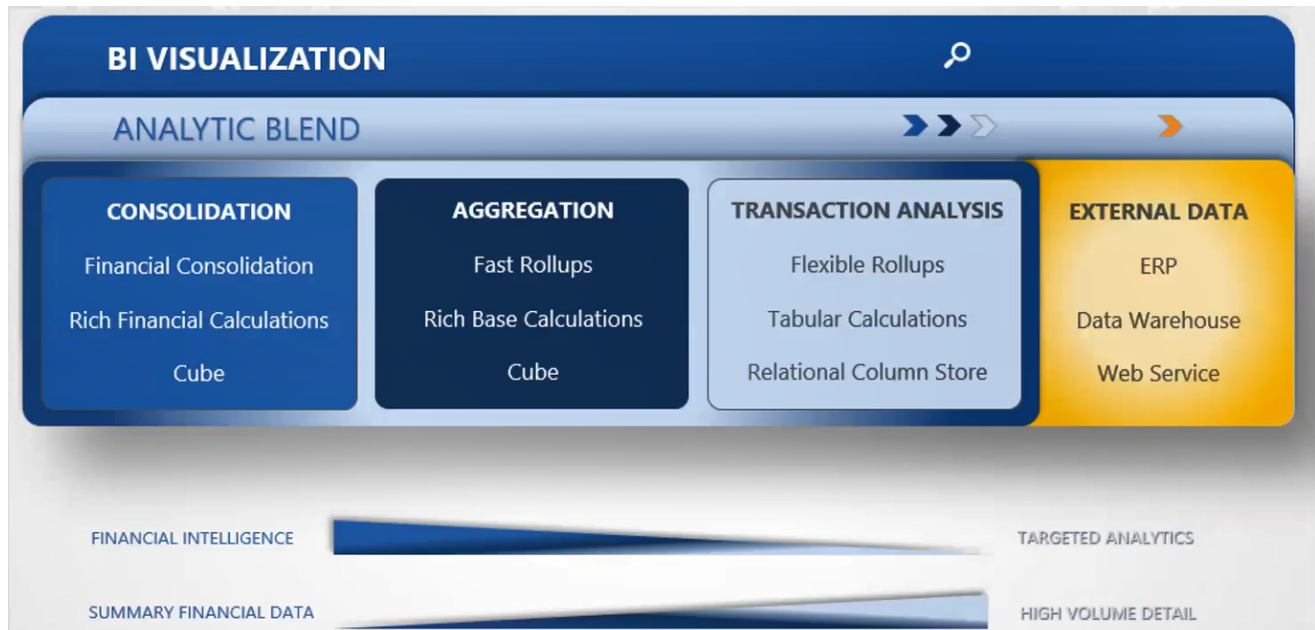
Key Model Differences

Consolidation Model

Typically defined by requiring a high level of precision and transparency in the data. This is related to precision in the accuracy of the data at base and parent levels, as well as integrity/transparency across time to support audit requirements. Supports flexible options for data collection and load or source data, manual inputs and input through Journal entry.

- Cube based

- Known Data

- Rich and complex financial calculations

- Predictable, scheduled data population

- Durable, sacred results

- Structured, uniform reporting format

- Financial Intelligence

- Stored Complex Calculated Data

- Parent level totals within Data Units are derived, such as Account subtotals.

Aggregation Model

The Aggregation Model is identified by an iterative process to populate the data. Reporting at the parent level Entities typically requires aggregated results, as opposed to the complex calculations, such as Eliminations, found in the Consolidation model. Supports flexible options for data collection and load or source data, manual inputs and input through Journal entry.

- Cube based

- Fast rollups

- Iterative data population

- Purpose driven results

- Rich/complex calculations limited to base level Entities

- Structured and Ad Hoc data exploration

- When you hover over the Aggregation Controls in BI_Blend Settings, the following message will display.

    "NotUsed" will return all source record members, Syntax for Aggregation Controls is - TopMember; ExpansionDetail (Optional); Label (Optional); Star Schema Control (Optional). Example - Houston;Children;ND or Houston;;;;;SSLeveled. Valid Expansions are Member, Children, TreeDescendants or SSOnly. Blank returns TreeDescnedatnsInclusive. Valid Labels are N, D, or ND, Blank returns Name. Valid Star Schema Control is SSLeveled.

Transaction Analytics Model

Data requires some level of aggregation but contains constantly changing information that should not be loaded to a cube.   Contains quantities of information that are generally unknown and change regularly, such as project codes and invoices. Data is loaded and updated in bulk, no trickle feed or incremental loads.

- Read Only Access

- Very Large data sets, transactional in nature

- Fast Aggregation as Stored Relational Aggregation

- Flexible and changing records

- Data may be related to Cube Data

- Data populated and rebuilt on demand

- Ancillary, supplemental data

BI Blend Features

BI Blend is a blend of Multidimensional and Transactional Data. This is done by utilizing the OneStream platform application to generate a database structure for OLAP reporting. By utilizing the existing OneStream Workflow processes, users need to be familiar with the interface and tools required to develop BI Blend reporting solutions.

Leverage Stage Engine

The Stage Engine is used to integrate the transactional data. The source data is processed through the standard Stage Engine. Here the data uses the common OneStream tools to Parse and Transform the records. BI Blend source data will contain additional member which are defined in the integration as Attribute members. The Stage Derivative Rules can be used to enhance the data with groupings available in reporting.

New BI Blend Stage Cache Engine

This is a new "in memory" process unique to the BI Blend feature.   This new Engine is used to rationalize the transactional data by leveraging the OneStream Metadata Engine. During processing the BI Blend Cache will utilize the Cube dimensions to transform the records into a unified reporting format. The Cube Hierarchies are used to derive aggregation points to be stored for reporting. The BI Blend Stage Cache will also accept properties on Entities and Accounts to perform direct method translation to the application reporting currency. The Engine ultimately generates a finalized table with a Column Store Index, creating a structure for OLAP reporting. The table can be accessed with the standard OneStream BI Dashboard and Pivot Grid reporting tools for analytic reporting as well as Relational Blending into Cube Views and Dashboards.

Transaction Analysis

WORKFLOW

STAGE CACHE

IN MEMORY PROCESS

PARSE

DERIVE

TRANSFORM

DERIVE NEW

USES STANDARD
STAGE ENGINE

STAGE CACHE NEW

IN MEMORY PROCESS

BI BLEND
CACHE

TRANSLATE

TABLE PARTITION

PARTITION
AGGREGATION

NEW ENGINE
READS/CONSUMES
ANALYTIC METADATA

BI Dashboards

BI Blend

FINALIZE TABLE
WITH
COLUMN-STORE
INDEXES

SQL Server
COLUMN STORE
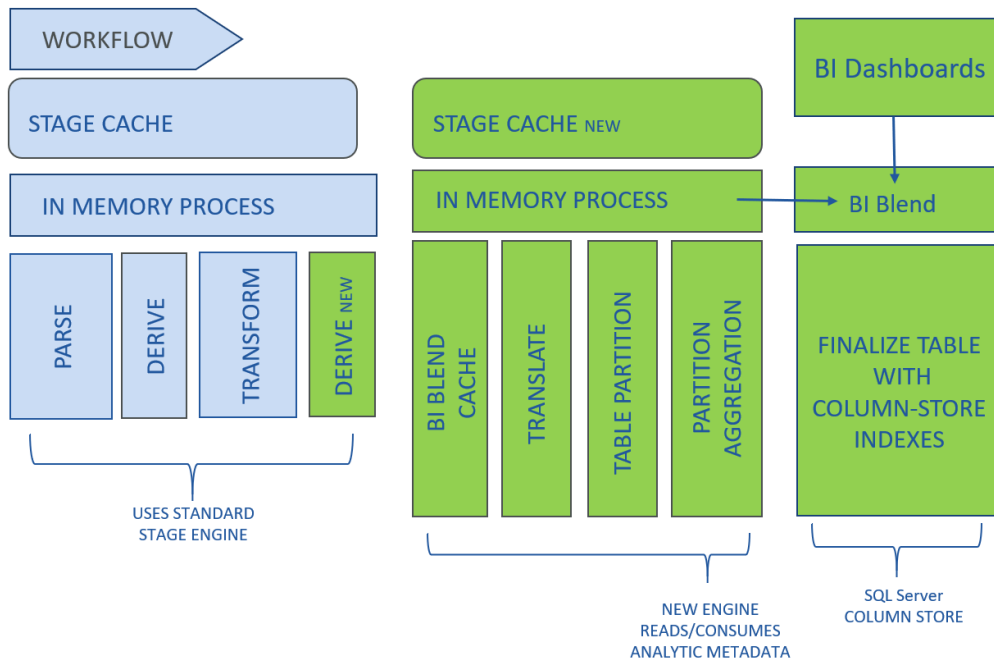
Use Cases for BI Blend

BI Blend is intended to provide focused reporting tables that are aggregated and saved as stored parent intersections for fast reporting at a later point in time. BI Blend is not intended to replicate and entire cube, but rather focus on specific reporting use cases that result in many parent intersections that would not perform well under Calc-On-Fly aggregation.

BI Blend also solves for use cases that are not pure analytic reporting problems. Leveraging OneStream hierarchies, along with BI Blend configuration settings, it is possible to aggregate on a few dimensions (Entity or Account as an example) while including transaction information (Invoice number) that is not associated with a cube. The ability to combine the dimensional structure with transaction details allows for selective enrichment of transactional data.

Transactional Model

Use Case Example: Highly changing ERP transactional results must be analyzed to determine Project and Customer profitability. The extremely high volume of details, such as invoices and other customer details, would never be loaded to the conventional OneStream Cube. This is because the volume of metadata and sparsity would be difficult to manage and impact the overall application performance. Analytic reporting would also be impacted due to the volume of members available to derive results.

The BI Blend Solution for External Reporting of transactional detail provides a solution that does not impact the application size or performance, while providing seamless, integrated reporting.

This requirement is primarily driven to support standard reporting on fluid, transactional details. No base level calculations, requiring only aggregation and perhaps basic translation.

- No impact of the Cube metadata having to support transactional details, such as names, invoices or codes, which routinely change.

- Leverage Cube Dimensionality for hierarchies for aggregated reporting

- Leverage Transformation rules to standardize reporting in approved naming conventions

- High performance reporting using column store index relational table better supports Analytic Reporting.

BI Blend Aggregation Model

The Aggregation model is one where the driving factors may include base level calculations and quick aggregation for reporting summary results. Rich calculations on base intersections can be performed within the Cube. BI Blend may be utilized to meet the requirement for fast aggregation, bypassing the overhead at parent levels, such as Calculation Status and managing Intercompany Eliminations.

- Utilize Cube capabilities for base level calculations

- Bypass the overhead and time required for aggregation using the finance engine / cube

- Aggregate based on Cube Dimensionality, or define a new hierarchy, (outline cube/special BI Blend hierarchies)

- Fast turnaround for reporting on base level member changes

- High performance, high-volume reporting using column store index relational table

BI Blend Configuration Overview

BI Blend is designed around all the common elements used in all OneStream applications. It does not require additional training. Any users familiar with setup, Workflow and Reporting will be comfortable using BI Bend as it utilized all the functionality around:

- Data Integration

- Metadata

- DataSources

- Transformation Rules

- Workflows

- Reporting tools and Dashboards

Workflow and Cube Settings

The interface to the BI Blend Model is the OneStream Workflow interface. The use of Workflow is a key element of BI Blend. The Workflow defines establishes the Cube Dimensions which are used to derive the metadata and aggregation points in the resulting BI Blend / Transactional Analytics Relational Tables.

The designer of BI Blend will need to know the reporting requirements and understand the source of the data.

Cube and Integration Settings

The Cube's Integration Settings will define the Dimensions the BI Blend Engine can utilize for the blending of the transactional data to a unified reporting format.

A BI Blend Integration can use all the Integration Dimension elements.

- Cube Dimensions

- Label Dimensions

- Attribute Dimensions

- Attribute Value Dimensions

In the below example, the Integration settings on the cube are enabled for Attribute1 and Attribute2 to expand the details to be collected for BI Blend analytic reporting, such as and Invoice Number and Customer Code.

Cube and Cube Dimensions

The Cube Dimensions assigned to the Cube are used by the BI Blend Stage Cache Engine to define the target for transforming records. The assigned Cube Dimension's hierarchies will generate the aggregation points for the resulting BI Blend Relational Table. This is a primary difference between the BI Blend Model and a Standard OneStream Cube Model. Within each Data Unit, the parent levels, such as an Account parent, are derived dynamically in the Standard OneStream Cube Model. The BI Blend Model will derive these totals and save them as stored values in the output relational table.

Leveraging Dimensions

The BI Blend Engine will derive the hierarchical structure and summary subtotals based on the OneStream metadata. BI Blend records processed will use Transformation Rules targeting the Cube's assigned dimensions.

The Dimensions assigned to the Integration used by BI Blend, by Scenario Type for example, must contain the base member targets defined in the Transformation rules. Therefore, the resulting BI Blend aggregation can differ from that used in other reporting cubes. A dimension and its hierarchy can be unique for BI Blend reporting.

The BI Blend Stage Cache Engine functionality supports:

- Hierarchy Aggregation, no complex calculations

- Simple currency translation using the Direct Method only. Any destination currency can be defined, but only one per BI Blend process.

- Limited use of Member Properties

- Does not utilize dimension Relationship Properties such as Aggregation Weight, Percent Consolidate or Percent Ownership.

- Limited parent level calculations using Derivative Rules

- Basic Time math using Helper Rules

- Supports simultaneous multi-period data loads by record for up to 12 periods using Attribute Value Dimension Members

BI Blend Dimension Property Usage

- Scenario Properties

    - Workflow Tracking Frequency

    - Input Frequency

- Time

- ○ Time Defined in the BI Blend table name is driven off the Workflow Tracking Frequency or based on the Time defined in the source file

- Entity

  - ○ Currency is referenced for simple Translation

  - ○ No other Member or Relationship Properties are used

  - ○ Aggregation weights are not used and will double count alternate hierarchies

  - ○ BI Blend does not utilize Entity Relationship Properties. Therefore, Entities as a shared member is not supported.

- Account

  - ○ Account Types is used for hierarchical aggregation and translation rates

- Flow

  - ○ Aggregation weights are not used and will double count alternate hierarchies

  - ○ Does not recognize Flow Members impact of Switch Type on Account Types

  - ○ Does not recognize Flow Members impact of Switch Sign

  - ○ Does not support complex currency calculations or alternate input currencies

  - ○ No other Member or Relationship Properties are used

- User Defined

  - ○ Does not support data stored as User Defined Attribute Members

  - ○ Aggregation weights are not used and will double count alternate hierarchies

  - ○ No other Member or Relationship Properties are used

- User Defined 8

  - ○ The UD8 Dimension may contain members, organized in hierarchies, which are used to aggregate Attribute Members contained in the data records.

Cube Dimensions

- **Consolidation**: Is limited to functionality for Local and Translation. The results for Translation will be limited to a single target reporting currency per BI Blend process. Complex translation is not supported and only the Direct Method is used for all Account Types.

- **Origin**: The Origin member is only supported for data generated through the Workflow BI Blend Engine as the Import member

- **View**: View is not supported in BI Blend and must be derived in reporting tools or Time math Business Rules.

- **ICP**: ICP Partner detail can be included in the data records, but Eliminations are not performed.

BI Blend Specialty Cubes

BI Blend requires a Cube to determine assigned dimensions. Dimensions by Scenario Type or a Specialty Cube functioning as a dimension outline can be used to yield alternate reporting results than the standard application cubes.

Reporting by Time, Scenario and View

The Time details are controlled by the Scenario for Input Frequency and Workflow Tracking Frequency. The Scenario is defined through the Workflow point of view. The BI Blend process will always assign the Time as a stored column whether time is defined as a record or by Workflow. This is done to maximize the BI Blend table generation and reporting performance

The View dimension is not available or managed in BI Blend. Accumulating results, for YTD reporting, must be done in the reporting tools as a calculation or performed using business rules. The output results of BI Blend include an identifying column records identifying the Account Type to identify Flow and Balance Type accounts.

Additional BI Blend Details, Attribute Dimensions

External Cube information is collected using the Attribute Dimensions by enabling the fields on the Cube Integration Settings.
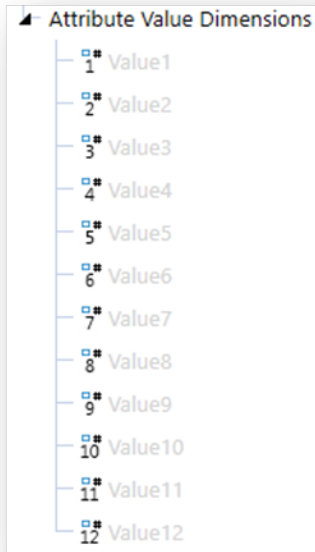
A major feature of BI Blend is the ability to utilize Attributes as an element of aggregations.   The Attribute Dimensions can be used to contain data record elements, such as "invoice" detail.   The Attribute Value is used in solutions to improve performance when loading large data sets for multiple periods (up to 12). In these designs, each Attribute Value is associated with a base period.

Designing BI Blend

Designing the BI Blend Workflow definitions should be determined by understanding how the data will be consumed and analyzed in reporting. The common use of BI Blend will fall into requirements within the models of Transaction Analysis and External Data. However, it can be considered to support an Aggregation model where the BI Blend aggregation processing may be a preferable solution of the standard Cube design.

Blend Unit

Determining which dimension should be defined as the Blend Unit is a key decision which will affect the performance of the BI Blend process and the reporting results.

The Blend Unit acts to define the dimension which used to break down the data into effective pages for processing, or partitions. Blend Unit's partition used to process the aggregations defined in the BI Blend settings, such as Account or UD aggregations.

Each Blend Unit member's aggregations are executed as part of a Multi-Threading process. The selection of the Dimension as the Blend Unit can impact the performance of the application. The larger the number of Blend Unit pages, the more opportunity there is for multi-threading tasks to be initiated.

Examples

Entity	UD3
Houston	CostCenter
South Houston	Admin
Houston Heights	(100 Base Members)
	IT
	(100 Base Members)
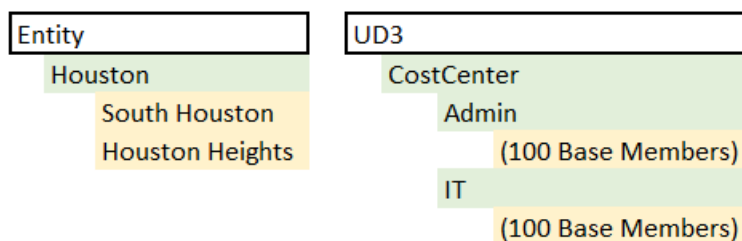
- Entity Structure as three members requires all aggregation to happen within only three members.

- Assigning the larger Dimension as the Blend Unit, such as UD3 (200 members) as the Blend Unit, would allow multi-threading to process more aggregations on smaller datasets.

- Larger Blend Unit members enhance performance through a more even distribution of records.

- When a Blend Unit page completes the aggregation process the engine loops over the rows on the page and summarizes any duplicate rows, but not with duplicates created in another page.

- Derivative rules run on each page, for a single row, in an exclusive manner for that page and do not cross pages.

- Many members in blend unit means more smaller pages which leads to better memory management, faster aggregation performance and more parallel processing.

Performance Settings

BI Blend processing is a CPU and Memory intensive process. The number of table records is heavily impacted by the Attribute details in the records and level of aggregations defined for Cube Dimensions. In each BI Blend configuration setting, the performance can be tailored to the environment with the Performance Controls properties.

- **Max Degree of Parallelism (No SQL)**: Defaults to 8 processors

- **Max Degree of Parallelism (SQL)**: Defaults to 4 processors

- **Row Limit**: Sets a maximum row limit to return to control potential server queries

- **Application Servers**: Allows a named server to be dedicated for BI Blend processing.

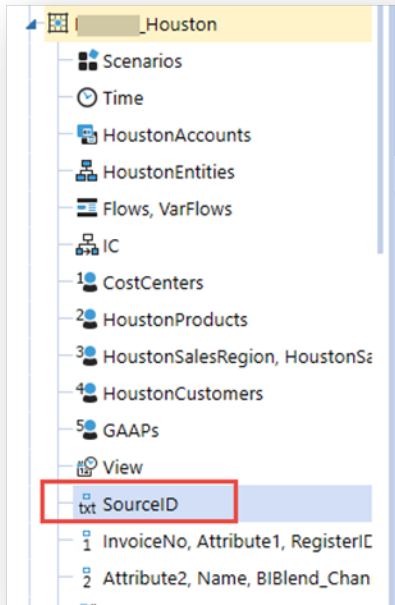| Performance Controls	
Max Degree of Parallelism (No SQL)	8
Max Degree of Parallelism (SQL)	4
Row Limit	10,000,000
Application Servers (optional, use commas and *?)	...

Blend Unit Partitioning

The concept of partitioning a Blend Unit is a performance solution for large data sets aggregating within a Blend Unit. This permits multithreading of the aggregations within the Blend Unit.

Blend Unit Partitioning is accomplished by assigning the Data Source "Source ID" property to a record field. The members within this field will be used as the key to partition aggregations within the Blend Unit page. The resulting records will be not be summarized across the Source ID / Blend Unit partition. Important to note, is the use of Derivative Rules in BI Blend function within a partition and cannot not reference data found in other partitions.   If required, the summarization will need to be performed in the database or in the reporting layer.

srcEntity	srcFlow	srcICP	srcView	srcAccoun	srcUD1	srcUD2	srcUD3	srcUD4	srcUD5	srcSourceID
H200	None	None	YTD	41100	None	HY-LT	1210	ES7349	None	CC100
H200	None	None	YTD	41100	None	HY-LT	1210	ES7349	None	CC100
H200	None	None	YTD	41100	None	HY-LT	1210	ES7349	None	CC100
H200	None	None	YTD	41100	None	HY-LT	1210	ES7349	None	CC100
H200	None	None	YTD	41100	None	HY-LT	1210	ES7349	None	CC200
H200	None	None	YTD	41100	None	HY-LT	1210	ES7349	None	CC200
H200	None	None	YTD	41100	None	HY-LT	1210	ES7349	None	CC200
H200	None	None	YTD	41100	None	HY-LT	1210	ES7349	None	CC200
H210	None	None	YTD	41100	None	HY-LT	1210	ES7349	None	KO001
H210	None	None	YTD	41200	None	IR-LT	1250	AH2347	None	KO001

BI Blend's assignment of a Blend Unit and aggregation generates "pages", which are visible in the Log File. Each "Blend Unit Page" cannot exceed 2gb. If the 2gb size is exceeded, the aggregation will fail. The error message "Error Array Dimensions Exceeded Supported Range" will be presented. A solution if impacted by size constraints is to increase the size of the Blend Unit Dimension being used or reduce the number of Dimensions used in BI Blend.

BI Blend Essential Design Considerations

Understand Data Records

- Any member not within the data record aggregation path will be bypassed. This means that a datasource can contain a complete set of records. BI Blend can then bypass those records by "filtering" or selecting an aggregation point that excludes those record sets.

- Attribute Members can be aggregated by being associated as a base member in a UD8 hierarchy. If the source record Attribute does not find a base target UD8, no error message is presented, and the record is ignored. This allows the data set to be flexed easily to adapt to reporting changes. Users should review the messaging for by passed Aggregations. If the member is not within the aggregation path it will be ignored.

Use of Common Members

Common Members reference metadata designs which have the same member names across dimensions. An example of Common Members is where Dimensions such as UD2 and UD3 both have members called "Top" with children as "None".

- Caution when using common members (Top / None) across dimensions as the common members may cause inconsistent results.

- If fully summarized intersections are required, the designer should consider selecting another Blend Unit.

- Limiting a Blend Unit to a member may not be optimal for BI Blend processing but will yield fully aggregated dimension results.

- If duplicate records are encountered, accept the duplicates but use aggregation queries when you consume or query the BI Blend Table (Group By on dimensions while performing a sum on Measures).

- Unique top members across dimensions are preferred, change dimension aggregation information to pick a parent that does not include common members, such as None or Top.

Understanding BI Blend Aggregation

Aggregation is a simple aggregation based on the Cube Dimension hierarchy. Additional aggregation points for non-cube / attribute members can be included in the results using Derivative Transformation Rules.

The BI Blend Engine aggregation utilizes the Cube Dimension hierarchies. This is done by evaluating parent members and identifying all Base members within the hierarchy. These base members are aggregated to the parent. Each parent within the hierarchy is evaluated using the same methodology. BI Blend processing has no concept of "sub-parent" rollups. Each parent is evaluated and aggregated according to the base members within its hierarchy.

Entity properties such as "Percent Consolidate" and other Dimension's "Aggregation Weight" are not used in the BI Blend processing. The aggregation is derived strictly from each Parent as a sum of its base members. Therefore, duplicate members within a hierarchy (shared members), should be avoided to eliminate "double-counting" of results.

BI Blend supports changing the "Blend Unit" from Entity as the "page dimension", to any other dimension in the Cube. This requires that Currency translation to be defined as a simple translation based on the Cube's default currency, not the Parent Entity's currency property. This allows the correct results when the Entity is not the Blend Unit, by providing a common currency throughout the aggregation levels to yield correct results.

Each defined aggregation is stored in a cache, and each can be calculated independently.

Cube	USD
Rate Type Revenues and Expenses	AvgRate
Rate Type Assets and Liabilities	EOMRate

TotalCorporation				
NorthEast				
Connecticut				
HartfordOffice				
StamfordOffice				
Massachussetts				
BostonOffice				
SpringfieldOffice				
SouthEast				
Florida				
DaytonaBeachOffice				
FortLauderdaleOffice				
Georgia				
AtlantaOffice				
SavanahOffice				

TotalCorporation	NorthEast	SouthEast	Connecticut	Massachussetts
HartfordOffice	HartfordOffice	DaytonaBeachOffice	HartfordOffice	BostonOffice
StamfordOffice	StamfordOffice	FortLauderdaleOffice	StamfordOffice	SpringfieldOffice
BostonOffice	BostonOffice	AtlantaOffice		
SpringfieldOffice	SpringfieldOffice	SavanahOffice		
DaytonaBeachOffice				
FortLauderdaleOffice				
AtlantaOffice				
SavanahOffice				

BI Blend Processing and Performance

The requirements related to the BI Blend environment will vary widely by the volume of source records together with the BI Blend Settings definition.

Default Server Selection

BI Blend processes will be queued across the available Stage servers. On the BI Blend WorkFlow Settings, a defined server can be assigned to dedicate all BI Blend processing.

Learning Mode

Learning Mode occurs during the first process instance of BI Blend and the design related to the choice of Blend Unit. This mode restricts multi-threading to two threads by two Blend Units to generate predictive statistics based on the number of records generated from the BI Blend settings. Subsequent processes will be optimized to multi-thread each Blend Unit. Should the Blend Unit be changed, the Learning Mode again be enabled. Additionally, if the number of aggregating dimensions is increased from the prior settings, the Learning Mode will again be enabled.

- Only two threads run

- Default mode when BI Blend Task is run for the first time

- Limited multi-threading is done to help ensure free memory is not exceeded

Log File Statistics

- Blend Unit

- Base Rows

- Parent Factorial

- Explosion Factor

Second Pass Processing

After successful learning mode, the same thread can be evaluated. BI Bend processing will observe the current number of rows in each Page Dimension (Blend Unit) and apply the Explosion Factor to determine if the process will exceed the amount of free memory available on the server.

- If a new Blend Unit is added to the file, the calculation estimations for logging and memory usage will use an average across the Page member statistics in its calculations.

BI Blend Database Table Creation and Structure

The output of BI Blend is an SQL Database table in a Column Store Index format. These are read only fact tables, optimized for reporting by having a high level of compression. The data is highly structured, containing the parent member values, much like Cubes. These tables do not need to be pre-defined or configured. The BI Blend process will create the data tables and corresponding error table automatically.

OneStream Application Database Tables

A database table captures the activity related to BI Blend tasks. This table has an associated MethodQuery in Dashboard DataAdapter as BIBlendInfo.

- StageBlendInformation table is a table generated in the OneStream application database tables to manage the tables generated through the BI Blend Workflow process

```
]SELECT TOP (1000) [Wfk]
        ,[Wsk]
        ,[Wtk]
        ,[TaskActivityID]
        ,[BlendTableDbLocation]
        ,[BlendTableName]
        ,[MapErrorTableName]
        ,[MapErrorsCount]
        ,[FailedCheckRuleCount]
        ,[FailedEventRuleCount]
        ,[BaseRowCount]
        ,[TotalRowCount]
        ,[StatisticsBytes]
        ,[Parameters]
        ,[TimeStamp]
        ,[UserID]
        ,[UserName]
        ,[LiveTotalRowCount]
        ,[LiveTimeStamp]
        ,[LiveParameters]
```

BI Blend Database Tables

The BI Blend assigned database will have tables created for each BI Blend Workflow Task by Workflow Tracking Frequency. Each time a table is created, a matching "error" table is created. The error table will be created whether or not an error was present.

- BI Blend Table - Each BI Blend task will create a table as:

    ◦ Prefix – "BIB_"

    ◦ Application Name

    ◦ Workflow Channel Name

    ◦ Workflow Scenario

    ◦ Workflow Time

For example:

⊞ ⊞ dbo.BIB_GolfStreamDemo_v18_BIBlend_AdministrationImport_Actual_2018M1
⊞ ⊞ dbo.BIB_GolfStreamDemo_v18_BIBlend_AdministrationImport_Actual_2018M1_ME

- BI Blend Error Table – Each BI Blend table will have a corresponding error table:
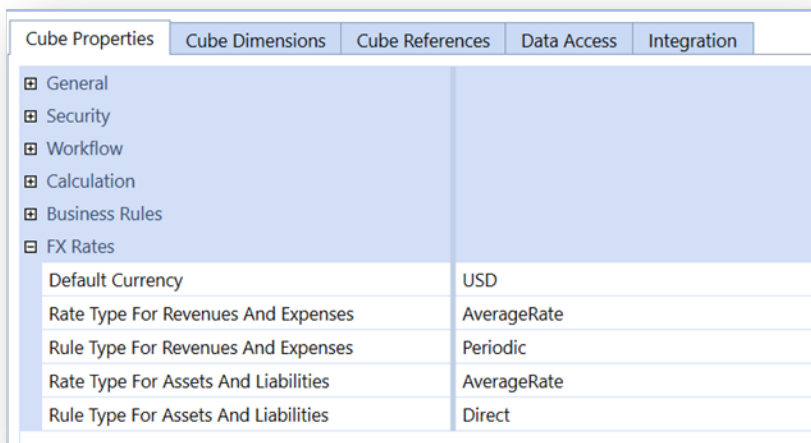 - Suffix – "_ME"

BI Blend and Cube Settings

A key element of the BI Blend solution is to rationalize data for BI-Reporting. The BI Blend Reporting solutions must be tied to a Workflow and the assigned Cube.

Cube Properties

Currency translation is primarily controlled by the Cube Properties. Rule Type is not used, all translations use the Direct Method. The rate is determined by Account Type.

- RateType for Revenues and Expenses – rate used by Account Type Property

- RateType for Assets and Liabilities – rate used by Account Type Property

Cube Properties	Cube Dimensions	Cube References	Data Access	Integration
⊞ General				
⊞ Security				
⊞ Workflow				
⊞ Calculation				
⊞ Business Rules				
⊟ FX Rates				
Default Currency	USD			
Rate Type For Revenues And Expenses	AverageRate			
Rule Type For Revenues And Expenses	Periodic			
Rate Type For Assets And Liabilities	AverageRate			
Rule Type For Assets And Liabilities	Direct			

Cube Dimensions

The Cube Dimensions are key to BI Blend results. The assigned Dimensions, by Scenario Type, are used to rationalize the data for reporting and derive the reporting subtotals.
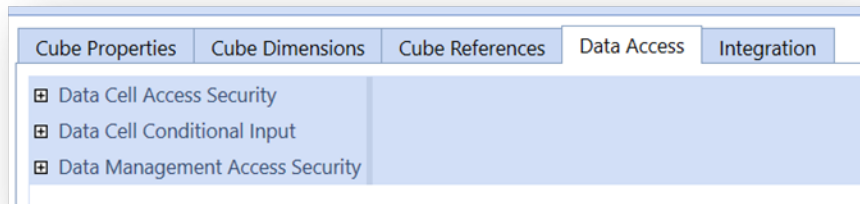
Cube References

Cube References, and the associated Extensibility they manage, are fully supported by the BI Blend Engine. Extensible Dimension hierarchies will be used if BI Blend is based on a top-level Cube.

Cube References, and the associated Extensibility related to extended Cubes and Dimensions, have limited support when BI-Blend loads are performed using a top level cube to load to extended cubes. Use of a top level cube to load to Extended Cubes and Dimension only supports two levels of Extensibility. The first being the Cubes Main dimension and the second being the next level extended dimension.

Data Access

Data Access and other data cell level controls are not supported in BI Blend.

Integration

The Integration for the Cube provides the Dimensionality available to the BI Blend Engine. The active Dimensions can be used in both the Transformed and Un-Transformed states just like all Data Integration processes in the standard Stage Processing. The activation of the additional 20 Attribute Dimension members can be activated to support the inclusion of non-cube data in the BI Blend output. The Attribute Value Dimension member can be activated and used in BI Blend to improve performance when loading large volumes of records containing many time periods by allowing all time to be associated by record.
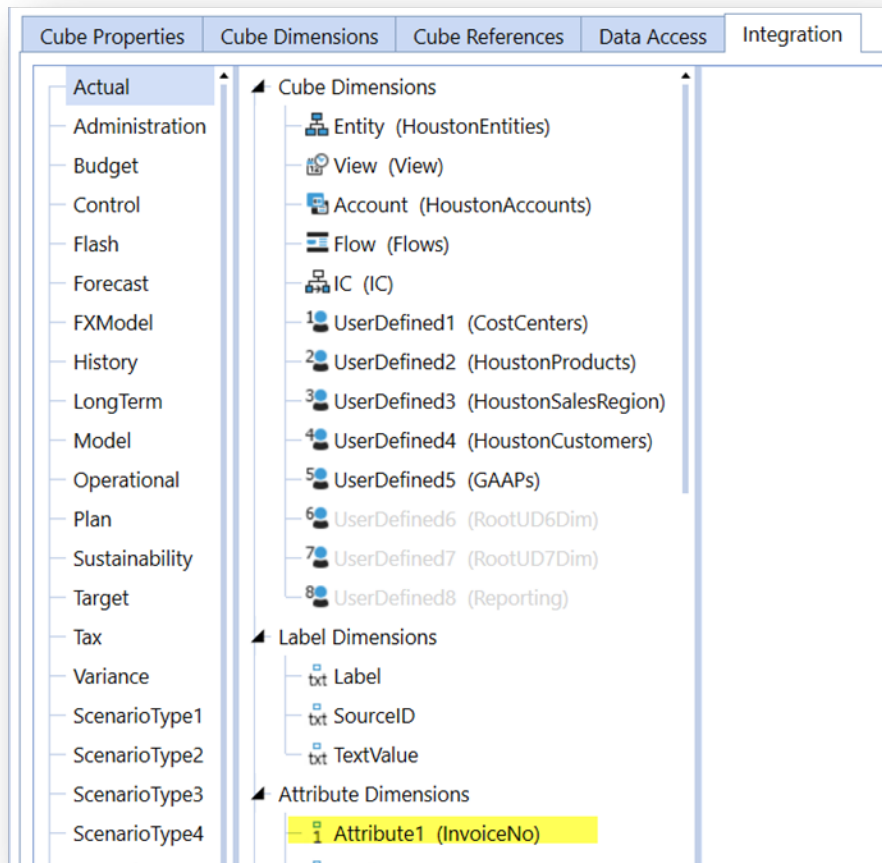
BI Blend Processing Example

Processing BI Blend used the standard Workflow environment and tools to provide the users with familiar environment and eliminates the need for any additional training.

Importing Blend Data

The Workflow has a new Workflow Name as Blend or Blend as Workspace. Users load data the same as other OneStream Workflows that commonly result in Cube data. Load and Transform will the process of data.

The configuration of BI Blend is managed in the Workflow Profile. However, some of the Stage Engine's Workflow Properties may not be valid for, or used in BI Blend configurations. The primary configuration for BI Blend is done using the Workflow Name and BI Blend Settings. The BI Blend Engine's architecture, such as in-memory processing, makes some WorkFlow properties, such as the Append Default Load Method, an invalid selection. Such selections will not have an effect on Workflow Blend behaviors.
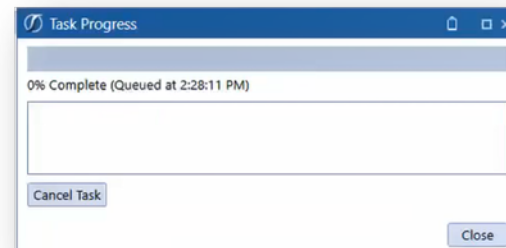
Data can be collected from:

- Files

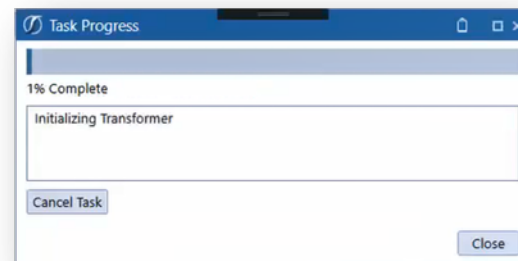- Connectors

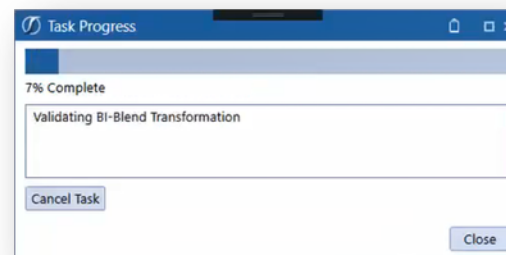- Other WorkFlow Stage Data
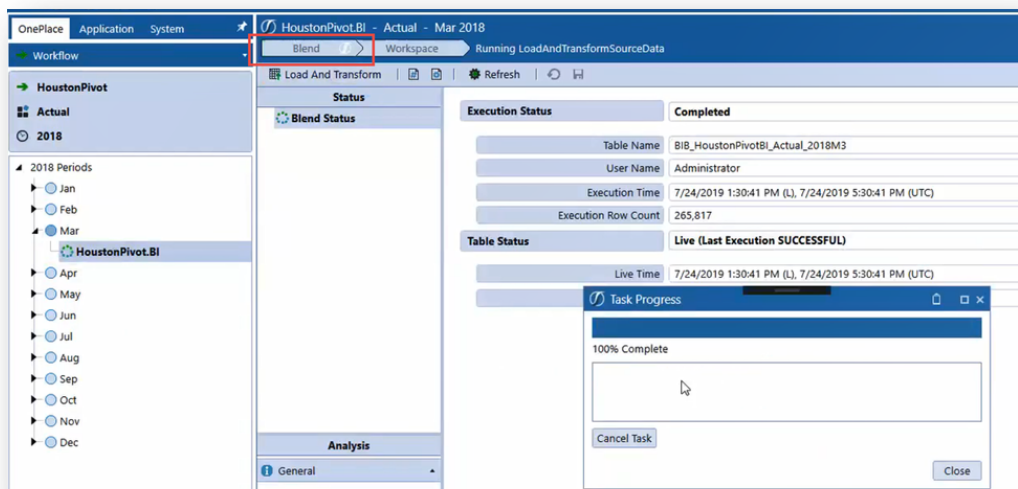
Blend Steps
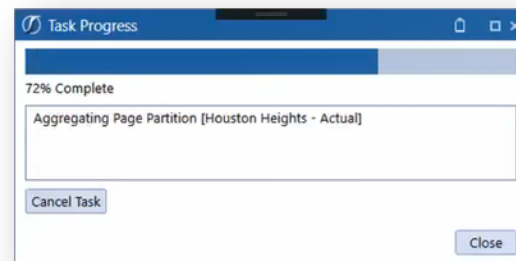
Data is Queued for Processing on a Data Management Server
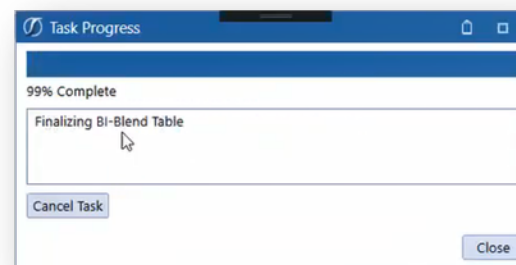
Cube Dimensions are integrated

Transformation Rules are executed

Multithreading of Page Dimensions or "Blend Units" begins

Write to target table

Validating Members

BI Blend is processing the records in relation to the Cube Dimensions and the assigned Transformation Rules to create a common reporting solution. Should any member in a record not find a target in the Cube Dimension through Transformation Rules, a mapping error will be presented. Validation is limited to target mapping. Validation of data intersections is not performed, such as those from Constraint properties.
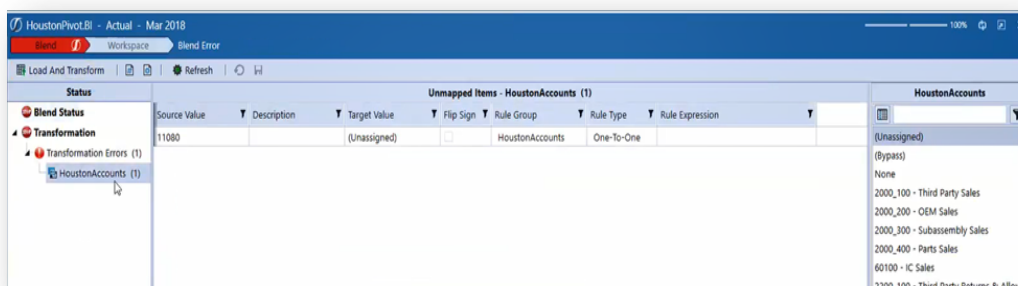
BI Blend Status

The BI Blend Status page displays the information for the current Blend task. This helps to indicate to the user what results are available to the users. Should there be a failure in the Blend task, the correction must be performed, and the entire task must be re-executed.

- Execution Status: Displays the results of the most current Blend task noting information such as target table name and time. If errors were encountered, the "Reload" status will be displayed to indicate the Blend task must be re-executed.

- Table Status: This status displays the state of current table to denote the last time the table was updated and its size.
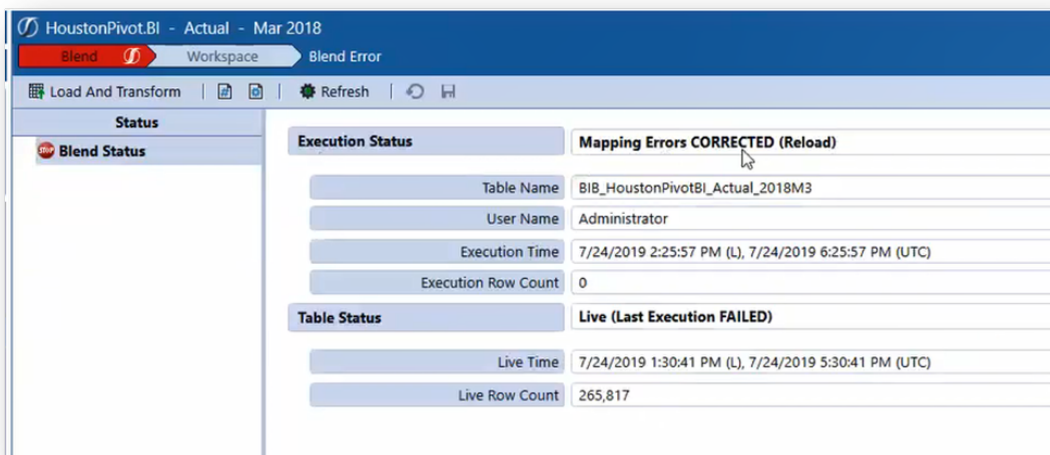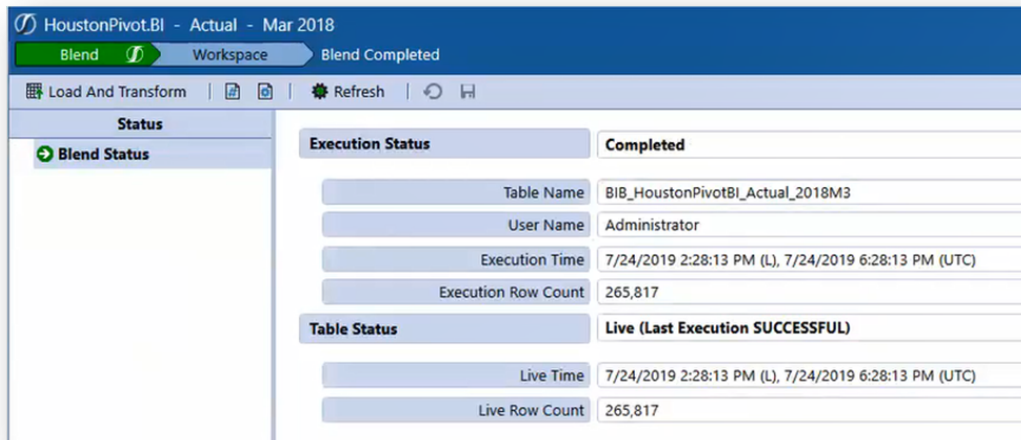
BI Blend Processing Logs

Upon the successful completion of a Blend task, statistics and information are written to the log file.

Basic Log File Parsing

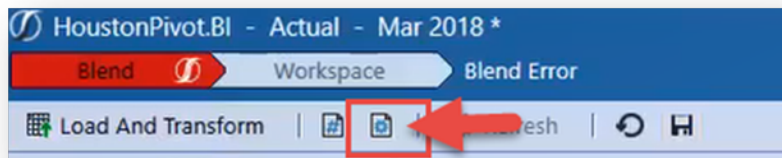The Log File is intended to provide statistics to manage the Blend task. The source file can identify data records. BI Blend generates records not only by the base member, but also on the Parent level hierarchies found in the associated Workflow/Cube dimensions. The Log File gives insight to this structure to estimate the processing needed on the current and potential future files being processed.

- **Blend Unit**: is the identified dimension used as a partition, generating page records

- **Base Page Rows**: the number of data record rows for the Page member

- **Parent Factorial**: total potential data cell intersections available for population at parent level.

- **Explosion Factor**: the number of data record cells generated by the base rows across the Dimensions and parent levels

```
AFTER AGGREGATION
*********************************************************************************
BI-BLEND ENGINE STATISTICS

        Houston
                Partitions: 1
                Base Page Rows (Observed): 1,049                    [Parent Factorial: 0]
                Base Page Size (Estimate): 490,932      [Bytes/Row (Estimate): 468]
                ----------------------------------------------------------------
                Explosion Factor (Observed): 4.9
                ----------------------------------------------------------------
                Exploded Page Rows (Observed): 5,163
                Exploded Page Size: (Observed): 2,416,284
                Persist Page (MS): 104
                Update Supporting Metadata (MS): 76

                Partition Processing Durations:
                ----------------------------------------------------------------
                RevenueMgmt
                        Aggregation (MS): 95
                        Summarize (MS): 85
                        Derivatives (MS): 0


        Houston Heights
                Partitions: 1
                Base Page Rows (Observed): 224          [Parent Factorial: 0]
                Base Page Size (Estimate): 108,416      [Bytes/Row (Estimate): 484]
                ----------------------------------------------------------------
                Explosion Factor (Observed): 6.0
                ----------------------------------------------------------------
                Exploded Page Rows (Observed): 1,344
                Exploded Page Size: (Observed): 650,496
                Persist Page (MS): 128
                Update Supporting Metadata (MS): 23

                Partition Processing Durations:
                ----------------------------------------------------------------
                RevenueMgmt
                        Aggregation (MS): 93
                        Summarize (MS): 16
                        Derivatives (MS): 1
```

Task Activity

A new Task Activity status has been added to monitor BI Blend processes, BI Blend Load and Transform. This tracking breaks out the various processing steps for analysis.

Technical Overview

BI Blend Workflow Settings

The primary design element of BI Blend are the BI Blend Settings found on the Workflow Import Channel, by Scenario Type. These settings allow the BI Blend Administrator to define and optimize the generation BI Blend tables to meet the reporting requirements.

Aggregation Controls	
Translate	NotUsed
Blend Unit Dimension Token	E#
Entity Aggregation Info	NA Clubs;Children
Account Aggregation Info	62000;TreeDescendants
IC Aggregation Info	NotUsed
Flow Aggregation Info	NotUsed
UD1 Aggregation Info	NotUsed
UD2 Aggregation Info	Top;Member
UD3 Aggregation Info	NotUsed
UD4 Aggregation Info	NotUsed
UD5 Aggregation Info	NotUsed
UD6 Aggregation Info	NotUsed
UD7 Aggregation Info	NotUsed
UD8 Aggregation Info	NotUsed
Attribute 1 Aggregation Info	NotUsed
Attribute 2 Aggregation Info	NotUsed
Attribute 3 Aggregation Info	NotUsed
Attribute 4 Aggregation Info	NotUsed
Attribute 5 Aggregation Info	NotUsed
Attribute 6 Aggregation Info	NotUsed
Attribute 7 Aggregation Info	NotUsed
Attribute 8 Aggregation Info	NotUsed
Attribute 9 Aggregation Info	NotUsed
Attribute 10 Aggregation Info	NotUsed
Attribute 11 Aggregation Info	NotUsed
Attribute 12 Aggregation Info	NotUsed
Attribute 13 Aggregation Info	NotUsed
Attribute 14 Aggregation Info	NotUsed
Attribute 15 Aggregation Info	NotUsed
Attribute 16 Aggregation Info	NotUsed
Attribute 17 Aggregation Info	NotUsed
Attribute 18 Aggregation Info	NotUsed
Attribute 19 Aggregation Info	NotUsed
Attribute 20 Aggregation Info	NotUsed

Setting	Function
Measure Type	Defines how the time dimension column will be determined in the relational tables.
Content Type	Controls the type and level of detail displayed in the relational tables for base level records only (analysis of source against aggregated parents is not supported). The designer controls how the BI Blend tables utilizes dimension hierarchies and the inclusion of Attributes related to the Cube Integration Settings.
Create Star Schema (Optional)	Creates the output as a Star Schema table set and related Views. Supporting Dimension tables will be created for each Cube Dimension assigned an Aggregation Control. The Dimension tables will contain column fields for **MemberName**, **MemberDesc**, **NameAndDesc**, **ParentName**, **IndentLevel**, **IsBase**and **MemberSeq**, which can be used in reporting against the Star Schema view. This is an optional setting.
DB Location	Sets the name of the SQL Database that holds the BI Blend results tables.
Explosion Adjustor	Estimates the size of tables generated, as derived from the initial Learning Mode.
Column Alias	Defines custom column names for the output database table. Spaces in the names are not recommended. Use underscores to represent spaces.

| Setting	Function
Translate	Enables translation to a single destination currency.
Blend Unit Token	Token to assign a Dimension to partition as a Page Dimension and the basis of the BI Blend multi-threading.
Dimension Aggregation	Parent member point to set top level aggregation member. Can be set as a parent level, children, list, distinct member or no aggregation.
Max Degree Non SQL Parallelism	The default setting is 8.
Max Degree SQL Parallelism	The default setting is 4.
Row Limit	Estimated by analyzing the Blend Log File, base level rows, and the Explosion factor to determine free memory usage.
Shrink After Finalize	Compacts the BI Blend database upon table finalization. The default setting is "True".
Application Servers	The BI Blend processes default to the Stage Servers, with queuing controls. This can be overridden with named servers.

Aggregation of Attributes

You can manage Attribute Dimension in the following ways:

- Attributes as records can be "aggregated" within a Pivot Grid as a reporting tool feature.

- Aggregations of Attributes can be calculated using BI Blend Derivative Rules.

- Attributes can be associated with a base member UD8 metadata members and aggregated as a member of a hierarchy.

Associating an Attribute with a UD8 base member requires the Attribute Dimension Record exists as a base member in a UD8 Dimension. Only UD8 can be used to generate Attribute Dimension Members for aggregation. Because the record must be reflected as a dimension member, Aggregation Attributes may not be appropriate for highly transactional, changing, record member.

sEntity	sAccount	sUD1	sAtt1	Amt
A300	gl10001	P110	20oz	1000
A300	gl10001	P111	20oz	1000
A300	gl10001	P171	16oz	1000
A300	gl10001	P200s	3.5oz	1000
A300	gl10001	P300s	1.5oz	1000
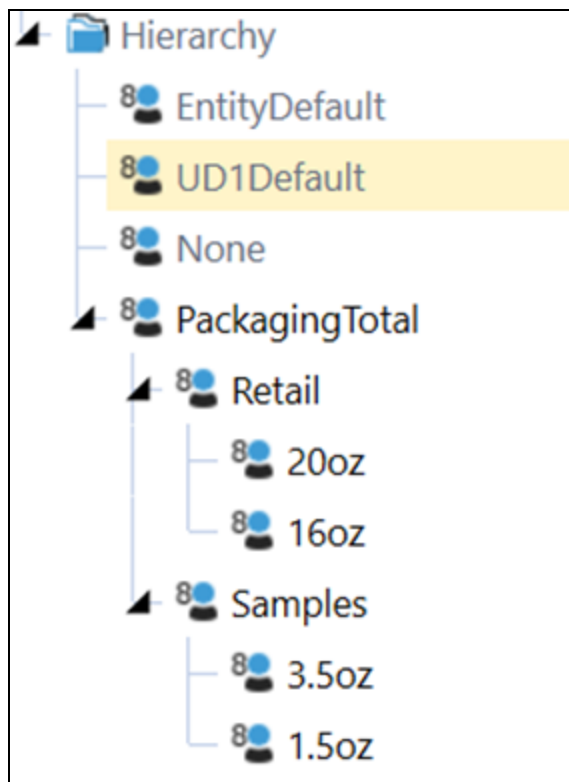
BI Blend Settings Options

Measure Type - Determines how Time will be assigned in the BI Blend Table. Time is always generated as a database column based on the file contents or Workflow settings.

- **TimeSource** – Create Time columns by the time referenced from members found as records in the source file.

- **TimeWFView**– Time columns are generated from the Workflow Tracking frequency. As an example, a Yearly Tracking frequency for a monthly input Scenario would create 12 time columns regardless of the number of periods in the data records. This option can be utilized for Time Math helper rules.

- **TimeWFViewAV** – This type is used to utilize the use of the Attribute Value Dimension. This solution requires Time based value records to be associated with Attribute Value Dimension members in the Datasource. Each Value (1-12) will be associated with a Time period column in the output table. This method is used to more efficiently process records in large multiperiod datasets.

Content Type- Define the type of detail written as records in the BI Blend results.

Example Source File

srcEntity	srcFlow	srcICP	srcView	srcAccount	srcUD1	srcUD2	srcUD3	srcUD4	srcUD5	srcSourceID	srcTextV	srcAtt1	srcAtt2	srcAmt
H200	None	None	YTD	41100	None	HY-LT	1210	ES7349	None	KO001	SuperGLSys	OneStream1	Commercial	5000

- **TargetCubeDims** – Results will be limited to the transformed Cube target members.

	Rt	SourceID	Entity	Cons	Scenario	View	Account	Flow	Origin	IC	UD1	HoustonProducts
1	0	KO001	Houston	USD	BIAct	YTD	2000_100	None	Import	None	None	Hybrid LT

- **TargetCubeDimsSource** – Results will include both the Source and transformed Cube target members.

100 %

	Rt	SourceID	sEntity	Entity	sCons	Cons	sScenario	Scenario	sTime	sView	View	sAccount	Account	s
1	0	KO001	H200	Houston	Local	USD	(Current)	BIAct	(Current)	YTD	YTD	41100	2000_100	

- **TargetCubeDimsAttributes** – When records contain non-cube dimension records, such as invoices assigned as an Attribute, TargetCubeDimsAttributes must be used to process and include the detail. Results are transformed Cube target members including Attributes.

Account	Flow	Origin	IC	UD1	HoustonProducts	HoustonSalesRegion	HoustonCustomers	UD5	BIBlend_Invoices	BIBlend_Channel
2000_100	None	Import	None	None	Hybrid LT	Southeast	East Sports	None	inv_OneStream1234	Commercial

- **TargetCubeDimsSummaryAll** – Output to table provides all available Source, Target and Attribute results.

sAccount	Account	sFlow	Flow	sOrigin	Origin	sIC	IC	sUD1	UD1	sUD2	HoustonProducts	sUD3	HoustonSalesRegion	sUD4	HoustonCustomers	sUD5	UD5	BIBlend_Invoices	
41100	2000_100	None	None	Import	Import	None	None	None	None	HY-LT	Hybrid LT	1210	Southeast	ES7349	East Sports	None	None	inv_OneStream1234	

ColumnAlias – Provided functionality to rename the output table columns using the Stage Table Keywords. If left blanks, the system will generate Column names based on the default table and dimension labels.

Prefix Alias Keys:

- User Defined: U1T through U8T

- Attributes: A1 through A20

- Label: Lbl

- TextValue: Tv

Example: to custom label columns for User Defined Dimensions and a Stage Attribute

U1T=CostCenter,U2T=Products,U3T=Regions,A1=DepartmentCode,A2=ExpenseID

Blend Unit Token- Assigns the Cube Dimension as the partitioning dimension to generate as pages and the corresponding level of multithreading. Available dimensions for Blend Unit are:

MaxMembersDim	Setting will evaluate the source and assign the largest member set as the Blend Unit
E#	Entity
F#	Flow
U1-U8	User Defined
A1-A20	Attribute Dimensions

Aggregation Controls – Used to set the parent level for the top-level aggregation. The intent of these filters is to limit the return of records to the parents that need aggregation for reporting. The complete dataset will be used, however, records outside any defined aggregation path will be ignored. As an example, if an aggregation control for Entity was set for US Clubs, any records present in the dataset for Entities outside the US Clubs hierarchy will be ignored. This allows BI Blend to generate results focused on specific reporting and analysis requirements.

When using BI-Blend with extended dimensions, BI-Blend will ONLY aggregate the ultimate base members of an extended dimension. Loading a member that becomes a parent as a result of dimension inheritance, such as mapping source data to a parent member, will result in NO aggregation for the extend parent.

Summary Aggregation Behavior

NOTE on Filtering – Applying Aggregation Information Filters will return both the Filtered Parents and the base member records. The exception to this behavior is related to the when the Blend Unit Dimension uses the ;Member filter. The "Member" filter on a Blend Unit dimension will return only that member.

- **Not Used** - This will not include any parent level members in the results. The records will be collected at the row record level detail if included in the Data Source.

- **Parent Level Selection** - Selecting a single parent will limit the records to the members within that hierarchy and generate records for all members with data within the hierarchy. Selecting NA Clubs would create records for NA Clubs as TreeDescendantsInclusive.
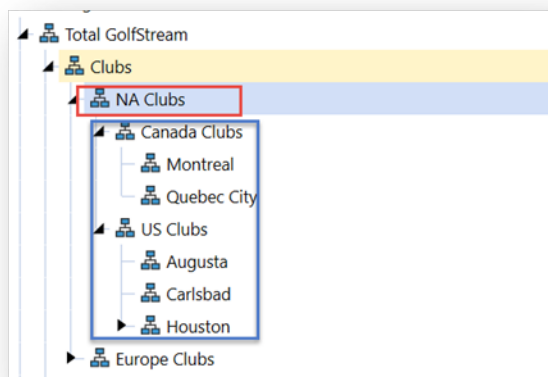
- **Parent Level Selection, as Member** – The dimension assigned as the Blend Unit can be restricted to return a summary parent member. This is a situation where the results are only required at an aggregated parent. By entering the setting as NA Club;Member, all the descendant's results will be aggregated to the parent, but the descendant members will not be included in the output. This option for a specified Parent Members is only available on the Blend Unit. For non-Blend Unit dimensions, using ;Member will return the Parent Member defined, as well as all the base members found within the parent member's hierarchy.
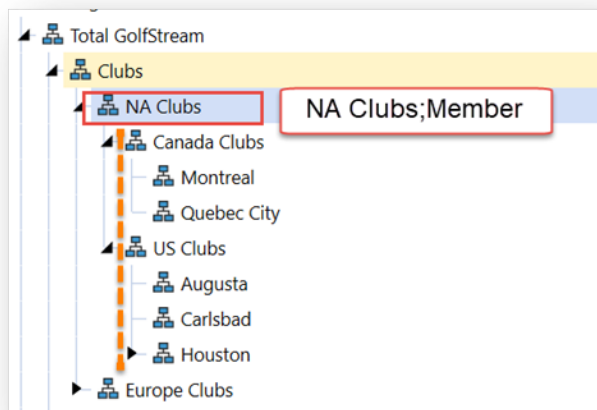
- **Parent Level Selection, Children** – Focuses aggregation points to a member's children. The selection of NA Clubs;Member.Children, property will return the Children of the Parent member as Children Inclusive. Alternatively, NA Clubs;Children can be used as a "non-inclusive" filter. Base members are also returned.

- **Parent Level Selection, TreeDescendants** – This is a non-inclusive option written as NA Clubs;TreeDescendants.

- **Star Schema Only** – When "Create Star Schema" is set to "True" the SSOnly filter can be used, NA Clubs;SSOnly. This setting will not generate parent level records in the BI-Blend data table. The "SSOnly" Aggregation Control will create a complete Star-Schema Dimension table containing fields MemberName, MemberDesc, NameAndDesc, ParentName, IndentLevel, IsBase and MemberSeq for Cube Dimension hierarchy as specified by the parent member in the Aggregation Control.

- **Star Schema Leveled Hierarchy** – The "SSLeveled" property, CostCenters;;;SSLeveled, requires "Create Star Schema" is set to "True". If enabled, the corresponding Star-Schema Dimension table will have zero-based column fields added, which correspond to the hierarchical structure of the dimension. Not valid for use on Account and Attributes. See the section on Leveled Hierarchy.

  > **NOTE:** Aggregation must be defined on the Blend Unit dimension. The Blend Unit Dimension requires the dimension hierarchy.

- **Reporting Labels as Name, Description or Name and Description** – By default, any Dimension set as an Aggregation will return the results using the Name field found in the Dimension properties. To modify the results, the Dimension or Attribute must be used as an Aggregation Control. Using the label properties on a standard, non-Star Schema, BI Blend table will replace the record with the labeling method. The delimited field for the label can be modified as:

  - N – Name

  - D – Description

  - ND – Name and Description

- Cube Dimension Aggregation Control Syntax

  - TopMember;RestrictMember;Labels;StarSchemaControl

    Houston;;ND

    Houston;Member;ND

    Houston;Member.Children;D

    Houston;;;SSLeveled

Attribute Dimension Syntax, the number of fields is larger because to the inclusion of the Dimension Name.

- ○ Dimension Name; TopMember;RestrictMember;Labels

  UD8BlendAttributes;Contracts;;D

  UD8BlendAttributes;Contracts;Member;D

Leveled Hierarchy

Leveled hierarchy is for aggregation reporting where parent values reflect the aggregation points. It is activated as a BI-Blend Aggregation Control impacting the Star-Schema Dimension tables with the creation of Leveled and IsBaseBIBlend column fields. Leveling adds the hierarchy context that is useful in Pivot Grid and custom Dashboard reports.

Syntax

- TopMember;RestrictMember;Labels;SSLeveled

- Clubs;;;SSLeveled

The column fields generated by the "SSLeveled" property will be created to the maximum depth of the hierarchy, starting from the defined parent to the base member. The leveled columns will only be created on base members and their ancestors where data is populated in the BI-Blend data table. Levels greater than those containing data will not be created.   The property is not valid for use on Account and Attribute Aggregation Controls.

The leveling process will generate two field placeholders.

- **XFLeveled** - Created only on base-level records for data intersections whose hierarchy is less than the maximum depth

- **XFStored** – Created only on parent level members. This represents the data intersection of stored-parent values created by the BI-Blend engine

Leveled Hierarchy Processing

The "SSLeveled" property enables the generation of the hierarchy leveling fields. Efficiency is accomplished by limiting the leveling only to members that contain data in the BI-Blend data table. However, leveling is an additional process used by the BI-Blend Engine which will affect the overall BI-Blend processing time.

Star-Schema Leveling Column Fields

The "SSLeveled" property enabled generates new columns in corresponding Star-Schema Dimension Tables. BiBlend will ONLY aggregate the ultimate base members of an extended dimension. Loading a member that becomes a parent as a result of dimension inheritance will result in NO aggregation for the extend parent.

- IsBIBlendBase: Reflects the hierarchy status of where the actual data records exist. Restricts the creation of leveling only on members that contain data in the data table. Designates the member as a base (1) or parent member (0) by the usage in the BI-Blend data table.

- Level x: Zero-based to maximum descendent depth only on member where BI Blend data exists.

> **NOTE:** The BI Blend leveled hierarchy is only available for star schema-enabled workflows.

Example: Using the Leveled Hierarchy in the Large Data Pivot Grid

In the example below, the table has a hierarchical view.

Notice in this example the items labeled "XFStored" indicate that the table is being "leveled" to create leveled columns based on the maximum depth of hierarchy from the BI-Blend data.

Leveled Large Pivot Grid				
				◢ CoreRegions
				◢ ServiceSales
				USGeography
◢ Clubs	◢ Irons	◢ Hybrids	◢ Extended_Hybrid1	XFLeveled	1,000.00
			◢ Extended_Hybrid2	XFLeveled	1,000.00
			◢ XFStored	XFStored	2,000.00
		◢ Wedges	◢ Extended_Wedges1	XFLeveled	1,000.00
			◢ Extended_Wedges2	XFLeveled	1,000.00
			◢ XFStored	XFStored	2,000.00
		◢ XFStored	◢ XFStored	XFStored	4,000.00
	◢ Putters	◢ Belly	◢ Extended_Belly1	XFLeveled	1,000.00
			◢ Extended_Belly2	XFLeveled	1,000.00
			◢ XFStored	XFStored	2,000.00
		◢ Conventional	◢ Extended_Conventi...	Extended_Conv...	1,000.00
				Extended_Conv...	1,000.00
				XFStored	2,000.00
			◢ XFStored	XFStored	2,000.00
		◢ Long	◢ Extended_Long1	XFLeveled	1,000.00
			◢ Extended_Long2	XFLeveled	1,000.00
			◢ XFStored	XFStored	2,000.00

Setting IsBIBlendBase

Use the IsBaseBIBlend parameter to show where actual data records exist in the table.

1. From the Leveled Large Pivot Grid, right-click in the blue row and select Show Prefilter.
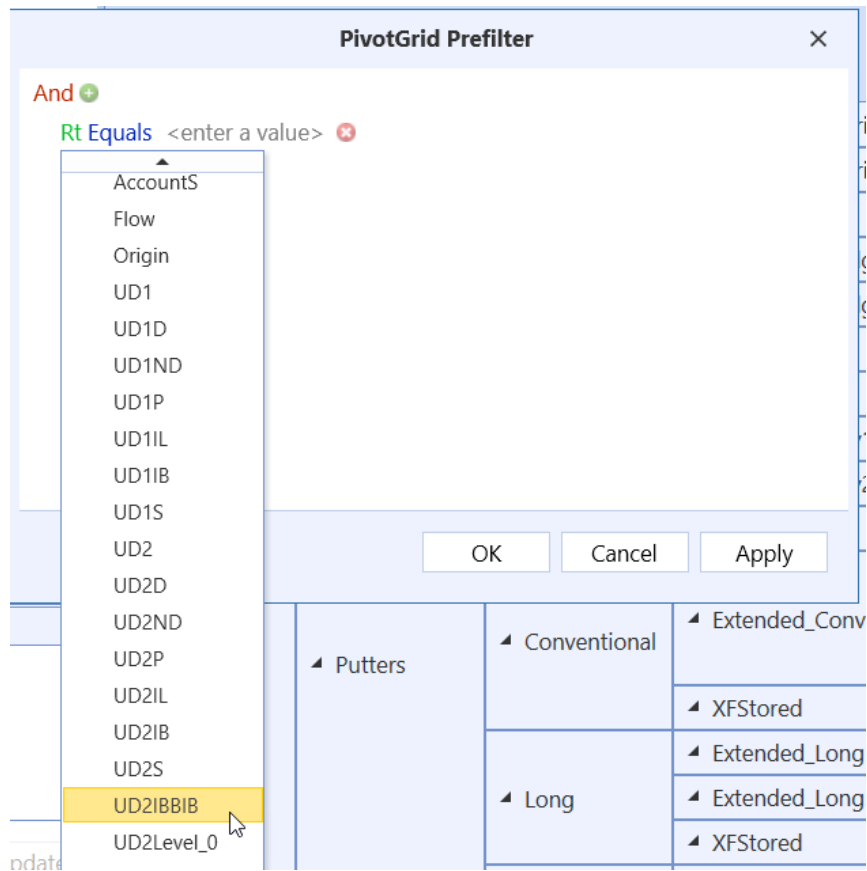
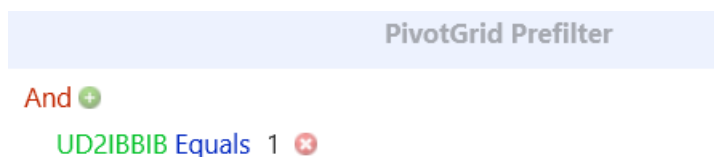The PivotGrid Prefilter dialog box opens.
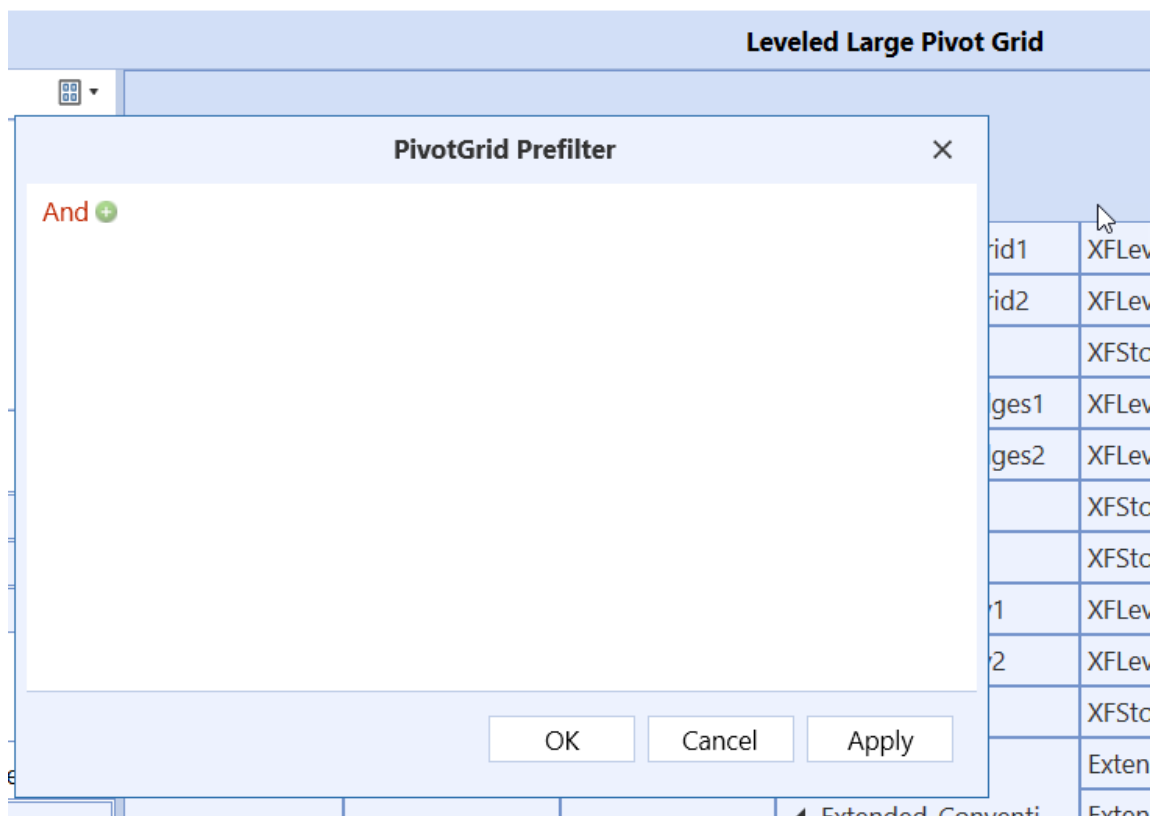
2. Click the green +.
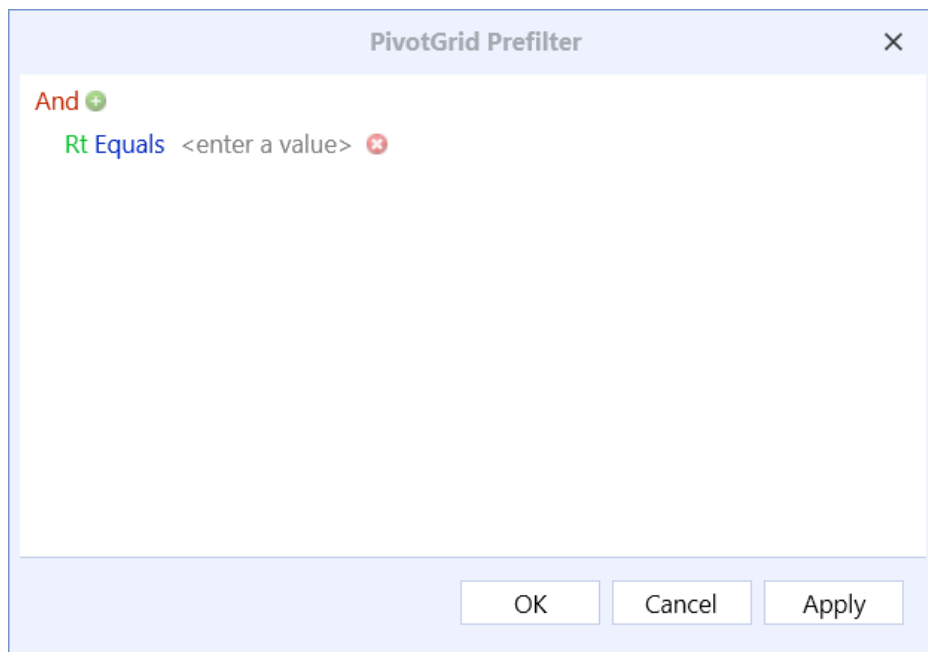
3. Click Rt and scroll through the list to select UD2IBBIB.

4. Click <enter a value> and type 1.

5. The equation looks like this:

6. Click **OK**.

7. After applying IsBaseBIBlend, the leveled table now changes to show only those rows that contain actual data (XFLeveled) and has removed the rows that were added for leveling (XFStored).

8. Click **Save** to save your changes.

Using Translation

Translation can be defined as any destination currency. Only the direct method is used, based on the Rates defined on the Cube Settings, determined by Account Type. No complex currency translation is supported, by sub-parent levels. All Entities will be translated to the destination currency.

Translation functionality requires Translate to be set, with rates input. Entity and Account must have an member defined for Aggregation Info to determine the Local currency and Rates to use.

BI Blend Application Setup

Creating a Target SQL External Database

The BI Blend target table must be a separate external database, that is not in the Application Database. You will update the Application Server Configuration Utility to confirm a connection to the new BI Blend target database.

2. In SQL Studio create the database, specifying SQL Server driver as the driver. You cannot use SQL OLE.

3. Launch the Server Configuration Application and perform these steps:

   a. Open **appserverconfig.xm**l.

   b. In **Databases**, add the target database to Collection and apply the appropriate settings. Set the Database as **Is External Database = True**.

   c. Ensure that Access Group for Ancillary Tables assigns a group that contains all users requiring access to BI Blend tables from product tool such as Workflow, Workspace
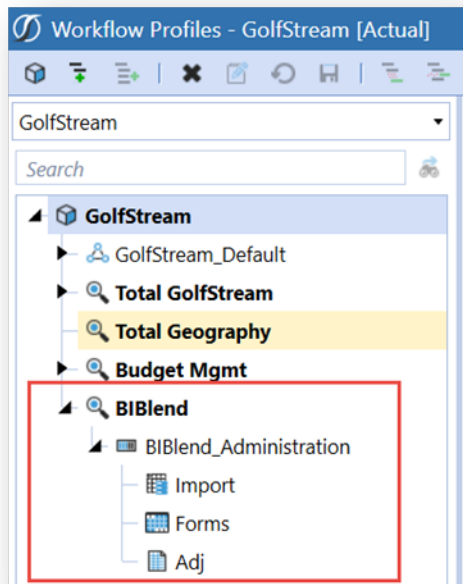
or Dashboards.
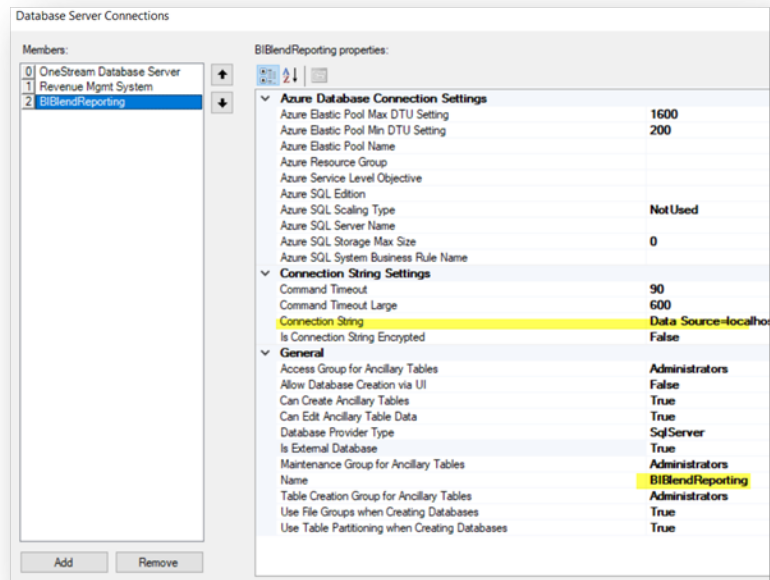
4. Close and Save the file changes

Set Up Workflow for BI Blend

1. Create a Workflow structure to manage BIBlend. Apply the appropriate Security groups as required. (Administrator)

2. Select the Import channel to Blend, by Scenario Type

    a. **Blend**: Limited to the BI Blend data processing

    b. **Blend – Workspace**: Additional support to display Workspace Dashboards

3. If designing BI Blend as a Workflow Process, Dashboards can be assigned through the Validate Profile.

4. Assign the DataSource and Transformation Rule Profile which BI Blend will utilize to derive the metadata and subtotals in the resulting table

5. Set the BI Blend Parameters which is a collection of properties which define the BI Blend process defined in this document.

Fdx Specialty Connector BRAPI's

FDX, Fast Data Extract, BRAPIs allow a variety of options for connecting to DataSources for BI Blend. A key differentiator between the FDX BRAPI's and other collection methods is support of parallel processing, in memory processing and management of the Time dimension.

FDX BRAPI's provide functionality to build Connectors to extract data from:

*   **Cube Views**: Extract data through a Cube View definition. Ideal for defining data definitions through a Cube View, including Dynamic Calc results.

- **Across Cube Data Units**: Extract Cube data to a BI Blend target table through defined Data Unit filters.

- **Stage Workflow Imports**: Ability to leverage existing Stage Data.   Uses may be reporting on existing "attribute" records contained in Stage, or simply enhanced dashboard reporting on Stage data.

- **Source Systems / Data Warehouses**: Performance oriented solution to connect to source system.

Performance is gained through the BRAPI's ability to parallel process. For example, extracting data by Cube Data Unit will parallel process all the Data Units defined in the filter. Second, the FDX BRAPI's do not generate a ".CSV" file as do Data Management File "Export Data" or "Export File" processes. The results of the export are managed during the BI Blend "in-memory" processing.

In cases of very large data sets, which where multiple periods are loaded, the processing time can be slow because each period is reflected as a data record. FDX BRAPI's offer solutions to pivot the Time records to columns in order to create a matrix data layout. The Datasource can associate each of the periods with an "Attribute Value" field within the Integration settings. The design will treat each record as a collection of 12 periods when processing.

- **FdxExecuteCubeView**: Extract data defined through a Cube View. Any data presented in the Cube View will be extracted, such as Dynamic Calculated results.

- **FdxExecuteCubeViewTimePivot**: Cube View Data will generate all time as Columns which can be assigned as Attribute Value members in the Data Source.

- **FdxExecuteDataUnit**: Cube Data extract solution to extract data from Data Unit members.

- **FdxExecuteDataUnitTimePivot**: Cube Data extract solution to extract data from Data Unit members. Generate all time as Columns, which can be assigned as Attribute Value members in the Data Source.

- **FdxExecuteStageTargetTimePivot**: Extract existing Workflow's Stage Data. Generate all time as Columns, which can be assigned as Attribute Value members in the Data Source.

- **FdxExecuteWarehouseTimePivot**: Extract data from an external source system.

- **FdxGetCubeViewOrDataUnitColumnList**: Connector BRAPI used to return field names.

- **FdxGetStageTargetColumnList**: Connector BRAPI used to return field names.

- **FdxGetWarehouseColumnList**: Connector BRAPI used to return field names.

Data Unit Example:

```
Public Function Main(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Transformer, ByVal args As ConnectorArgs) As Object
    Try

        Select Case args.ActionType
            Case Is = ConnectorActionTypes.GetFieldList
                'Return Field Name List
                Dim timeMemFilter As String = "T#2018.Base"
                Dim isTimePivot As Boolean = True
                Dim useGenericTimeColNames As Boolean = True
                Return BRApi.Import.Data.FdxGetCubeViewOrDataUnitColumnList(si, timeMemFilter, isTimePivot, useGenericTimeColNames)

            Case Is = ConnectorActionTypes.GetData
                'Parameter Format for WorkflowProfile Text4
                'EntityMF=[E#[Total GolfStream].Base],ConsName=[Local],ScenarioTypeName=[Actual],ScenarioMF=[S#Actual],TimeMF=[T#2017.Base],ViewName=[YTD],SuppressNoData=[0],Filter=[Account Like '1*' or Account Like '2*' or Account Like '3*' or Account Like '4*' or Acco

                'Get Parameters Stored in Text4
                Dim nvb As New NameValueFormatBuilder(api.WorkflowProfile.GetAttributeValue(api.ScenarioTypeID, SharedConstants.WorkflowProfileAttributeIndexes.Text4))
                Dim cubeName As String = api.CubeName
                Dim entityMemFilter As String = nvb.NameValuePairs.XFGetValue("EntityMF", String.Empty)
                Dim parentName As String = nvb.NameValuePairs.XFGetValue("ParentName", String.Empty)
                Dim consName As String = nvb.NameValuePairs.XFGetValue("ConsName", String.Empty)
                Dim scenarioTypeName As String = nvb.NameValuePairs.XFGetValue("ScenarioTypeName", ScenarioType.Actual.Name)
                Dim scenarioTypeId As Integer = ScenarioType.GetItem(scenarioTypeName).Id
                Dim scenarioMemFilter As String = nvb.NameValuePairs.XFGetValue("ScenarioMF", String.Empty)
                Dim timeMemFilter As String = nvb.NameValuePairs.XFGetValue("TimeMF", String.Empty)
                Dim viewName As String = nvb.NameValuePairs.XFGetValue("ViewName", String.Empty)
                Dim suppressNoData As Boolean = ConvertHelper.ToBoolean(nvb.NameValuePairs.XFGetValue("SuppressNoData", "1"))
                Dim useGenericTimeColNames As Boolean = True
                Dim filter As String = nvb.NameValuePairs.XFGetValue("Filter", String.Empty)
                Dim parallelQueryCount As Integer = 8
                Dim logStatistics As Boolean = False

                'Process Data
                Dim dt As DataTable = BRApi.Import.Data.FdxExecuteDataUnitTimePivot(si, cubeName, entityMemFilter, consName, scenarioTypeId, scenarioMemFilter, timeMemFilter, viewName, suppressNoData, useGenericTimeColNames, filter, parallelQueryCount, logStatistics)
                If Not dt Is Nothing Then
                    api.Parser.ProcessDataTable(si, dt, True, api.ProcessInfo)
                Else
                    BRApi.ErrorLog.LogMessage(si, "FDX DU Params", nvb.GetFormatString())
                End If

                Return Nothing
```

Using Fdx Connectors

Workflow Connectors are required to realize the benefits of the parallel processing and Time Pivot capabilities of the FDX BRAPI's. Once created, the Connectors are assigned to the BI Blend Workflow. Each type of Connector is defined using "name value pairs" on the properties of the Workflow Text fields.

| BI-Blend Parameters	
⊟ Substitution Text Settings	
Text 1	
Text 2	
Text 3	
Text 4	CubeViewName=[FDX_TrendTrailingSales],EntityDimName=[CorpEntities],EntityMF=[E#[Total GolfStream].Base],

Server Roles

Optional Application Server

The default processing will be BI Blend processed are queued across the available Stage Servers. Within the Workflow BI Blend Settings, a defined server can be established for all BI Blend processes. This can be as a list and support wild cards.

Batch Processing Support

Batch Processing is supported through automating the BI Blend process.

Common Error Messaging

Cannot find a matching base member in the data record, no data intersectons are found.
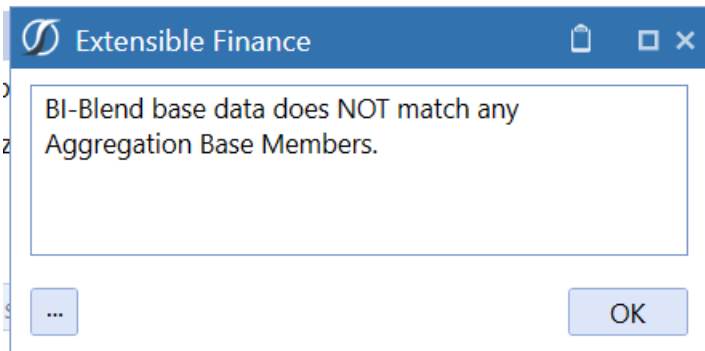
Reporting Solutions

Dashboard Adapter – BIBlendInfo Method Query

BIBlendInfo Method Query is available to retrieve the relative information from the StageBIBlendInformation Table.

Large Pivot Grid

The Large Pivot Grid is a Dashboard Component which is designed to allow pivot style dashboard reporting and analysis on external database tables. Key features of this Component are:

- Server based processing – Key solution for accessing data in large tables. Pivoting requests are performed on the server, returning only the requested "slice" of data.

- Supports paging to manage large data sets

- Measures support only a single aggregation type, (Sum, Min, Max)

Component Properties	
General (Component)	
Name	BIBLargePivGrid_Att
Description	
Maintenance Unit	BIBlend
Component Type	Large Data Pivot Grid (Windows App Only)
Large Data Pivot Grid	
Show Toggle Size Button	True
Database Location	External
External Database Connection	BIBlendReporting
Table Name	BIB_|AppName|_|WFText1|_|WFScenario|_|WFTime|
Row Fields	Account
Column Fields	Entity
Data Fields	|WFTime|
Data Field Aggregation Types	
Excluded Fields	Flow,Origin,IC,UD1,View,SourceID,TextValue,Rt,Label,Cons,Scenario,Account Type,UD5
Page Size	500

BI Blend Data Adapter

The BI Blend Adapter is used to simplify writing queries to the BI Blend tables by eliminating the need for SQL scripting. As an Adapter, it cannot support the complete contents of very large BI Blend tables. The BI Blend Adapter should contain "where" clauses to slice the results into reporting slices. The BI Blend Adapter does not support paging to manage large volumes of records.

General (Data Adapter)	
Name	BIBlend_DataAdapter
Description	BI Blend Adapter
Maintenance Unit	BIBlend
Data Source	
Command Type	BI-Blend
Results Table Name	BIBLEND1
Table Info	BIB_|AppName|_|WFText1|_|WFScenario|_|WFTime|
Group By	Entity,HoustonProducts,HoustonCustomers
Data Field Aggregation Types	AggType1 = [2018M1, Sum],AggType2 = [2018M1, Count]
Where Clause	HoustonCustomers= 'shanks'

BI Blend Derivative Rules

This is a new class of Derivative rules specifically designed for use with the external database tables generated from the BI Blend Workflow.

- BlendUnit All

- BlendUnit Base

- BlendUnit Parent