



# Design and Reference Guide

---

8.0.1 Release

Copyright © 2023 OneStream Software LLC. All rights reserved.

Any warranty with respect to the software or its functionality will be expressly given in the Subscription License Agreement or Software License and Services Agreement between OneStream and the warrantee. This document does not itself constitute a representation or warranty with respect to the software or any related matter.

OneStream Software, OneStream, Extensible Dimensionality and the OneStream logo are trademarks of OneStream Software LLC in the United States and other countries. Microsoft, Microsoft Azure, Microsoft Office, Windows, Windows Server, Excel, .NET Framework, Internet Information Services, Windows Communication Foundation and SQL Server are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. DevExpress is a registered trademark of Developer Express, Inc. Cisco is a registered trademark of Cisco Systems, Inc. Intel is a trademark of Intel Corporation. AMD64 is a trademark of Advanced Micro Devices, Inc. Other names may be trademarks of their respective owners.

# Table of Contents

Introduction .....	1
About the Financial Model .....	2
Extensible Dimensionality .....	2
Cube Design Choices within an Application .....	5
Application Model Design Considerations .....	5
Standard Cube Design .....	6
Standard Large Sparse Cube Design .....	7
Hybrid Cube Design .....	8
BI Blend .....	10
Key Elements of BI Blend .....	10
Relational Blending .....	10
Relational Blending Types .....	11
Model Blending .....	11
Relational Blending API .....	13
Cache Level Types .....	14
Relational Model Blending API Functions .....	15
Formulas .....	25
OneStream API Details and Database Documentation .....	25
Formula Structure .....	26

## Table of Contents

---

Applying Formulas .....	28
Formula Types .....	29
Formula Execution .....	30
Writing Formulas .....	33
Writing Stored Calculations .....	40
Avoiding Data Explosion .....	52
Key Functions .....	55
Account .....	56
Consolidation .....	57
Data .....	59
Entity .....	64
Flow .....	64
FX .....	65
Journals .....	68
Member .....	68
Scenario .....	70
Status .....	71
Time .....	72
Functions .....	72
Examples of Key Functions in Use .....	73
Finance Function Types .....	83

## Table of Contents

---

Create Two Accounts .....	87
Consolidation .....	88
Example of a Consolidation .....	90
Launching a Consolidation .....	91
Calculation Status .....	93
Currency Translation .....	95
How the Origin Dimension Works with the Consolidation Dimension .....	96
Intercompany Elimination .....	98
Direct and Indirect Eliminations .....	100
Equity Pickup .....	102
Entity Aggregation .....	103
How Entity Aggregation Works .....	103
Example of an Entity Aggregation .....	104
Launching an Aggregation .....	107
Entity Aggregation Input .....	108
Working With Hybrid Scenarios .....	109
Data Bindings .....	110
Data Filters .....	111
Share Data From the Source Scenario .....	111
Copy Data From the Source Scenario .....	112

## Table of Contents

---

Navigation .....	113
About the OneStream Windows Application .....	113
Launching the OneStream Windows App .....	114
Defining Server Connections .....	114
Logging In .....	115
OnePlace Layout .....	119
Point of View .....	125
Navigating Mobile .....	128
Mobile Toolbar .....	128
Collecting Data .....	132
Data Sources .....	132
Delimited Data Source .....	132
Connector Data Source .....	132
Loading Cell Text as an Annotation Data Attachment .....	153
Forms .....	154
Forms Channel Workflow .....	154
Form Allocations .....	155
Applying Literal Value Parameters to Form Templates .....	160
Loading Data through Excel Templates or CSV .....	162
Loading Stage Data .....	163

## Table of Contents

---

Loading Form Data .....	165
Loading Journal Data .....	171
Loading Cell Detail .....	178
Cell Detail CSV Template .....	181
Exporting Data Attachments .....	183
Loading Excel Templates to Custom Tables .....	183
Filter Runtime Data and Reduce Maintenance with Parameters .....	187
About Parameters .....	187
Parameter Types .....	188
Literal Value Parameters .....	189
Input Value Parameters .....	189
Delimited List Parameters .....	190
Bound List Parameters .....	191
Member List Parameters .....	192
Member Dialog Parameters .....	192
Benefits .....	192
Ways to Use Parameters .....	193
Cube Views .....	193
Data Adapters and Dashboard Components .....	194
Extensible Documents .....	194

## Table of Contents

---

Books .....	195
Requirements .....	195
Definition and Assignment .....	195
Syntax .....	195
Best Practices .....	196
Design .....	196
Naming .....	197
About Parameter Definitions .....	198
Property Types .....	198
Create a Parameter .....	198
Specify Literal Value Settings .....	200
Specify Input Value Parameter Settings .....	201
Specify Delimited List Settings .....	201
Specify Bound List Settings .....	202
Sample Method-Based Parameter .....	205
Sample SQL-Based Parameter .....	205
Specify Member List Settings .....	206
Specify Member Dialog Settings .....	207
Sample Delimited List Parameter in a Cube View .....	209
Manage Parameters .....	211
Assign a Parameter .....	212

---

## Table of Contents

---

Assign to a Report Subheader .....	212
Assign to a Cube View Column Header .....	214
Modify a Parameter .....	215
Copy a Parameter .....	216
Delete or Remove a Parameter .....	216
Rename a Parameter .....	217
Identify Parameter Assignments .....	218
Nest Parameters .....	219
Presenting Data With Extensible Documents .....	222
Extensible Document Framework .....	222
About Creating Extensible Documents .....	223
Creating a Document in Microsoft Word .....	226
Creating a Document in Microsoft PowerPoint .....	237
Creating a Document in Microsoft Excel .....	239
Using Extensible Documents in OneStream .....	243
Implementing Security .....	252
Application Security .....	252
Security Best Practices .....	253
Import .....	256
Forms .....	256

## Table of Contents

---

Adjustments .....	257
Entity Security .....	257
Relationship Security .....	257
Change the Relationship Security .....	258
Security Configurations .....	262
System Security .....	265
Managing a OneStream Environment .....	265
Disable Inactive Users .....	266
Set the Inactivity Threshold .....	267
Review Settings .....	270
Business Rules .....	272
System Table .....	272
Business Rules .....	272
Platform Engines .....	273
BRApi .....	276
Business Rule Classifications .....	276
Business Rule Organization and Referencing .....	284
Cube View Extender: Advanced Cube View Formatting .....	289
Common Cube View Extender Business Rule Examples .....	295
Dashboard Extender Business Rule Examples .....	303
Dashboard DataSet Business Rule Examples .....	304

## Table of Contents

---

System Extender Business Rules .....	307
Direct Load Business Rules .....	307
Data Management Automation With PowerShell .....	308
Instantiation of Standard .NET Classes .....	309
Using PowerShell Script Editor .....	309
Configuring PowerShell for the Client API .....	309
Running Unsigned Scripts .....	310
Configuration for .Net Framework v.4.0 .....	310
Learning PowerShell .....	310
Using the Client API in a PowerShell Script .....	311
Exposing Data Management Automation through OneStream Web API .....	313
OneStream Web API endpoints: .....	314
Extracting and Loading Dimensions .....	315
Extracting Dimension Changes .....	315
Loading the XML File .....	319
Common Load Errors and Resolutions .....	320
Project Extract and Load .....	321
Project File .....	321
File Extract .....	323
Zip Extract .....	325

## Table of Contents

---

File Load .xfProj .....	326
Zip Load .....	327
Cubes .....	328
Dimensions .....	328
Restricted Characters .....	330
Reserved Words .....	331
Dimension Library Toolbar and Right-Click Options .....	333
Dimension Member Right-Click Options .....	335
Dimension Grid View .....	336
Entity Dimension .....	337
Scenario Dimension .....	344
Account Dimension .....	355
Flow Dimension .....	361
User Defined Dimensions 1-8 .....	363
Consolidation Dimension .....	371
Time Dimension .....	372
View Dimension .....	374
Origin Dimension .....	376
Intercompany (IC) Dimension .....	377
Time Profiles .....	377

## Table of Contents

---

Profile .....	377
Time Periods .....	378
Cubes .....	380
Cube Properties .....	380
Cube Dimensions .....	385
Cube Reference .....	385
Data Access .....	386
Data Cell Conditional Input .....	391
Data Management Access Security .....	391
Integration .....	392
Foreign Exchange Rates .....	393
Grid Settings Intersection .....	393
Lock FX Rates .....	395
System Changes: Administrator Access to Lock FX Rates .....	396
Lock and Unlock Rates .....	397
BRAPI Rule Functions .....	401
Spreadsheet or Excel Functions .....	401
Member Filters .....	402
Member Script Abbreviations and Examples .....	403
Member Script Constants .....	405
Member Expansion Functions .....	405

## Table of Contents

---

Reverse Order Member Expansions .....	415
Workflow Member Expansions .....	419
Member Expansion Where Clause .....	420
Time Functions and Workflow References .....	423
Time Functions and Global References .....	424
Time Functions and General References .....	425
Substitution Variables .....	428
Workflow Substitution Variables .....	430
Cube View Substitution Variables .....	433
General Substitution Variables .....	435
Member Filter Substitution Variables .....	436
Custom Substitution Variables .....	437
Commonly Used Member Filter Functions .....	437
Member Filter Builder .....	442
Workflow .....	444
Workflow .....	445
Workflow .....	445
Workflow Profiles .....	448
Data Units .....	476
Workflow Stage Import Methods .....	486

## Table of Contents

---

Workflow Channels .....	499
Confirmation Rules .....	505
Certification Questions .....	508
Data Collection .....	511
Data Sources .....	511
Data Source Properties .....	511
Fixed Files .....	512
Delimited Files .....	513
Connectors .....	513
Data Management Export Sequences .....	514
Source Dimensions .....	514
Source Dimension Properties .....	514
Matrix Settings .....	517
Numeric Settings .....	518
Bypass Settings .....	519
Stored Text Settings .....	520
Using Excel as a Data Source .....	521
Transformation Rules .....	521
Transformation Rules Toolbar .....	521
General Settings .....	522

## Table of Contents

---

Mapping Types .....	523
Derivative .....	525
Logical Operator .....	532
Derivative Type .....	533
Order .....	534
Lookup .....	534
Transformation Rule Profiles .....	535
Form Templates .....	535
Form Template Group Properties .....	535
Security .....	535
Form Template Properties .....	536
Form Template Profiles .....	540
Managing Form Templates .....	541
Journal Templates .....	543
Journal Template Group Properties .....	543
Security .....	543
Journal Template Properties .....	543
Long Running Requests .....	546
Presenting Data With Books, Cube Views and Dashboards .....	547
Presenting Data Using Report Books .....	547

## Table of Contents

---

Book Designer Toolbar .....	548
Book Properties .....	549
File Type .....	550
Excel Export Item .....	551
Report .....	551
Loop .....	552
Loop Variables .....	553
Change Parameters .....	554
If Statement .....	556
Else/Else If Statement .....	557
Create a Report Book .....	557
Book Preview Tool Bar .....	559
Build Reports Through Cube Views .....	561
Build a Simple Cube View .....	561
Cube View Shortcuts .....	570
Member Filter Builder .....	570
Cube View Performance .....	577
Advanced Cube Views .....	583
Format a Cube View .....	585
Calculations .....	636
Workspaces .....	666

---

## Table of Contents

---

Find and Assign Values Using Object Lookup .....	720
Cube Views Example .....	722
Extensible Document Settings .....	724
Options .....	725
XFCCell .....	726
Use Business Rules in Extensible Documents .....	726
Application Tools .....	727
Application Security Roles .....	727
Application Properties .....	731
General Properties .....	731
Dimension Properties .....	734
Standard Reports .....	735
Header Labels .....	736
Header Bar .....	736
Footer .....	737
Business Rules .....	737
Business Rule Search .....	743
Business Rule Properties .....	746
Business Rule Types .....	748
Parser Rule .....	750

## Table of Contents

---

Connector Rule .....	752
Conditional Rule .....	754
Derivative Rule .....	756
Cube View Extender .....	758
Dashboard DataSet .....	759
Dashboard Extender .....	759
Dashboard XFBRString .....	760
Extensibility Rule .....	760
Extender .....	764
Journals Event Handler .....	766
Save Data Event Handler .....	766
Forms Event Handler .....	766
Data Quality Event Handler .....	767
Data Management Event Handler .....	768
Workflow Event Handler .....	768
Client Updater .....	768
Data Management .....	770
Search .....	770
Data Management .....	770
Sequences .....	771
Steps .....	773

## Table of Contents

---

Execute Scenario Hybrid Source Data Copy .....	774
Clear Data .....	774
Copy Data .....	775
Custom Calculate .....	777
Execute Business Rule .....	779
Export Data .....	779
Export File .....	781
File Source Type .....	782
Export Report .....	783
Reset Scenario .....	784
Run a Sequence or Step .....	785
Spreadsheet .....	785
Microsoft Excel 2016 Chart Types .....	788
Retrieve Functions .....	793
File, Open and File, Save As Functionality .....	793
Reduce Cell Text Size .....	794
OneStream Ribbon .....	795
Task Scheduler .....	795
Grid View .....	796
Task Scheduler Details .....	797
Load .....	809

---

## Table of Contents

---

Extract .....	810
Create a New Task .....	813
User Authentication and Single Sign-On .....	845
How Users are Configured for Authentication .....	846
About Managing Users and Groups .....	847
Ways to add Users and Groups to an Application .....	847
Creating and Managing Users .....	847
Requirements .....	848
Tips and Best Practices .....	849
Creating Users .....	849
Managing Users .....	852
Loading and Extracting Users .....	854
Creating and Managing Groups .....	856
Requirements .....	856
About Groups and Inherited Security .....	857
About Exclusion Groups .....	858
Tips and Best Practices .....	858
Creating Groups .....	858
Creating Exclusion Groups .....	859
Managing Groups .....	860

## Table of Contents

---

Managing Group Membership .....	861
Loading and Extracting Groups .....	861
System Security Roles .....	864
Application Server Configurations .....	864
System Roles .....	865
System Configurations .....	865
General System Configurations .....	866
Environment System Configurations .....	868
Recycling System Configurations .....	872
Database System Configurations .....	873
Audits .....	875
Manage System Security .....	877
Access System Security Roles .....	878
Manage System Security Users .....	879
Manage System Security Groups .....	879
Manage System Security Roles .....	880
Load and Extract .....	880
BRApi .....	881
Combined Roles .....	881
File Share Security Roles .....	881

## Table of Contents

---

System User Interface Roles .....	882
System Security Users and Groups .....	884
Managing Users and Groups Using BRApi Functions .....	884
Examples .....	885
Create a User .....	886
Create a Group .....	886
International Settings .....	886
Specifying Culture Settings on the Client Machine Operating System .....	887
Specifying Server Side User Culture Settings .....	887
Working With System Applications .....	888
System Dashboards .....	889
Logging .....	890
Logon Activity .....	890
Task Activity .....	891
Error Logs .....	893
System Tools .....	894
System Business Rules .....	894
Sample System Business Rule .....	895
Database .....	895
Environment .....	896

---

## Table of Contents

---

Monitoring Environments .....	896
Specifying General Settings .....	897
Specifying Monitoring Settings .....	898
Specifying Metrics Settings .....	898
Web Servers .....	900
Configuration .....	900
Audit .....	902
Mobile Web Servers .....	903
WebTo App Server Connections .....	905
Working With Application Server Sets .....	906
Accessing Server Sets Behaviors .....	907
Server Sets Configuration .....	908
Using Server Sets Audit .....	910
Monitoring Server Sets Performance .....	910
Application Server Behavior .....	911
Application Server Configurations .....	912
Application Server Hardware .....	912
Application Server Audit .....	913
Application Server Performance .....	914
Database Servers (Connection Items) .....	915
Behaviors .....	915

---

## Table of Contents

---

Configuration .....	916
Hardware .....	916
Audit .....	917
Performance .....	918
Diagnostics .....	919
OneStream Database Servers (Schema Items) .....	921
Configuration .....	921
Audit .....	922
Diagnostic .....	923
Azure Configurations .....	923
Azure Subscription Settings .....	923
Environment Monitoring .....	924
Task Load Balancing .....	927
Database Server Connection .....	928
Azure Server Sets Settings .....	929
File Explorer .....	930
File Share .....	930
Permissions .....	931
Content Folder .....	931
Uploads .....	932
Supported File Sizes .....	933

---

## Table of Contents

---

Whitelist File Extensions .....	933
Overview .....	933
Whitelist Files Extensions Prerequisites .....	936
Define the List of File Extensions .....	936
Add Documents using File Explorer .....	939
Troubleshooting .....	940
Conclusion .....	940
Application Database/System Database .....	941
Documents .....	941
Load/Extract System Artifacts .....	941
Time Dimensions .....	943
Using OnePlace Workflow .....	945
Workflow Tasks .....	948
Import .....	948
Right-Click Options .....	950
Validate .....	950
Load .....	951
Pre-Process .....	952
Input Forms .....	952
Journals Toolbar .....	955

## Table of Contents

---

Input Journals .....	955
Process .....	957
Confirm .....	958
Certify .....	958
Multi-Period Processing .....	959
Analysis Pane .....	960
Cube Views and Dashboards for Analysis .....	960
Intercompany Matching .....	960
Using OnePlace Cube Views .....	962
Toolbar .....	963
Shortcuts, Scaling, and HotKeys .....	965
Shortcuts .....	965
Scaling .....	966
HotKeys .....	967
Right-Click Options .....	967
Spreading Type .....	968
Spreading Properties .....	970
Allocation .....	973
Data Attachments for Selected Cell/Data Unit .....	981
Spell Checking in Data Attachments .....	982

## Table of Contents

---

Cell Detail .....	985
Cell POV Information .....	987
Cell Status .....	990
Data Unit Statistics .....	991
Drill Down .....	992
Show Cube View as a Report .....	995
Using OnePlace Dashboards .....	997
Dashboard Toolbar .....	998
Printing Options .....	998
Storing Documents .....	999
About Excel Add-In .....	1001
Get Started .....	1002
Add the OneStream Ribbon .....	1002
Log Into the Excel Add-In .....	1003
Update the Excel Add-In .....	1005
Excel Add-in Task Pane .....	1006
Quick Views .....	1007
Manage Quick Views .....	1007
Create and Modify Quick Views .....	1009
Cube Views in Spreadsheet .....	1020

## Table of Contents

---

Define Cube View Connections .....	1020
Configure Cube View Connections .....	1020
Format Cube View Connections .....	1025
XF Retrieves .....	1029
Absolute Cell References .....	1031
Relative Cell References .....	1031
Table Views .....	1031
Table Views Spreadsheet Control .....	1032
Navigating the Excel Add-In .....	1034
OneStream Ribbon .....	1034
Logon .....	1034
Data .....	1034
Calculation .....	1035
Analysis .....	1036
Quick Views .....	1036
Cube Views .....	1036
Retain Formulas in Cube View Content in Excel and Spreadsheet .....	1038
Dynamically Highlight Evaluated Cells in Excel or Spreadsheet .....	1039
Using Retain Formulas and Dynamically Highlight Evaluated Cells .....	1042

## Table of Contents

---

File Explorer .....	1048
General .....	1050
Spreading .....	1051
Spreading Types .....	1051
Administration .....	1052
Display Context Pane .....	1052
Preferences .....	1052
Excel Calculation .....	1053
Right-Click Options .....	1054
Retain Formatting in Cube Views with Selection Styles and Conditional Formatting .....	1059
Using Selection Styles .....	1059
Creating a Selection Style .....	1059
Using an Existing Style .....	1062
Adding a Selection Styles Shortcut .....	1063
Reviewing Styles and Ranges .....	1063
Using Right-click Menu Options .....	1065
Modifying and Duplicate Styles .....	1065
Merging Styles .....	1066
Conditional Formatting .....	1067

## Table of Contents

---

Create Conditional Formatting in an Existing or New Cube View .....	1068
Styles .....	1071
Creating a Custom Style .....	1071
Creating Excel Forms and Reports .....	1078
Named Regions .....	1078
Retrieve Functions .....	1079
Visual Basic for Applications (VBA) Procedures .....	1082
Error Logs .....	1086

# Introduction

The purpose of this guide is to provide best practices, tips, in-depth explanations of key capabilities, and configurations for implementers, administrators, and end users. Use this guide to enhance your understanding of applications, from specific functions to comprehensive reporting.

# About the Financial Model

The financial model includes Extensible Dimensionality, which allows you to extend the Dimensions to suit your own financial purposes. Cubes and design model considerations, along with the use of an aggregate storage model called BI Blend, which supports large-volume reporting for data that is not appropriate to store in a traditional Cube, allow you to further customize financial data modeling. Finally, Hybrid Scenarios help to improve data query performance when analyzing smaller sets of data that contain high volumes of account-level detail.

These tools, along with other data modeling techniques, allow you to build robust financial data models.

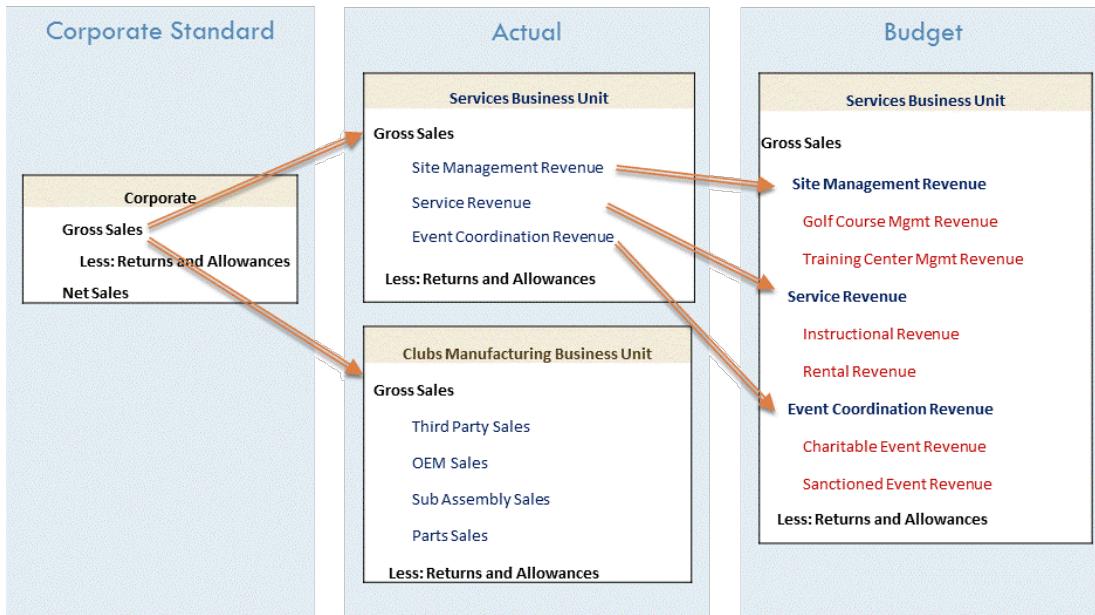
## Extensible Dimensionality

Business units can inherit standard dimensions from a standard set that corporate may maintain, but Extensible Dimensionality® also allows them to extend those dimensions to suit their own process and reporting needs. This allows for operational significance for the business units yet grants control of the overall process to corporate.

The diagram below shows an example of how a certain account can be extended differently across a service business unit vs. a manufacturing business unit as well as across the Actual and Budget scenarios. Notice how each business unit can look at Gross Sales differently in the Actual scenario. Also, the Services business unit can look at Gross Sales at an even greater level of detail in their Budget scenario.

## About the Financial Model

---



This is possible due to three reasons:

1. Extensible Dimensionality® can inherit dimensions and extend them. In the example above, there are four different Account dimensions that inherit from each other like this:
  - Corporate Accounts
    - Club Manufacturing Accounts
    - Services Accounts

## About the Financial Model

---

- Services Budget Accounts

Corporate Accounts is the main chart of accounts. Club Manufacturing takes that dimension and extends it to add its own accounts, but it cannot change what is in Corporate Accounts. Services also takes Corporate Accounts and extends it to meet its needs for the Actual scenario and extends its own Services Accounts to meet its need for more detail in its Budget scenario in the Services Budget Accounts dimension. However, when designing an extended dimension, members are either added below inherited members (for additional detail), or to an alternate rollup by creating new parent members and then referencing inherited members below the new parents. Both actions cannot be done at the same time in the same section of the hierarchy.

2. Different dimensions can be assigned to different cubes and that dimensional assignment can be different for each Scenario type. In the above example, there are three cubes: Corporate, Clubs, and Services. When looking at the Corporate cube, data from the three cubes is all there for analysis. In Corporate, the Corporate Accounts dimension is assigned to all Scenario types. In Clubs, the Clubs Manufacturing Accounts dimension is assigned to all Scenario types. In the Services cube, the Services Accounts dimension is assigned to every dimension except for Budget, which is where the Services Budget Accounts dimension is assigned.
3. The Clubs and Services cubes have their own respective Entity dimensions referenced in the Corporate cube. The Entity dimensions tie the data together.

**NOTE:** Other dimensions such as Flow and the User Defined dimensions can also be extended and have flexible cube assignment if needed.

**NOTE:** Using Extensible Dimensionality to extend accounts used in the Intercompany Matching process is not a recommended practice.

### Extensible Cube

There can also be separate cubes for different uses, such as Human Resource data or Cost Drivers. These cubes can still reference other cube data through the CB# qualifier in Member formulas.

### Extensible Workflow

There can be different Workflow Profile hierarchies per Scenario type which is defined at the cube level. For example, an Actual scenario might be loaded from 12 GL systems across 500 entities, and Budget Forecast, and Variance can define a Workflow for each of the 500 entities with regional review and signoff levels.

## Cube Design Choices within an Application

There are a few choices for the use of Cubes within an application, driven by the relationships between data.

1. Single or “**Monolithic**” Cube is the simplest Application design. These typically have one Workflow Profile structure, though that can vary by Scenario Type.
2. “**Linked Cubes**” is possible via relationships between multiple Entity dimensions into one superset Entity dimension. The “Top” or “Parent” Cube is configured with Cube References to others. Typically, Extensible Dimensionality is deployed with other dimensions, such as Accounts, allowing the Business Unit Cubes to satisfy their management reporting requirements. There is typically one Workflow Profile structure for all Linked Cubes, though that can vary by Scenario Type.
3. “**Exclusive Cubes**” are separate Business Unit Cubes that move their data from a Business Unit to a “Parent” Cube typically via Business Rules or the use of Data Management instead of through configured Cube References. Each separate Cube requires its own Workflow Profile structure, though that can vary by Scenario Type.
4. “**Specialty Cubes**” refers to special data collections outside of the typical Trial Balance or Planning data loads and is typically encompassed with no parent/child relationships between Cubes. Examples are for headcount or budget drivers. These figures might be referenced by other Cubes via Business Rules or Data Management. Each Specialty Cube would have separate Workflow Profile structures.

## Application Model Design Considerations

Application designers understand that, as a platform, each application must flexibly respond to growth, evolving business needs, and enhancements in MarketPlace solutions. Review the following design options to best direct the overall design and performance expectations of your application.

## About the Financial Model

---



When designing an Application, consider these questions to identify the model Cube design best suited to your needs:

- What is the overall size of the Entity, Account, Flow and User Defined dimensions that estimate the Data Unit size potential? A OneStream Data Unit is comprised of Cube, Entity, Parent, Consolidation, Scenario and Time.
- What is the size and relationship between dimensions? For example, are the User Defined Dimensions large (thousand members) and are the members closely tied to a single or very few Entities ("sparsely populated")?
- Are dimension members more transactional in nature? Transaction members such as Employee ID, SKU and Project Code likely change frequently. While this level of detail may be needed for analysis, it can complicate a financial model. Other modeling techniques could help.
- Is the data related to the Cube a "specialty" collection, such as Human Resources, Product Profit or Capital Expenditures?
- Do you fully understand how the data is managed? Determine if the Legal Entity is appropriate for the Entity dimension so you can properly define the ultimate Data Unit Controlling Dimension.
- Have you collected information how the data is assembled and used? Determine if the Cube data is integrated for different types of data such as Actual and Budget.

## Standard Cube Design

Following are considerations for standard cube design.

## Guidelines

- Common design for a standard Statutory Consolidation Cube.
- Primary metadata driven application.
- Data Unit approximately 250k data records at higher end.

## Performance Profile

- Consolidation times for 12 periods generally range from minutes to ~2 hours on the higher end, depending on complexity and volume.
- Performance generally impacted by metadata parent structures, alternate hierarchies, and calculations executing on parent Entity members.

## Reporting Guidelines

- Standard Cube Views with minimum number of rows and columns should perform well, rendering within a few seconds to a minute (depending on requirements).
- Slow Cube View performance of a reasonable number of non-calculated rows and columns, rendering greater than a minute should flag a Cube View or Application design review.
- Cube View Performance is strongly dependent on concurrent use, Cube View design and metadata involved.

## Standard Large Sparse Cube Design

Following are considerations for standard large sparse cube design.

## Design Guidelines

- Common for a Statutory Consolidation Application. Contain sparse intersections due to collecting transactional members, such as SKU, Part Number or Names, or other large dimensions, such as Product or Project.

## About the Financial Model

---

- Data Unit size approximately 750k or more data records. Larger Data Units demand more CPU processing power and time to calculate.

## Performance Profile

- Optimize consolidation times by addressing sparse dimensions. With good extensibility you can significantly reduce the size of the Data Unit consolidating, for faster consolidation.
- Applications that must be consolidated and contain large sparse dimension members, may have long consolidation times and lower overall performance. Review these applications designs to consider alternate design solutions that include extensibility or other Cube and Model solutions.
- Data Unit size, number of Data Units, Business Rule best practices and other factors effect consolidation times.

## Reporting Guidelines

- Putting a “top” member such as UD2#Top for a large dimension on a Cube View , means OneStream loops through each child member and aggregates on-the-fly. This can consume large amounts of processing power and extend report processing.
- It is best to use Sparse Data Suppression settings on Cube Views which are based on Single Data Unit.
- You may need custom Sparse Cube View Business Rules for “top of the house” queries.
- Quick Views may be impacted as they do not include Sparse Row Suppression settings.

## Hybrid Cube Design

Following are considerations for hybrid cube design.

### Guidelines

Hybrid Applications are generally designed to support Analytics and not the Consolidation model. Hybrid cubes:

## About the Financial Model

---

- Typically require complex calculations at base level Entities.
- Predominately present is aggregation having simple needs at parent Entity levels. Parents do not require special calculations or logic.
- Rely on Dynamic Business Rules attached to a dynamic member such as in UD8. This means parent Entity data is not stored. You can set the Is Consolidated property on these Entities to False.
- Determine if parent Entity data must be exported to external system so you can understand the impact of Dynamic Business Rules.
- Usually have a data unit size of up to one million data records at the top parent Entity level.

## Performance Profile

- Performance will vary by application size and structure.
- Data calculation is not stored at the parent level.
- “Entity Aggregation on the Fly” consolidation simulation rules are required. You cannot consolidate parent entities. This involves the API.Functions.GetEntityAggregationDataCell function in a dynamic member.
- You must support the export of calculated data which could involve unique Scenarios or cubes to replicate dynamic calculations as stored values. These processes can be automated using scheduled Data Management jobs.

## Reporting Guidelines

- Apply Sparse Suppression settings on Cube Views.
- You may need custom Sparse Cube View Business Rules for “top of the house” member queries.
- Intermediate rules may run as Workflow-based or be driven by Dashboards or Calculation Definition settings.
- Quick Views may be impacted because they do not include Sparse Row Suppression settings.

## BI Blend

BI-Blend is a “read-only” aggregate storage model that supports large-volume reporting for data that is not appropriate to store in a traditional Cube. BI Blend data is large in volume and often transactional. For example, to analyze data by invoice, a standard cube requires metadata to store the data records. Soon, most of the invoice metadata would not be needed given the transactional nature of the data. Therefore, storing transaction data in a Cube design is not a best practice.

A key challenge of reporting on transactional data is presenting data in a uniform format for standardized reporting, while flexibility supporting ever changing records and reporting requirements. The overall large size of the data sets requires a model suitable for responsive reporting and analysis.

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

Use BI-Blend to analyze large volumes of highly changing data, such as ERP system transaction data, which typically would not reside in a Cube. Processing is free 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 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

For more information, see the *BI Blend Design and Reference Guide*.

## Relational Blending

The dynamic blending of data stored in a relational database with data stored/derived by the in-memory analytic engine is supported.

Relational blending is a data management approach. This enables analytic modelers to better manage the trade-offs between building unsustainable/poor performing complex analytic models that contain too much detail and the need to report on that level detail (data that may be transactional in nature or constantly changing). Relational Blending makes it possible to seamlessly integrate detail data points into the analytic reporting process.

## Relational Blending Types

There are three methods of relational blending.

### 1. Drill-Back Blending (One-to-Many Relationship)

This method of relational blending is used to provide access to detailed information that does not exist in the analytic model. This capability is delivered right out-of-the box with its predefined drill back to stage detail data. In addition, the stage integration engine provides drill-back and drill-around capabilities against external data.

### 2. Application Blending (One-to-Many Relationship)

This method of relational blending leverages the OneStream MarketPlace Specialty Planning and Compliance applications. This collects information in a transactional register format and seamlessly maps/loads summarized data into an analytic model. These applications also provide predefined transactional level reports as well as predefined drill-back connectors allowing drill-down from a summarized analytic model to the detailed register transaction data.

### 3. Model Blending (One-to-One Relationship)

This method of relational blending combines the power of the in-memory analytic engine with the flexibility of relational database storage. This functionality is provided as part of the Finance Engine API and the functions can be found under the api.Functions path in a Finance Business Rule.

## Model Blending

The following sections provide sample use-cases and explanations of how Model Blending is used to seamlessly integrate relational data points into an analytic model using Dynamic Member Calculations that leverage the relational blending API functions of the finance engine.

A common challenge in analytic modeling is how to build a sustainable model when the definition of metadata and data becomes blurred. The two use cases below provide examples of this modeling challenge.

## Use Case 1

In many cases, analytic modelers are faced with the challenge of building a model containing Dimension Members that are unknown at design time or forced to build a Dimension containing Members that will be constantly changing. Without relational blending, analytic modelers are forced to build models full of unknown members (TBD1, TBD2, etc.) with the hopes that users of the system do not need values beyond these placeholder members. (This data is transactional and not a good candidate for an analytic model, Workforce Planning is a good example of this problem)

This is a one-to-many issue so Drill-Back and Application Blending work well if the summary Cube is the primary focus of analysis and transactions are only used for supporting details. Model blending can provide a benefit as well, but keep in mind that model blending must relate an analytic cell POV to a relational row or summarized row (one-to-one). For Model Blending to be useful in these circumstances, the relational data must be returned in an aggregate format (avg, min, max, sum, count) in order to reduce the one-to-many relationship to a one-to-one Relationship.

## Use Case 2

Analytic models that depend on Dimensions with Members that are constantly changing. Consider business problem where the analytic model is based on a fixed number of members (facility with rooms and beds). This is easy from a modeling perspective; however, the user requirement is to build a model that is aware of the current occupant of the bed. The logical metadata definition is room/bed, but the business problem requires the “occupant” to be defined as a Member for the model to be meaningful. If occupant is used as a Member in the model, it is almost guaranteed that the analytic model will eventually become unsustainable due to the changing nature of the room/bed/occupant Dimension. The administrator of this model now has the burden of constantly changing and rebuilding the model to reflect the current occupant data.

This is a one-to-one issue, so Model Blending fits well and provides a tremendous amount of value. Detailed and changing information can be continuously loaded and updated as attribute information in the OneStream Staging tables, Custom Relational Tables and the Model Blending API can be used to dynamically incorporate this information into analytic model through dynamic member formulas.

## Model Blending Benefits

Relational blending is similar to the OneStream Staging engine in that it is a tool to protect the analytic engine. Analytic modelers are aware that there is a powerful force with which they must contend when they are trying to create a well-designed, well-performing and maintainable model. That powerful force is Factorial Combination Math. Analytic modelers understand that numbers of possible cells in a model (combinations) is determined by the number of Members in each Dimension multiplied by each other (1,000 Accounts x 100 Cost Centers x 10,000 Employees x 20 Regions = 20 billion combinations). This phenomenon means that analytic modelers are in a constant battle. They are trying to capture the data points required to understand the business process being modeled and model performance challenges created by the computational physics of factorial combination math. In summary, it is easy to create an analytic model with a massive number of potential cells and as a result, end up with a poor performing model.

Relational blending can help keep the size of an analytic model to a manageable level by allowing leaf level members to be kept in the relational table and only keeping summarize/static Members in the analytic model definition (Dimension Members). Relational blending is not a cure-all, but it is an important tool for building maintainable well performing analytic models when a model has some dependencies on detailed information that cannot be clearly defined as metadata or data. In other words, the information is useful in the model, but it is so detailed or changes so much that it is difficult to incorporate into a rational metadata structure.

## Relational Blending API

The Relational Blending API functions listed below can be used to efficiently lookup a value in a cached relational table. This is based on the current POV values of the analytic engine or by providing specific override values for the Cube Name, Entity Name, Scenario Name, Time Name and Account Name. It is important to understand how caching of the data table is done. The CacheLevel parameter is used to control the cache granularity which will in turn control cache efficiency. To choose the most efficient CacheLevel value, determine how the data will be used in a Cube View. If the primary Cube View data request will be on a single data unit (Entity, Scenario, Time), then the best cache level choice would be BlendCacheLevelTypes.WfProfileScenarioTimeEntity.

This is an efficient choice because the first time a cell is requested for the Entity, Scenario, Time combination, a query will run to load all the stage data for the combination. Then all subsequent cell requests would read values from cache. On the other hand, if the primary Cube View data request is for multiple Entities, then BlendCacheLevelTypes.WfProfileScenarioTime cache level would be a more efficient choice. This is more efficient because a single query would run to load all the data for the Scenario and Time into cache and then all subsequent cell requests would read values from cache. As a cautionary note, be aware that using coarse cache levels (reading more data at once into cache) only benefits performance when the target Cube View can read many values for the cache. If the target Cube View is only focused on one Entity/Account and the BlendCacheLevelTypes.WfProfileScenarioTime cache level is chosen, all rows for the entire Scenario and Time would need to be read into memory when only values for one Entity/Account combination was needed. In this case, BlendCacheLevelTypes.WfProfileScenarioTimeEntityAccount would be a more efficient cache level.

In summary, choose a cache level that will minimize the number of actual database queries needed to run in order to get the desired cells for the target Cube View. This is not an exact science, and it may be difficult to choose a cache level that works efficiently for all target Cube Views. If there is a diverse set of Cube Views using relational blend data, consider creating specific Members that implement different cache levels that match the Cube View data pattern.

## Cache Level Types

### **BlendCacheLevelTypes.WfProfileScenarioTime**

Query will be run and cached using the supplied Workflow Profile, POV Scenario and POV Time as criteria and cache key.

### **BlendCacheLevelTypes.WfProfileScenarioTimeEntity**

Query will be run and cached using the supplied Workflow Profile, POV Scenario, POV Time and POV Entity as criteria and cache key.

### **BlendCacheLevelTypes.WfProfileScenarioTimeAccount**

Query will be run and cached using the supplied Workflow Profile, POV Scenario, POV Time and POV Account as criteria and cache key.

### **BlendCacheLevelTypes.WfProfileScenarioTimeEntityAccount**

Query will be run and cached using the supplied Workflow Profile, POV Scenario, POV Time, POV Entity and POV Account as criteria and cache key.

### **BlendCacheLevelTypes.Custom**

Intended to be used with custom table query (Cache level is explicitly controlled by the supplied SQL query). Query will be run and cached using the supplied cache name.

## Relational Model Blending API Functions

### GetStageBlendTextUsingCurrentPOV

#### Function Prototype

```
Public Function GetStageBlendTextUsingCurrentPOV (ByVal cacheLevel As  
BlendCacheLevelTypes, ByVal cacheName As String, ByVal wfProfileName As String, ByVal  
fieldList As String, ByVal criteria As String, ByVal fieldToReturn As String, ByVal textOperation As  
BlendTextOperationTypes) As String
```

#### Description

Read a stage text attribute value from a cached ado.net data table using the current POV values and optionally perform concatenation on the results.

**NOTE:** Cache only lives for the duration of the WCF call.

## Parameters

#### cacheLevel

Cache granularity level used to control how much information is cached in each chunk (Less granular cache level helps repeated calls but hurts requests for a single cell because more data is cached than necessary).

#### cacheName

Short name used to identify the values placed in the cache (Full CacheID will be CacheName + CacheLevel Values).

#### wfProfileName

Name of the Import Workflow profile containing the values to be looked up. (Pass an empty String to look up the workflow based on the POV Entity, use \*.YourWFSuffix to get workflow profiles with the specified suffix.)

#### fieldList

List of STAGE fields that will be used as criteria and/or returned.

#### Criteria

Criteria statement used to select rows in the cached data table.

#### fieldToReturn

Name of the stage field to return.

## About the Financial Model

---

### **textOperation**

Text operation to perform on the resulting data table (Note: FirstValue returns the first matching row if there is more than one stage value for the specified cell).

Return Type

String

## Example

```
'UD8 DynamicCalc - Lookup Attribute 1 From Stage
If Not api.Entity.HasChildren Then
    Dim criteria As New Text.StringBuilder
    criteria.Append("U1T = '" & api.Pov.UD1.Name & "' ")
    criteria.Append("And U2T = '" & api.Pov.UD2.Name & "' ")
    Return api.Functions.GetStageBlendTextUsingCurrentPov(BlendCacheLevelTypes.
        WfProfileScenarioTimeEntity, "DU",
        "*.Sales Detail", "U1T,U2T,A1,ConvertedAmount", criteria.ToString,
        "A1", BlendTextOperationTypes.FirstValue)

Else
    Return String.Empty
End If
```

## GetStageBlendText

### **Function Prototype**

```
Public Function GetStageBlendText (ByVal cubeName As String, ByVal entityName As String,
    ByVal scenarioName As String, ByVal timeName As String, ByVal accountName As String, ByVal
    cacheLevel As BlendCacheLevelTypes, ByVal cacheName As String, ByVal wfProfileName As
    String, ByVal fieldList As String, ByVal criteria As String, ByVal fieldToReturn As String, ByVal
    textOperation As BlendTextOperationTypes) As String
```

### **Description**

Read a stage text attribute value from a cached ado.net data table using the specified POV values and optionally perform concatenation on the results.

**NOTE:** Cache only lives for the duration of the WCF call.

## Parameters

### **cubeName**

Name of the Cube to use for the POV.

## About the Financial Model

---

### **entityName**

Name of the Entity to use for the POV.

### **scenarioName**

Name of the Scenario to use for the POV.

### **timeName**

Name of the Time to use for the POV.

### **accountName**

Name of the Account to use for the POV.

### **cacheLevel**

Cache granularity level used to control how much information is cached in each chunk (Less granular cache level helps repeated calls but hurts requests for a single cell because more data is cached than necessary).

### **cacheName**

Short name used to identify the values placed in the cache (Full CacheID will be CacheName + CacheLevel Values).

### **wfProfileName**

Name of the Import Workflow profile containing the values to be looked up (Pass an empty String if to look up the workflow based on the POV Entity, use \*.YourWFSuffix to get workflow profiles with the specified suffix).

### **fieldList**

List of Stage fields that will be used as criteria and/or returned.

### **Criteria**

Criteria statement used to select rows in the cached data table.

### **fieldToReturn**

Name of the Stage field to return.

### **textOperation**

Text operation to perform on the resulting data table (Note: FirstValue returns the first matching row if there is more than one stage value for the specified cell).

### **Return Type**

String

## **GetStageBlendNumberUsingCurrentPOV**

## About the Financial Model

---

### Function Prototype

```
Public Function GetStageBlendNumberUsingCurrentPOV(ByVal cacheLevel As  
BlendCacheLevelTypes, ByVal cacheName As String, ByVal wfProfileName As String, ByVal  
fieldList As String, ByVal criteria As String, ByVal fieldToReturn As String, ByVal mathOperation  
As BlendNumericOperationTypes) As Decimal
```

### Description

Read a stage numeric attribute value from a cached ado.net data table using the current POV values and optionally perform aggregation math on the results.

**NOTE:** Cache only lives for the duration of the WCF call.

## Parameters

### cacheLevel

Cache granularity level used to control how much information is cached in each chunk (Less granular cache level helps repeated calls but hurts requests for a single cell because more data is cached than necessary).

### cacheName

Short name used to identify the values placed in the cache (Full CacheID will be CacheName + CacheLevel Values).

### wfProfileName

Name of the Import Workflow profile containing the values to be looked up (Pass an empty String if you want to look up the workflow based on the POV Entity, use \*.YourWFSuffix to get workflow profiles with the specified suffix).

### fieldList

List of stage fields that will be used as criteria and/or returned.

### criteria

Criteria statement used to select rows in the cached data table.

### fieldToReturn

Name of the stage field to perform math on and return.

### mathOperation

Math operation to perform on the resulting data table (Note: FirstValue returns the first matching row if there is more than one stage value for the specified cell).

### Return Type

Decimal

### Example

## About the Financial Model

---

```
'UD8 DynamicCalc - Lookup Average ConvertedAmount From Stage
If Not api.Entity.HasChildren Then
    Dim criteria As New Text.StringBuilder
    criteria.Append("U1T = '" & api.Pov.UD1.Name & "' ")
    criteria.Append("And U2T = '" & api.Pov.UD2.Name & "' ")
    Return api.Functions.GetStageBlendNumberUsingCurrentPov(BlendCacheLevelTypes.
        WfProfileScenarioTimeEntity, "DU",
        "*.Sales Detail", "U1T,U2T,A1,ConvertedAmount", criteria.ToString,
        "ConvertedAmount",
        BlendNumericOperationTypes.AverageSkipZero)
Else
    Return 0
End If
```

### GetStageBlendNumber

#### Function Prototype

```
Public Function GetStageBlendNumber(ByVal cubeName As String, ByVal entityName As String,
    ByVal scenarioName As String, ByVal timeName As String, ByVal accountName As String, ByVal
    cacheLevel As BlendCacheLevelTypes, ByVal cacheName As String, ByVal wfProfileName As
    String, ByVal fieldList As String, ByVal criteria As String, ByVal fieldToReturn As String, ByVal
    mathOperation As BlendNumericOperationTypes) As Decimal
```

#### Description

Read a stage numeric attribute value from a cached ado.net data table using the specified POV values and optionally perform aggregation math on the results.

**NOTE:** Cache only lives for the duration of the WCF call.

## Parameters

### **cubeName**

Name of the Cube to use for the POV.

### **entityName**

Name of the Entity to use for the POV.

### **scenarioName**

Name of the Scenario to use for the POV.

### **timeName**

Name of the Time to use for the POV.

### **accountName**

Name of the Account to use for the POV.

## About the Financial Model

---

### **cacheLevel**

Cache granularity level used to control how much information is cached in each chunk. Less granular cache level helps repeated calls but hurts requests for a single cell because more data is cached than necessary.

### **cacheName**

Short name used to identify the values placed in the cache (Full CacheID will be CacheName + CacheLevel Values).

### **wfProfileName**

Name of the Import Workflow profile containing the values to be looked up. Pass an empty String to look up the workflow based on the POV Entity, use \*.YourWFSuffix to get workflow profiles with the specified suffix.

### **fieldList**

List of Stage fields that will be used as criteria and/or returned.

### **criteria**

Criteria statement used to select rows in the cached data table.

### **fieldToReturn**

Name of the stage field to perform math on and return.

### **mathOperation**

Math operation to perform on the resulting data table

**NOTE:** FirstValue returns the first matching row if there is more than one stage value for the specified cell

### **Return Type**

Decimal

## **GetStageBlendDataTableUsingCurrentPOV**

### **Function Prototype**

```
Public Function GetStageBlendDataTableUsingCurrentPOV(ByVal cacheLevel As  
BlendCacheLevelTypes, ByVal cacheName As String, ByVal wfProfileName As String, ByVal  
fieldList As String) As DataTable
```

### **Description**

Read stage data into a cached ado.net data table using the current POV values so that it can be Queried / Analyzed in memory on the application server.

**NOTE:** Cache only lives for the duration of the WCF call.

# Parameters

### **cacheLevel**

Cache granularity level used to control how much information is cached in each chunk (Less granular cache level helps repeated calls but hurts requests for a single cell because more data is cached than necessary).

### **cacheName**

Short name used to identify the values placed in the cache (Full CacheID will be CacheName + CacheLevel Values).

### **wfProfileName**

Name of the Import Workflow profile containing the values to be looked up (Pass an empty String to look up the workflow based on the POV Entity, use \*.YourWFSuffix to get workflow profiles with the specified suffix).

### **fieldList**

List of Stage fields that will be used as criteria and/or returned.

### **Return Type**

DataTable

### **Example**

```
'Lookup Attribute 1 From Stage Cached Data Table
If Not api.Entity.HasChildren Then
    Dim result As String = String.Empty

'Get the DataTable from cache
Using dt As DataTable =
    api.Functions.GetStageBlendDataTableUsingCurrentPov(BlendCacheLevelTypes.WfProfileScenarioTimeEntity,
    "DU", "*.Sales Detail", "U1T,U2T,A1,ConvertedAmount")
    If Not dt Is Nothing Then
        'Execute a query against the data table and return the first matching row
        Dim criteria As New Text.StringBuilder
        criteria.Append("U1T = '" & api.Pov.UD1.Name & "' ")
        criteria.Append("And U2T = '" & api.Pov.UD2.Name & "' ")
        Dim rows As DataRow() = dt.Select(criteria.ToString)
        If rows.Count > 0 Then
            result = rows(0)("A1")
        End If
    End If
End Using
Return result
Else
    Return String.Empty
End If
```

### **GetStageBlendDataTable**

## About the Financial Model

---

### Function Prototype

```
Public Function GetStageBlendDataTable(ByVal cubeName As String, ByVal entityName As String, ByVal scenarioName As String, ByVal timeName As String, ByVal accountName As String, ByVal cacheLevel As BlendCacheLevelTypes, ByVal cacheName As String, ByVal wfProfileName As String, ByVal fieldList As String) As DataTable
```

### Description

Read stage data into a cached ado.net data table using the specified POV values so that it can be Queried / Analyzed in memory on the application server.

**NOTE:** Cache only lives for the duration of the WCF call.

## Parameters

### **cubeName**

Name of the Cube to use for the POV.

### **entityName**

Name of the Entity to use for the POV.

### **scenarioName**

Name of the Scenario to use for the POV.

### **timeName**

Name of the Time to use for the POV.

### **accountName**

Name of the Account to use for the POV.

### **cacheLevel**

Cache granularity level used to control how much information is cached in each chunk. (The less granular cache level helps repeated calls but hurts requests for a single cell because more data is cached than necessary.)

### **cacheName**

Short name used to identify the values placed in the cache (Full CacheID will be CacheName + CacheLevel Values).

### **wfProfileName**

Name of the Import Workflow profile containing the values to be looked up (Pass an empty String to look up the workflow based on the POV Entity, use \*.YourWFSuffix to get workflow profiles with the specified suffix).

### **fieldList**

List of Stage fields that will be used as criteria and/or returned.

## About the Financial Model

---

### Return Type

DataTable

### GetCustomBlendDataTableUsingCurrentPOV

#### Function Prototype

Public Function GetCustomBlendDataTableUsingCurrentPOV(ByVal cacheLevel As BlendCacheLevelTypes, ByVal cacheName As String, ByVal sourceDBLocation As String, ByVal sourceSQL) As DataTable

#### Description

Read data from a custom table into a cached ado.net data table using the current POV values so that it can be Queried / Analyzed in memory on the application server.

**NOTE:** Cache only lives for the duration of the WCF call.

## Parameters

### cacheLevel

Cache granularity level used to control how much information is cached in each chunk (Less granular cache level helps repeated calls but hurts requests for a single cell because more data is cached than necessary).

### cacheName

Short name used to identify the values placed in the cache (Full CacheID will be CacheName + CacheLevel Values).

### sourceDBLocation

Database location name to query (Application, Framework, or a Named External Connection).

### sourceSQL

SQL statement that defines the DataTable to cache for in memory querying.

### Return Type

DataTable

Example:

```
'Lookup Invoice Number From a Custom Table
If Not api.Entity.HasChildren Then
    Dim result As String = String.Empty

    Dim queryToCache As New Text.StringBuilder
    queryToCache.Append("Select * ")
    queryToCache.Append("From InvoiceMaterialDetail ")
    queryToCache.Append("Where PlantCode = 'H200' And CustId = 'NH2421' And InvYear = 2011
```

## About the Financial Model

---

```
    and InvMonth = 'M3'      ")
'Get the DataTable from cache
Using dt As DataTable = api.Functions.GetCustomBlendDataTableUsingCurrentPov(BlendCacheLevelTypes.
    Custom, "Material",
    "Revenue Mgmt System", QueryToCache.ToString)
If Not dt Is Nothing Then
'Execute a query against the CUSTOM data table and return column InvNo for the first matching row
    Dim criteria As New Text.StringBuilder
    criteria.Append("WorkDay = 15")
    Dim rows As DataRow() = dt.Select(criteria.ToString)
    If rows.Count > 0 Then
        result = rows(0)("InvNo")
    End If
End If
End Using
Return result
Else
    Return String.Empty
End If
```

### GetCustomBlendDataTable

#### Function Prototype

Public Function GetCustomBlendDataTable(ByVal cacheLevel As BlendCacheLevelTypes,  
ByVal cacheName As String, ByVal sourceDBLocation As String, ByVal As String) As DataTable

#### Description

Read data from a custom table into a cached ado.net data table using the specified POV values so that it can be Queried / Analyzed in memory on the application server.

**NOTE:** Cache only lives for the duration of the WCF call.

## Parameters

### cacheLevel

Cache granularity level used to control how much information is cached in each chunk. (The less granular cache level helps repeated calls but hurts requests for a single cell because more data is cached than necessary.)

### cacheName

Short name used to identify the values placed in the cache (Full CacheID will be CacheName + CacheLevel Values).

### sourceDBLocation

Database location name to query (Application, Framework, or a Named External Connection).

### sourceSQL

SQL statement that defines the DataTable to cache for in memory querying.

### Return Type

DataTable

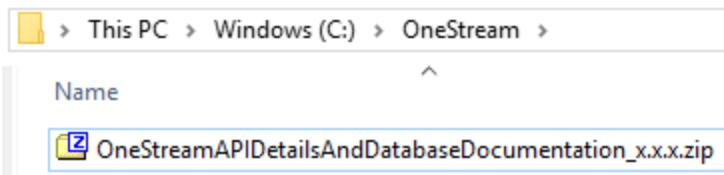
## Formulas

A formula is a set of calculation instructions to compute values. Formulas are written using Microsoft Visual Basic .NET procedures that use OneStream API function libraries and member script expressions. These combined capabilities provide a powerful programming environment delivering reliable compiled formula definitions.

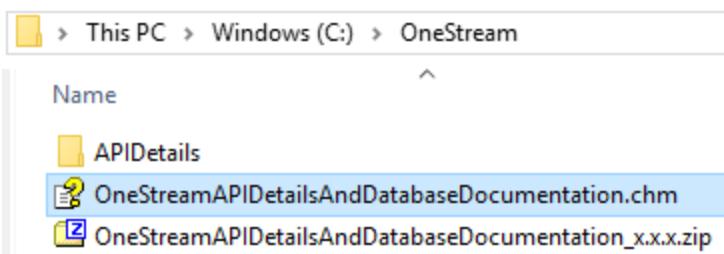
## OneStream API Details and Database Documentation

Download the OneStream API Overview Guide for detailed background on Business Rules and the Platform engines. This can be found on MarketPlace under Software Download. Also available in the same download is more information on OneStream API functions and details on the OneStream Framework and Application database tables and indexes. This can be found in the OneStream API Details and Database Documentation.

To use the OneStream API Details and Database Documentation, create a folder on the PC on which this will be loaded and copy the related zip file:



Right-click and extract the zipped file's contents here. Double-click the file that ends in chm. This file launches the API Guide.



## About the Financial Model

---

Contents are organized by the related Platform Engine (see Platform Engines). These are broken down into Classes (e.g. DataApi), Overload Lists, Methods (e.g. GetDataCell), Syntax and Parameters. The Index and Search tabs can be used to search by function name, enumerations, properties, etc.

## Formula Structure

### Microsoft Visual Basic.NET With OneStream API and Member Scripts

All formulas and business rules run as compiled VB.NET code. In a VB.NET function or subroutine, calls are made to specific API functions which enable the rule writer to interact with the Analytic Engine. Specific API functions are used to process member script expressions, and create calculated values in the analytic data model.

### Common Finance APIs That Use Member Scripts

```
api.Data.GetDataCell("A#Cash")
```

This returns a single numeric cell value.

```
api.Data.GetDataCellEx("A#Cash")
```

This returns a single numeric or text cell value.

```
api.Data.SetDataCell("A#Cash", 5.00, False)
```

This saves/writes a single data cell value.

```
api.Data.Calculate("A#Cash = A#Sales * 0.10")
```

This saves multiple data cells as stored values.

## Basic Dynamic Cell Calculation

The following return a single cell.

Calculate Current Ratio

```
Return api.Data.GetDataCell("A#15000:0#Top / A#22000:0#Top")
```

Formula Composition:

VB.NET	Language Keyword(s)
Return	

## About the Financial Model

---

### OneStream API Function

```
api.Data.GetDataCell("Member Script")
```

### OneStream Member Script

```
A#15000:0#Top / A#22000:0#Top
```

**TIP:** When using a nested function (YearPrior(|POVTime|)) in a GetDataCell call, you must enclose the function in square brackets []. This is necessary because GetDataCell and the functions use “(“ and “)” as indicators of the start/end points in the script. The inner functions must be wrapped in square brackets.

```
GetDataCell("S#[!ScenarioActual_GRT!]:T#[YearPrior1(|CVTime|)][Period  
(|CVTime|)]":Name("PYR Var"))
```

## Basic Stored Calculation

The following calculate and store multiple cells.

### Pull Prior Period Retained Earnings

```
api.Data.Calculate("A#28000 = A#28999:T#POVPrior1")
```

### Formula Composition

```
OneStream API Function  
api.Data.Calculate("Member Script")
```

### Member Script

```
A#28000=A#28999:T#POVPrior1
```

You can use formula variables in member scripts to significantly improve performance when the same formula is used for multiple members. When using formula variables, the formula text remains the same, so there is no need for continued parsing and evaluation.

Using variables can also improve performance if a member Id is used instead of a member name as the ID can also be used as the value in a formula variable. To use a formula variable in a member script, use a dollar sign \$ instead of a pound # sign before the member name, and use the variable name after the dollar sign.

### Example 1

```
api.Data.FormulaVariables.SetTextVariable("variableAccount", "8150")  
api.Data.Calculate("A#8250=A$variableAccount * 10")
```

### Example 2

## About the Financial Model

---

```
Dim acctMember As Member = api.Members.GetMember(DimType.Account.Id, "8150")
api.Data.FormulaVariables.SetMemberVariable("variableAccount",acctMember)
api.Data.Calculate("A#8250= A$variableAccount * 100")
```

### Example 3

```
Dim acctMember As Member = api.Members.GetMember(DimType.Account.Id, "8150")
Dim acctId As Integer = acctMember.MemberId
api.Data.FormulaVariables.SetIntegeVariable("myAccount",acctId)
api.Data.Calculate("A#8250 = A$myAccount * 1000")
```

## Applying Formulas

An administrator can create a formula on a Dimension Member or in a Business Rule.

### Member Formulas

The preferred and more common approach is to write formulas on Dimension Members. Member formulas are written using the Formula property on individual Scenario, Account, Flow, or User Defined Members in the Dimension Library. The primary reasons for writing formulas on Dimension Members are they provide intuitive formula organization and promote reusability of Dimension and Members with their associated calculation across multiple Cubes. They also enable parallel processing for performance optimizations using advanced multi-threading that executes multiple formulas at the same time. Finally, writing formulas on Dimension Members support drill down from a calculated amount to the amount used as the source for calculation and they support the ability to vary formulas by Scenario and Time. This is useful because changes can be made without affecting the calculation for older data or data in other Scenarios.

### Finance Business Rules

Formulas can also be written using one or more Business Rule files. While this typically is not the preferred approach, it provides the ability to put all formulas in one location, and it is similar to the approach used by some older competitive products. The primary reasons for writing formulas in a Finance Business Rule are the formula requires extensive cross-dimensional dependencies and it is not clear on which Member an equivalent Member Formula should be written, or the formula requires complex sequential logic, variable, or conditional statements that affect multiple dependent calculations. Another reason for writing formulas in a Finance Business Rule is when the application requires custom algorithms for Currency Translation, Share, or Intercompany Eliminations.

**TIP:** Due to the sophisticated built-in translation and consolidation algorithms, most applications only require Member Formulas for the Data Unit Calculation Sequence (DUCS) (i.e., Chart Logic). Custom Business Rules for Translation, Share, and Intercompany Eliminations are not typically needed.

**NOTE:** Due to the way OneStream stores its data, Decimal should always be used instead of Double or Integer when declaring variables that return a number within a Business Rule or Member Formula.

```
Dim myNumber as Decimal = Api.Data.GetDataCell("A#Sales - A#CostOfSales").CellAmount
```

See Business Rules in "Application Tools" on page 727 for more details.

## Formula Types

Two types of formulas can be applied to dimensional members or members in a business rule file:

- Dynamic Cell Calculation (Dynamic Calc): An in-memory calculation that runs on demand when a cell containing a dynamic member formula is requested. A Dynamic Calc formula computes a value for a single cell and runs whenever the cell needs to be displayed without storing the result.
- Stored Calculation: A persisted calculation that runs as part of the Data Unit Calculation Sequence (DUCS). With these formulas you can calculate many cells simultaneously, such as data buffer.

## Performance Considerations

**Dynamic Cell Calculation (Dynamic Calc):** Dynamic Cell Calculations enhance the consolidation process because the amount is calculated when requested for display and is not written to the database. For reporting however, performance may be impacted because data is calculated on demand. Dynamic calculations are usually used for ratio or percentage calculations.

**Stored Calculation:** Consolidation performance is directly impacted by the volume and complexity of stored calculations. Carefully consider each stored calculation since one poorly written rule can cause large amounts of data to be written to the cube, negatively impacting consolidation performance. If you use many member formulas or if data volumes are not considered in member formulas, over 100,000 stored numbers may be generated from just 1,000 initially loaded numbers. The quantity of stored numbers is a critical factor to consider to optimize consolidation performance.

## Reference and Aggregation Considerations

**Dynamic Cell Calculations:** Dynamic cell calculations can reference other dynamic cell calculations and stored calculations but should not be used by stored calculations because they do not naturally aggregate parent members. For example, if a parent member in an Account, Flow, or User Defined dimension has a child member calculated by a dynamic calculation formula, the parent excludes the child member from the aggregated amount. Achieve aggregation by writing another dynamic calculation on the parent member.

**Stored Calculations:** Stored calculations can reference other stored calculations and parent members aggregate naturally.

**TIP:** You do not need to write a stored calculation to add or subtract individual members. Instead, create an alternative member hierarchy and use the Aggregation Weight property, set to -1.0 to negate. This aggregated value is dynamic and supports drill-down.

**Stored Calculation Evaluation Tools:** You can access detailed process logging information. Run **Force Consolidate With Logging** and click **Task Activity** to analyze steps in the consolidation process to identify bottlenecks or errors for performance optimizations. Examples:

- Calculate with Logging
- Translate with Logging
- Consolidate with Logging

**Formula Calculation Threshold Monitoring:** Calculation time threshold values can be set in the Application Server Configuration File using the **NumSecondsBeforeLoggingSlowFormulas** setting to log formulas exceeding the specified threshold time.

## Formula Execution

This section defines the types of calculations and calculation sequences used during different analytic processing routines.

## Cube Processing

**Calculation:** This executes the standard calculation sequence for a single Data Unit. A Data Unit refers to a group of data cells for a specific Cube, Entity, Parent, Consolidation, Scenario, and Time Member. See Data Units in "Workflow" on page 445 for more details on Data Units. Except for Dynamic Cell Calculations, all Member Formulas are written to execute as part of a Data Unit's Calculation Sequence.

**Translation:** This executes a currency translation that occurs when the data for an Entity's local currency needs to be translated to a foreign currency. The translation step executes after the system has run the Data Unit Calculation Sequence for an Entity's local currency. After this is completed, the default translation algorithms use Foreign Exchange (FX) rates to generate and store a corresponding translated Data Unit. Finally, the Data Unit Calculation Sequence on the translated Data Unit to produce the final translated amounts are run.

**Consolidation:** The Analytic Engine provides pre-built financial intelligence through a statutory Consolidation Dimension that defines a sequence of Data Unit calculations and aggregations which include currency translations, Parent-level adjustments, complex ownership computations, and Intercompany Eliminations. For more details on Consolidation, see Consolidation.

**Dynamic Cell Calculation:** Dynamic Cell Calculations are a special type of Member Formula for Account, Flow, or User Defined Members. They are used to generate amounts for a Member on the fly (i.e., the results are calculated on-demand and are not stored). Dynamic Cell Calculations are often used for metric accounts (e.g., ratios involving other accounts) and are appropriate when the result of the Dynamic Calculation is not needed as the source number for another Stored Calculation. The use of Dynamic Cell Calculations can result in improved Consolidation performance because they do not generate stored numbers and are typically only executed when a number needs to be displayed, not during the Consolidation process.

## Data Units and Formula Execution

OneStream executes formulas at a specific unit of work call a Data Unit. This section details the order and combination of logical processes that execute for a Data Unit. See Using Data Units in "Workflow" on page 445.

**Guidelines on Formula Passes:** There are 16 Formula Passes and one Dynamic Calculation Formula Type available on each Member. Formulas that do not depend on one another can exist in the same formula pass and will be calculated in parallel using OneStream's advanced parallel formula processing engine.

As a basic guideline, customers should think about organizing formulas by account/collection type across OneStream's available formula passes using the following examples.

**Formula Pass 1 - 8: Trial Balance**

**Formula Pass 5-7:** Translation logic related to Cash Flow

**Formula Pass 8 or 9:** Balance Account and CTA account

**Formula Pass 9 - 16:** Non-trial Balance

All formulas in a pass are processed at the same time, so they cannot have dependencies on one another.

**Calculation Sequence of a Single Data Unit:** The items below detail the specific list of tasks executed for each Data Unit's calculation process. As an example, the following steps are executed for a single Data Unit when a user selects Calculate for a single Entity, Scenario, and Time period.

## Data Unit Calculation Sequences (DUCS)

1. Clear previously calculated data for the Data Unit.
2. Run the Scenario's Member Formula, which is typically used for seeding a Scenario's data from another Scenario or from a prior year.
3. Run reverse translations by calculating Flow Members from other Alternate Currency Input Flow Members. This is part of the built-in ability for an Entity to accept input using multiple currencies.
4. Execute Business Rules (1 and 2). Up to 8 Business Rule files can be attached to each Cube.
5. Run Formula Passes (1 – 4) for the Cube's Account Dimension Members, then Flow Members, and then User Defined Members. The Formula Pass is specified using each Member's Formula Type property in the Dimension Library.
  - Account, Flow, UD1, UD2, ... UD8 (Member Formula Execution)
6. Execute Business Rules (3 and 4).
7. Run Formula Passes (5 – 8).
  - Account, Flow, UD1, UD2, ... UD8 (Member Formula Execution)
8. Execute Business Rules (5 and 6).
9. Run Formula Passes (9 – 12).

- Account, Flow, UD1, UD2, ... UD8 (Member Formula Execution)
10. Execute Business Rules (7 and 8).
11. Run Formula Passes (13 – 16).
- Account, Flow, UD1, UD2, ... UD8 (Member Formula Execution)

Every time a data cell is written to the database, information with the stored cell about how it was stored (e.g., manually entered, calculated, consolidated, etc.) is included. If a number was calculated and stored as a result of a formula, it will always get cleared regardless of the metadata settings. The AllowInput property specifies whether a cell can be written to, and therefore if a formula stored a number, there would be information if the cell was calculated. If AllowInput is set to True, and a new number is typed over the same cell, it is stored as manually entered instead of a calculated cell.

## Formula Execution for Statutory Consolidation Sequence

The consolidation process is run for a hierarchy of Entities. See [Consolidation](#).

## Formula Level Controls

In addition to the Cube-level settings described above, VB.NET If statements are often used in a formula to have that formula execute only for certain types of Data Units. The most common usage is to contain checks for Base-Level Entities and local currencies. The following If statement will cause the formula to execute only for Base-Level Entities and the Entity's default currency. This is often used when the results of the calculation in the Base-Level Entity is intended to be consolidated to Parent-Level Entities.

```
If ((Not api.Entity.HasChildren()) And (api.Cons.IsLocalCurrencyForEntity())) Then  
End If
```

## Writing Formulas

The following section provide information on writing formulas.

# Writing Dynamic Cell Calculations

When writing a formula for a Dynamic Calc Account, Flow, or User Defined Member in either a Member Formula or a Business Rule (via `FinanceFunctionType.GetDataCell`), the goal is to return an amount for a single data cell. In that case, the system knows the full 18 Dimensions of the data cell it needs to display. Therefore, use any of the api methods that refer to a specific Account, Flow, Intercompany, User Defined Dimension, etc.

For example, a user selected a Cube View to view some numbers. The full 18 Dimensions for each cell the Cube View needs to display is determined that it needs to run a custom formula and it initializes the api, so it knows about all 18 Dimensions. The Member Formula then displays an amount on the Cube View.

## Dynamic Calc Formula Examples

### Return a Constant

The simplest Dynamic Calc formula is to return a constant (i.e., the same number regardless of the intersection). For example, select an Account (or a Flow or User Defined Member) in the Dimension Library, set the Formula Type property to DynamicCalc, and then enter the following line in the Member's Formula property for the Default Scenario Type and Default Time Member. After typing the code, press the Compile button on the Formula Editor's toolbar to make sure the VB.NET syntax is correct, press OK to close the dialogs, and then click Save.

```
Return 123.4
```

If a Cube View is used to display the result of the above calculation for the Dynamic Calc account for any numeric intersection (any Entity, Scenario, Time, UD, etc. for View set to YTD), the Cube View will always display 123.4. If the account's value is displayed for any text intersection (any Entity, Scenario, Time, UD, etc. for View set to Annotation), the Cube View will show an empty cell because the formula returned a numeric value, not a text value. To make the formula work for both numeric and text View Members, use an If statement to check which type of View Member is currently being requested. Then return a text value surrounded by double quotes when the View Member is an annotation type. After saving, run the Cube View using multiple View Members (YTD, Periodic, Annotation, Assumptions, etc.). The corresponding cell will display either of the two constants specified.

```
objViewMember As ViewMember = ViewMember.GetItem(api.Pov.View.MemberId)
If objViewMember.IsAnnotationType Then
    Return "My first OneStream Member Formula"
Else
    Return 123.4
End If
```

## Return Types for Dynamic Calc Formulas

As shown in the example above, the return value for a numeric data cell can be a number such as a constant or a decimal variable, and the return type for a text intersection can be a text value such as words within double quotes or a String variable. However, when those types are returned, the underlying engine converts them to a DataCell or a DataCellEx object automatically.

Therefore, if a DataCell or DataCellEx object was already contained by calling `api.Data.GetDataCell`, then it is recommended to return the full object instead of just the number contained within the DataCell object (i.e., which could be accessed using `objDataCell.CellAmount`). In order to specify status such as whether the cell is NoData, return a DataCell or DataCellEx and use its `CellStatus` property for those types of settings.

A DataCell object is a wrapper for a DataCellPk object that defines the cell's 18 Dimensional intersection, a decimal accessed using the `CellAmount` property to store the number, and a `CellStatus` containing other information about the cell such as `NoData` and `Invalid Status`.

A DataCellEx object is a wrapper for a DataCell object and a text property called `DataCellAnnotation` which is used for setting a string for an Annotation type View Member. It also contains some additional properties for `CurrencyId` and `AccountTypId` filled in and can be ignored when creating a DataCellEx object in a Dynamic Calc Member Formula.

The following example accomplishes the exact same result as the example above, except this uses DataCell and DataCellEx objects to illustrate what to do if a return value containing cell status is needed.

```
Dim objViewMember As ViewMember = ViewMember.GetItem(api.Pov.View.MemberId)
If objViewMember.IsAnnotationType Then
    Dim objDataCellEx As DataCellEx = New DataCellEx()
    objDataCellEx.DataCell.CellStatus = New DataCellStatus(True)
    objDataCellEx.DataCellAnnotation = "My first OneStream Member Formula"
    Return objDataCellEx
Else
    Dim objDataCell As DataCell = New DataCell()
    objDataCell.CellStatus = New DataCellStatus(True)
    objDataCell.CellAmount = 123.5
    Return objDataCell
End If
```

## Using Math in Dynamic Calc Formulas

OneStream's `api.Data.GetDataCell` function supports a powerful script parser that allows math equations to be written and operate on one or more data cells or constant amounts to calculate the values for a new data cell. For example, if a Dynamic Calc account needs to display Cash plus AcctsRec increased by 10%, this one-line Member Formula can do it.

```
Return api.Data.GetDataCell("(A#Cash + A#AcctsRec) * 1.10")
```

## About the Financial Model

---

Notice the `api.Data.GetDataCell` function accepts one string Parameter in double quotes which represents a Member Script equation. Each operand of the equation (e.g., `A#Cash`) takes the unspecified Dimensions using the data cell currently being calculated. Therefore, every operand points to a specific data cell identified by using an 18 Dimension intersection.

If the current data cell being calculated is:

`"E#CT:C#USD:S#Actual:#2013M1:V#YTD:A#NewAccount:F#None:0#Top:I#None:U1#None, ..."`,

then the first operand is:

`"E#CT:C#USD:S#Actual:#2013M1:V#YTD:A#Cash:F#None:0#Top:I#None:U1#None, ..."`

even though only “`A#Cash`” was specified.

When writing Dynamic Calc formulas, specify any or all the 18 Dimensions if necessary.  
(e.g., “`A#Cash:U1#AllProducts + A#AcctsRec:U1#None`”)

## Division in Dynamic Calc Formulas

Using Member Script equations in `api.Data.GetDataCell` as shown above is powerful for Dynamic Calc formulas, but there are occasions when all the math cannot be performed reliably in one line of script. For example, use this formula to divide by a data cell when the data cell being used as the denominator is zero or NoData (i.e., a number was never entered). Since dividing by zero results in infinity and is an invalid operation for computers, the formula needs some extra checking. To illustrate additional concepts, here are three different examples of performing division in a Dynamic Calc formula.

This first example uses the division operator (/) to calculate a data cell from one account divided by a data cell from another account. If the denominator (`A#AcctsRec`) is zero or NoData, it will automatically return a very large number (e.g., `9999999999999999999.0`) as the result. This is because dividing by zero in mathematics results in infinity, and the large number to approximate infinity which allows subsequent functions or math operators that refer to the result to continue to be processed is used.

```
Return api.Data.GetDataCell("A#Cash / A#AcctsRec")
```

Although an extremely large number is the best mathematical approximation for infinity, it is typically not what administrators want to display in their financial system when source numbers are not available. A Divide function that produces a NoData cell if either the numerator or the denominator is NoData is available.

```
Return api.Data.GetDataCell("Divide(A#Cash, A#AcctsRec)")
```

## About the Financial Model

---

The built-in Divide function is typically used when performing division in Dynamic Calc formulas. However, for completeness and to provide some insight about how to create more complex formulas, the following is an example of how to implement a formula that performs safe division. Notice that the Member Formula is performing division using two DataCell objects to create a resulting DataCell (Return numeratorDataCell / denominatorDataCell). This is a powerful capability that allows any type of math to be performed using any number of DataCell objects.

```
Dim numeratorDataCell As DataCell = api.Data.GetDataCell("A#Cash")
Dim denominatorDataCell As DataCell = api.Data.GetDataCell("A#AcctsRec")
If ((Not numeratorDataCell Is Nothing) And (Not denominatorDataCell Is Nothing)) Then
    If ((Not numeratorDataCell.CellStatus.IsNoData) And _
        (Not denominatorDataCell.CellStatus.IsNoData) And _
        (denominatorDataCell.CellAmount <> 0.0)) Then
        Return numeratorDataCell / denominatorDataCell
    End If
End If
Return Nothing
```

**TIP:** When using VB.NET, use the underscore character at the end of a line in order to continue a statement on the next line. For example, refer to the underscore after And in the code above. This was done here because the full statement did not fit on one line in this document. However, in the Member Formula editor, the If statement would be kept on one long line.

## Days Sales Outstanding

Days Sales Outstanding (DSO) is a common formula that is a required calculation for many applications. Consequently, OneStream has provided a pre-built function to encapsulate the logic required for this function.

```
Return api.Functions.GetDSODataCell("AcctsRec", "Sales")
```

The example below demonstrates a possible customized version of the DSO calculation.

```
Dim numDaysSum as Integer = 0
Dim currTimeId as Integer = api.Pov.Time.MemberPk.MemberId
Dim acctsRec as decimal = api.Data.GetDataCell("A#AcctsRec - A#SaleTax").CellAmount
If (acctsRec > 0.0) Then
    Dim salesSum as Decimal = 0.0
    Dim numPeriodsInYear = api.Time.GetNumPeriodsInYearForSameFrequency(currTimeId)
    For (numPeriodsToSubtract As Integer = 0 To numPeriodsInYear)
        Dim timeId as Integer
        If numPeriodsToSubtract = 0 Then
            timeId = currTimeId
        Else
            timeId = api.Time.AddTimePeriods(-1 * numPeriodsToSubtract, True)
        End If
        Dim timeName As String = api.Time.GetNameFromId(timeId)
        Dim numDaysInTimePeriod As Integer = api.Time.GetNumDaysInTimePeriod(timeId)
```

## About the Financial Model

---

```
Dim MemberscriptBldr = New MemberscriptBuilder("A#Sales:V#Periodic").SetTime(timeName)
Dim Memberscript As String = MemberscriptBldr.GetMemberscript()
Dim salesForTimePeriod as Decimal = api.Data.GetDataCell(Memberscript).CellAmount

If (salesForTimePeriod + salesSum >= acctsRec) Then
    Dim ratio As Decimal = (acctsRec - salesSum) / salesForTimePeriod
    numDaysSum = numDaysSum + (ratio * numDaysInTimePeriod)
    'We are done
    Exit For
Else
    numDaysSum = numDaysSum + numDaysInTimePeriod
    salesSum = salesSum + salesForTimePeriod
End If
Next
End If
Return api.Data.CreateDataCellObject(numDaysSum, False, False)
```

## Dynamic Simulation of Consolidation: GetEntityAggregationDataCell

Use a function called GetEntityAggregationDataCell for pseudo/approximate consolidation of a data cell with a Cube View when requested for display. The intention is not to produce a value that would tie to a formally consolidated number if custom Business Rules for ownership or translation are in play for such a financial model, but this on-the-fly dynamic value is presented to the user instantly as a convenience, typically during data entry.

This function employs standard currency translation using the Entity in the cell's POV's local currency as the source and a parent Entity's target and standard Percent Consolidation on the Relationship Properties to calculate Share. It does not take intercompany elimination into account unless that value was already consolidated and stored.

```
api.Functions.GetEntityAggregationDataCell(memberScript as string,
Optional useStoredAmountsWhereCalcStatusisOk as Boolean, Optional
fxRateTypeRevenueExpOverride as String, Optional
fxRuleTypeRevenueExpOverride as FxRuleType, Optional
fxRateTypeAssetLiabOverride as String, Optional
fxRuleTypeAssetLiabOverride as FxRuleType)
```

## About the Financial Model

---

When `useStoredAmountsWhereCalcStatusIsOK` is set to `True` (the default setting), the algorithm checks if `CalcStatus` is `OK` for the intersection and uses numbers that are already calculated, translated, ownership share calculated, eliminated, consolidated and stored. This setting is also useful when displaying multiple translated currencies for consolidated parent Entities dynamically. If the parent Entity's Local amount is `OK` and fully consolidated, the only dynamic calculation would be the last translation step.

The last four optional settings let you specify alternate named FX Rate Types (e.g. "AverageRate") and FX Rule Types (i.e. `FxRuleType.Direct` or `FxRuleType.Periodic`) to perform what-if simulations.

If any of these four are specified, `useStoredAmountsWhereCalcStatusIsOK` is ignored and is treated as `False` because consolidated amounts would not have been stored using the alternate FX rates.

Examples when used with a Dynamic Calc member (e.g. UD8):

```
Return api.Functions.GetEntityAggregationDataCell("A#NetIncome:S#Budget:UD8#None")
Return api.Functions.GetEntityAggregationDataCell("UD8#None", True, "AverageRate", FxRuleType.Periodic)
Return api.Functions.GetEntityAggregationDataCell("UD8#None", True, "AverageRate", FxRuleType.Periodic,
    "HistoricalRate", FxRuleType.Direct)
```

Example use of `GetEntityAggregationDataCell` to enter inputs and instantly see aggregated results upon saving a Form:

Create a UD8 Member named `EntityAggregation` and set its Formula Type to Dynamic Calc. Use this as the UD8 member's Formula:

```
Return api.Functions.GetEntityAggregationDataCell("UD8#None")
```

## Create a Cube View

Set Cube View POV tab to have Scenario set to Budget or whatever is desired on the Form and UD8 Member of `EntityAggregation`.

Define rows that display a hierarchy of Entities.

Define columns that display a few key Accounts. Include a base-level Account that supports input for both UD8#`EntityAggregation` and for input. E.g. A#Sales, A#Sales:UD8#None:Name("Sales Input")

Change Cube View settings for General Settings / Common with Can Modify Data and Can Calculate set to `True`.

Associate this Cube View with a Form Template which is assigned to a Workflow Profile.

## About the Financial Model

---

The user enters a number on the Form in the Account that accepts input and clicks Save. The user will see the dynamically aggregated Entity results without having to run a Consolidation.

## Dynamic Calc Member Formula

This formula returns the name and description of the Entity's UD1 Default Member. Other UD8 formulas can be created for the Entity's UD2 Default, etc. After adding the UD8 Member, the Entity's Default UD1 Member name can be displayed using XFGetCell or in a Cube View by accessing the cell: E#MyEntity:V#Annotation:U8#EntityDefaultUD2Name.

This assumes that the UD8 Member with the formula is named EntityDefaultUD2Name. If XFGetCell is being used in Excel, use None or any valid Member for all the other Dimensions.

```
'Display the Member name of the entity's DefaultUD1 Member using the Annotation View Members.  
If api.View.IsAnnotationType(api.Pov.View.MemberId) Then  
    Dim text As String = String.Empty  
    Dim udId As Integer = api.Entity.GetDefaultUDMemberId(api.Pov.Entity.MemberId, DimType.UD1.Id)  
    If udId <> DimConstants.Unknown Then  
        Dim udMember As Member = api.Members.GetMember(DimType.UD1.Id, udId)  
        If Not udMember Is Nothing Then  
            text = udMember.NameAndDescription  
        End If  
    End If  
    Return text  
End If  
'If this is a numeric View Member (e.g., Periodic, YTD), display the number from the U8#None Member.  
Return api.Data.GetDataCell("U8#None")
```

## Annotations on Dynamic Calc Members

To support the ability to calculate text or pull it from an external system to display in a Cube View, DynamicCalc or DynamicCalcTextInput formulas can be used. If the user wants the formula to calculate the annotations, use a DynamicCalc formula to display text such as Pass and Fail.

The DynamicCalcTextInput Formula Type works the same as a DynamicCalc formula, but it allows users to input annotations on Cube View cells without having to use the Data Attachment Dialog. When this formula is used, the user can make annotations on Dynamic Calc Members following the same method as a non-calculated Member.

## Writing Stored Calculations

When writing a Member Formula or a Business Rule for a Stored Calculation, the new calculated numbers are being determined to store for that Cube, Entity, Parent, Cons, Scenario, and Time combination (i.e., a Data Unit).

Return is never seen in a Member Formula for Formula Pass. Instead of being returned, many numbers are being calculated and stored. When running a Calculation, Translation, or Consolidation, the Member Formula calls for an entire Data Unit. It does not tell with which Account, Flow, or User Defined the numbers are being saved, this is the responsibility of the user. Initially, this may be confusing because Member Formulas are often written in an account's Formula property, and administrators believe it will only allow that specific Member Formula to write to that specific account. However, putting a Member Formula in an account's Formula property is only for organizational purposes. When it calls that formula, it is currently calculating a Data Unit and it will initialize the api with only the Data Unit Dimensions.

## Basic Stored Formula Examples

The formula examples in this section demonstrate how to calculate basic stored values driven by formulas consisting of OneStream Member Script expressions.

### Copy Data from another Account

The following formula would be implemented as a Member Formula on the Sales1 account. It is executed as part of the DUCS during the Formula Pass that was specified in the account's Formula Type setting.

```
api.Data.Calculate("A#Sales1 = A#Sales2")
```

Stored Formula passes use Data Buffer math, not the Data Cell math that occurs for the single cell Dynamic Calc formulas. Stored Formulas are multi-Dimensional. For example, the formula is executed for an entire Data Unit (e.g., Location1 Entity, USD Consolidation Member, Actual Scenario, January 2013 time period). That Data Unit is a portion of a Cube where the UD1 Dimension could contain 1,000 products to keep track of sales by product. Therefore, the data for the Sales2 account could contain a separate number for every Product (i.e., UD1), or if the Location1 Entity only sells some of the products, there might be 200 numbers for Sales2 and the other 800 products for Sales2 are NoData. That set of 200 numbers is called a Data Buffer. Data Buffers can get much larger and more complicated when multiple Dimensions are used for detailed analyses. However, since the same concepts still apply, it is easier to think about a smaller set of Dimensions as in this example.

The formula "A#Sales1 = A#Sales2" is equivalent to saying, "Take the 200 numbers stored in the Sales2 Data Buffer and copy them to a new Data Buffer, but change the account to Sales1, and then store the new Sales1 Data Buffer in the database." That one-line formula calculated and stored an additional 200 numbers that did not exist before the formula was executed.

## About the Financial Model

---

The formula below reads the Data Buffer for the Sales2 account (200 numbers) and then adds 50.0 to each of those 200 numbers to create a new Data Buffer that also contains 200 numbers. The account for each of the 200 numbers in the new Data Buffer is changed to Sales1 and it is then stored in the database.

```
api.Data.Calculate("A#Sales1 = A#Sales2 + 50.0")
```

The newly modified formula below uses three accounts. That formula reads the Data Buffer for the Sales2 account (200 numbers) and then reads the Data Buffer for the Sales3 account. For example, the Sales3 account contains 100 numbers broken out by product in UD1, and 25 of those 100 numbers use the same UD1 Members as some of the numbers from Sales2. The other 75 Sales3 numbers are for other products not used by the Sales2 account. OneStream automatically combines the numbers from Sales2 and Sales3 and adds the Sales2 and Sales3 numbers that share a common intersection and also adds the additional non-common intersections. The result is a new Data Buffer containing 275 numbers stored in the database for the Sales1 account.

```
api.Data.Calculate("A#Sales1 = A#Sales2 + A#Sales3")
```

## Copy Data from Another Scenario

The following formula would be implemented as a Member Formula on the Forecast Scenario. It copies all the data from the Actual Scenario and stores the results in the Forecast Scenario. The If statement in this example causes the data to be copied only for Base-Level Entities and each Entity's local currency. This is because the example application wants the numbers for Parent Entities and foreign currencies to be determined using OneStream's Consolidation and Translation algorithms, not by copying directly from another Scenario. Limit the formula to only copy data for certain Dimension Members by adding specific Members in the formula's Member Script. For example, if the Forecast Scenario needs to copy only the Import Members, add O#Import to both sides of the equation. If the Forecast Scenario needs to start with the sum of the Actual Scenario's Import plus Forms plus AdjInput data and copy that into the Forecast Scenario's O#Import Member, then use "S#Forecast:O#Import = S#Actual:O#Top". When writing Stored Formulas, any Data Unit Dimension not explicitly specified uses the Dimension Member for the Data Unit currently being calculated. For any Account, Flow, Origin, IC, or User Defined Dimensions not explicitly specified, OneStream will use #All which is the syntax that represents all existing data for that Dimension.

```
If ((Not api.Entity.HasChildren()) And (api.Cons.IsLocalCurrencyForEntity())) Then  
    api.Data.Calculate("S#Forecast = S#Actual")  
End If
```

To copy data from another Cube or Scenario that uses different Dimensionality, an example of the Scenario formula would be as follows:

## About the Financial Model

---

```
'Convert dimensionality
Dim destinationInfo As ExpressionDestinationInfo = api.Data.GetExpressionDestinationInfo("")
Dim sourceDataBuffer As DataBuffer = api.Data.GetDataBuffer(DataApiScriptMethodType.Calculate,
"Cb#AnotherCube:S#AnotherScenario",destinationInfo)
Dim convertedDataBuffer As DataBuffer = api.Data.ConvertDataBufferExtendedMembers("AnotherCube",
"AnotherScenario", sourceDataBuffer)
api.Data.SetDataBuffer(convertedDataBuffer, destinationInfo)
```

To drill down on this formula, use the following example in the Scenario's Formula for Calculation Drill Down setting:

```
If api.Pov.Cube.Name.XFEqualsIgnoreCase("TheDestCube") Then
    Dim result As New DrillDownFormulaResult()
    result.Explanation = "Pseudo-formula: Cb#TheDestCube:S#TheDestScenario=Cb#AnotherCube:
S#AnotherScenario"

    result.SourceDataCells.Add("Cb#AnotherCube:S#AnotherScenario")
    Return result
End If
Return Nothing
```

## Out-Of-Balance

The following formula would be implemented as a Member Formula on the Balance account. It stores the difference of two other accounts.

```
If ((Not api.Entity.HasChildren()) And (api.Cons.IsLocalCurrencyForEntity())) Then
    api.Data.Calculate("A#Balance = A#2899 - A#5999")
End If
```

## CTA Account Formula Examples

This is essentially the same formula as Out-Of-Balance, but only runs on translated Data Units.

```
If ((Not api.Entity.HasChildren()) And (api.Cons.IsForeignCurrencyForEntity())) Then
    api.Data.Calculate("A#CTA = A#2899 - A#5999")
End if
```

## Reading a Specific Data Cell in a Stored Formula

Stored Formulas are executed for an entire Data Unit, so unlike Dynamic Calc formulas, there is no context about a specific View, Account, Flow, Origin, IC, or User Defined Member within the Data Unit. Therefore, to read the value for a Data Cell using `api.Data.GetDataCell`, a Member needs to be explicitly specified for all non-Data Unit Dimensions. This is different than Dynamic Calc Member Formulas where the default setting for every Dimension Member comes from the Data Cell currently being displayed.

## About the Financial Model

---

Therefore, the following is incorrect when trying to read a specific Data Cell inside a Stored Formula:

```
Dim objDataCell As DataCell = api.Data.GetDataCell("A#Cash")
```

Instead, all non-Data Unit Dimensions need to be specified. If one or more of those Dimensions needs to be based on the other data stored in the Data Unit, then use Eval (for details on Eval see Advanced Stored Formulas using Eval below). Otherwise, the formula will look like this:

```
Dim objDataCell As DataCell = api.Data.GetDataCell("V#YTD:A#Cash:F#None:O#Import:I#None  
:U1#None:U2#None:U3#None:U4#None:U5#None:U6#None:U7#None:U8#None")
```

The syntax above is accurate but creating that long string for many Dimensions is tedious and error prone especially when using functions to determine what the Member names should be, and then concatenating multiple strings. Instead it is recommended to use the MemberscriptBuilder class when creating Member Scripts.

```
Dim MemberscriptBldr = New MemberscriptBuilder("V#YTD:A#Cash")  
MemberscriptBldr.SetFlow("None").SetOrigin("Import").SetIC("None").SetAllUDsToNone()  
Dim Memberscript As String = MemberscriptBldr.GetMemberscript()  
Dim objDataCell As DataCell = api.Data.GetDataCell(Memberscript)
```

## Advanced Stored Formulas using Eval

As described in the examples above, OneStream's Data Buffer math is extremely powerful and can process hundreds or thousands of numbers with just one simple equation. Without Data Buffer math or an equivalent scripting capability, a large multi-Dimensional financial application would not be feasible because every intersection would need to be considered separately. There is a consequence when processing data using Data Buffers instead of individual Data Cells. Additional capabilities are needed when wanting to perform math differently based on the individual data cell amounts.

Fortunately, OneStream accommodates that pattern using Eval. When implementing `api.Data.Calculate` functions, Eval has an advanced capability that provides the ability to get at the individual Data Cells in any Data Unit created while processing an `api.Data.Calculate` script. It even allows `Eval()` to be wrapped around a subset of the formula's math in order to evaluate the Data Buffer that was just created by running that math.

As an example for Eval, start with this formula:

```
api.Data.Calculate("A#Sales1 = A#Sales2")
```

The Sales2 numbers need to be copied to Sales1 for "green" Products. In this fictitious example, there is a special tax situation for green products and the sales numbers for those products need to be isolated into the special Sales1 account. The application uses the UD1 Member's `Text1` property to keep track of which products are green.

## About the Financial Model

---

The first thought might be to do something like the following (incorrect):

```
Dim ud1Id As Integer = api.Pov.UD1.MemberId
Dim text1 As String = api.UD1.Text(ud1Id, 1)
If (text1.Equals("green", StringComparison.InvariantCultureIgnoreCase)) Then
    api.Data.Calculate("A#Sales1 = A#Sales2")
End If
```

However, this would not work because Stored Formulas are executed for an entire Data Unit. A Data Unit represents all data for a Cube, Scenario, Entity, Parent, Cons, and Time Member. Since there is no single product (i.e., UD1 Member) for the Data Unit currently being calculated, the first line above does not make sense. A Data Unit cannot be asked what the UD1 MemberId is because a Data Unit has data for multiple UD1 Members (200 different products in the Sales2 example).

The solution is using Eval to evaluate the individual Data Cells in a Data Buffer. Put the Eval keyword around any portion of the api.Data.Calculate function including math statements. After OneStream reads or calculates the DataBuffer defined within the Eval statement, it executes the Eval function to give the opportunity to filter the list of Data Cells in the Data Buffer, or to completely change the list of Data Cells in the Data Buffer. After the Eval function is completed, OneStream uses the modified Data Buffer to perform the remaining part of the api.Data.Calculate function.

The example formula needs to be modified by adding the Eval keyword around the A#Sales2 Data Buffer, a helper function, typically the name OnEvalDataBuffer, needs to be implemented allowing an inspection, filter, and/or change to the Data Cells in the Data Buffer. The helper function Loops over each of the Sales2 Data Cells (200 in this example). If the Data Cell's UD1 Text1 setting says green, add that Data Cell to a new list of result cells. Otherwise, ignore the Data Cell causing it to be skipped. The result is a new modified Data Buffer containing only the Data Cells for green products (i.e., fewer than 200 Data Cells).

```
api.Data.Calculate("A#Sales1 = Eval(A#Sales2)", AddressOf OnEvalDataBuffer)

Private Sub OnEvalDataBuffer(ByVal api As FinanceRulesApi, ByVal evalName As String, _
    ByVal eventArgs As EvalDataBufferEventArgs)
    'Filter to list of Sales2 source numbers to only include numbers
    'for "green" Products using each dataCell's UD1 Text1 setting.
    'The final list of resultCells is what will be assigned to Sales1 by api.Data.Calculate.
    Dim resultCells As New Dictionary(Of DataBufferCellPk, DataBufferCell)
    For Each sourceCell As DataBufferCell In eventArgs.DataBuffer1.DataBufferCells.Values
        If (Not sourceCell.CellStatus.IsNoData) Then
            Dim ud1Id As Integer = sourceCell.DataBufferCellPk.UD1Id
            Dim text1 As String = api.UD1.Text(ud1Id, 1)
            If (text1.Equals("green", StringComparison.InvariantCultureIgnoreCase)) Then
                'Add this dataCell to the new list.
                resultCells(sourceCell.DataBufferCellPk) = sourceCell
            End If
        End If
    Next
```

```
'Assign the new list of DataCells to the result.  
eventArgs.DataBufferResult.DataBufferCells = resultCells  
End Sub
```

**NOTE:** If using Eval for NoData and ZeroCells, refer to the Remove Functions in Formulas section for alternative performance enhancing solutions.

## Advanced Stored Formulas using GetDataBuffer and SetDataBuffer

Most of the advanced stored formulas that need to process multiple Data Cells should use Eval. Eval allows the user to get at the individual Data Cells in any Data Unit while processing the Member Script in an api.Data.Calculate function. However, in some rare occasions, an appropriate Member Script may not be defined for the api.Data.Calculate function because multiple Data Cells that seem completely unrelated to each other are being processed and none of the Dimension Members are constant.

For those rare situations, use the GetDataBuffer and SetDataBuffer functions directly. GetDataBuffer and SetDataBuffer are more fundamental than Eval. They are part of the internal implementation of the Eval functionality. They allow the user to read some numbers using a Member Script, process or modify each cell in the result, and then save the changes.

In the following example, the UD2 and UD3 Dimensions are being used to analyze data based on each UD1 Member's default settings for UD2 and UD3. All data is initially loaded to the U2#Input:U3#Input Members, but that loaded data needs to be copied to the U2#DefaultUD1:U3#DefaultUD1 Members. GetDataBuffer needs to be used in order to read the loaded data because the destination UD2 and UD3 Members can be different for every Data Cell based on its UD1 Member's settings. Loop over each Data Cell and use the UD1 Member to get its default UD2 and default UD3 settings. Then, change the UD2 and UD3 Member IDs for the Data Cell in the Data Buffer. Finally, after Looping, call SetDataBuffer to save the new numbers.

When using api.Data.Calculate functions with or without Eval, it is important to know to which Member a formula is being attached. For example, if the formula starts with api.Data.Calculate ("A#Sales1 = ..."), put the formula in the Sales1 account Member's Formula setting. However, the formula in this example is not writing to a specific Member. Every Data Cell being saved is possibly written to a different UD2 and UD3 Member.

## About the Financial Model

---

Technically speaking, the formula can be put in any Member's Formula property even a seemingly unrelated Member. If the Formula Pass is set correctly, the formula executes before any other dependent formulas. Assigning stored formulas to Members is for organizational purposes only. The Member does not restrict what the formula can do. However, if a formula is attached to an unrelated Member, it will make the application difficult to maintain and understand. Therefore, decide to attach a formula like this to the Scenario's Member which means this formula needs to be processed before most other formula passes. If there are numerous Scenarios, this formula should be put in a Business Rule file, and the Business Rule file should be added to the Cube(s). This can be done under

Application Tab|Cube|Cubes.

See the completed formula using GetDataBuffer and SetDataBuffer below:

```
'Copy all "U2#Input:U3#Input" numbers for this dataUnit to the corresponding UD1 default Members for
UD2 and UD3.
Dim destinationInfo As ExpressionDestinationInfo = api.Data.GetExpressionDestinationInfo("")
Dim sourceDataBuffer As DataBuffer = api.Data.GetDataBuffer(DataApiScriptMethodType.Calculate, _
    "U2#Input:U3#Input", destinationInfo)
If Not sourceDataBuffer Is Nothing Then
    Dim resultDataBuffer As DataBuffer = New DataBuffer()
    For Each cell As DataBufferCell In sourceDataBuffer.DataBufferCells.Values
        If (Not cell.CellStatus.IsNoData) Then
            Dim ud1Id As Integer = cell.DataBufferCellPk.UD1Id
            cell.DataBufferCellPk.UD2Id = api.UD1.GetDefaultUDMemberId(ud1Id, DimType.UD2.Id)
            cell.DataBufferCellPk.UD3Id = api.UD1.GetDefaultUDMemberId(ud1Id, DimType.UD3.Id)
            resultDataBuffer.SetCell(api.DbConnApp.SI, cell)
        End If
    Next
    api.Data.SetDataBuffer(resultDataBuffer, destinationInfo)
End If
```

## Comparing Two DataBuffers Using Eval2

Eval2 is the same as Eval except two Members Scripts are specified to define two DataBuffers. When finished, the OnEvalDataBuffer function can compare all the numbers in the two DataBuffers. The example provided evaluates two separate Flow Members to see if they contain the same value.

```
'Use Eval2 to compare the numbers in 2 DataBuffers and store a value in
F#USDOVERRIDE_Check for each pair of numbers that don't match.
If ((api.Cons.IsLocalCurrencyForEntity()) And (Not api.Entity.HasChildren())) Then
    api.Data.Calculate("V#YTD:F#USDOVERRIDE_Check:0#Forms =
        Eval2(V#YTD:F#Local_Change_Validation:0#Top,
        V#YTD:F#USDOVERRIDE_Change_Validation:0#Top)", AddressOf OnEvalDataBuffer)
End If

Private Sub OnEvalDataBuffer(ByVal api As FinanceRulesApi, ByVal evalName As String,
```

## About the Financial Model

---

```
    ByVal eventArgs As EvalDataBufferEventArgs)

'This function compares the numbers in 2 DataBuffers and returns a new DataBuffer
that has a value for each pair of numbers that don't match.
    eventArgs.DataBufferResult.DataBufferCells.Clear()
    If Not eventArgs.DataBuffer1 Is Nothing And Not eventArgs.DataBuffer2
        Is Nothing Then

'For each cell in DataBuffer1, try to find a number for the same intersection in
        DataBuffer2.
    For Each cell1 As DataBufferCell In eventArgs.DataBuffer1.DataBufferCells.Values
        If (Not cell1.CellStatus.IsNoData) Then

            Dim cell2 As DataBufferCell = eventArgs.DataBuffer2.GetCell(api.SI,
                cell1.DataBufferCellPk)
            If Not cell2 Is Nothing Then
                If (cell1.CellAmount <> cell2.CellAmount) Then
                    'Since the numbers don't match, add a cell to the result
                    DataBuffer.
                    Dim resultCell As New DataBufferCell(cell1)
                    resultCell.CellAmount = 1.0
                    eventArgs.DataBufferResult.SetCell(api.SI, resultCell,
                        False)
                End If
            Else
                'A number exists in the 1st DataBuffer, but it doesn't exist in
                the 2nd
                DataBuffer.
                'Therefore, add a cell to the result DataBuffer.
                Dim resultCell As New DataBufferCell(cell1)
                resultCell.CellAmount = 1.0
                eventArgs.DataBufferResult.SetCell(api.SI, resultCell, False)
            End If
        End If
    Next
    'Now, for each cell in DataBuffer2, try to find a number for the same intersection in
        DataBuffer1
    (the opposite of the loop above).
    'Create a cell in the result DataBuffer for each DataBuffer2 cell that doesn't exist
        in DataBuffer1.
    For Each cell2 As DataBufferCell In eventArgs.DataBuffer2.DataBufferCells.Values
        If (Not cell2.CellStatus.IsNoData) Then
            Dim cell1 As DataBufferCell = eventArgs.DataBuffer1.GetCell(api.SI,
                cell2.DataBufferCellPk)
            If cell1 Is Nothing Then

                'A number exists in the 2nd DataBuffer, but it doesn't
```

## About the Financial Model

---

```
        exist in the
        1st DataBuffer.
        'Therefore, add a cell to the result DataBuffer.
        Dim resultCell As New DataBufferCell(cell12)
        resultCell.CellAmount = 1.0
        eventArgs.DataBufferResult.SetCell(api.SI, resultCell, False)
    End If
End If
Next
End If
End Sub
```

## Referencing a Business Rule from a Member Formula or Business Rule

Finance Business Rules from Member Formulas or other Business Rules can be called. This is helpful when the same code must be copied to multiple Member Formulas and instead of using the same complicated code, a Public Function with two lines of code written in a Business Rule can be called.

First, create a new Finance Business Rule (in this example, a Finance Business Rule called SharedFinanceFunctions was created) and then set the Contains Global Functions for Formulas property to True. If the Business Rule is only being used to hold Shared Functions, delete most of the content in the Main function. However, a Main function is still needed even if it is empty.

Next, create a Public Function or Sub in the Business Rule. See below for an example. If any edits to the Business Rule impact Calculation Status, assign the Shared Business Rule to the Cube under Application Tab|Cube|Cubes. This is recommended.

Use the Business Rule in a Member Formula by creating an instance of the Business Rule and assigning it to a variable. Then, any of the Business Rule's Public Functions or Sub can be called.

Sample Member Formula Code:

```
Dim sharedFinanceBR As New
OneStream.BusinessRule.Finance.SharedFinanceFunctions.MainClass
Dim myResult As String = sharedFinanceBR.Test(si, api, args)
```

Sample Business Rule:

```
Imports System
Imports System.Data
Imports System.Data.Common
Imports System.IO
Imports System.Collections.Generic
Imports System.Globalization
```

## About the Financial Model

---

```
Imports System.Linq
Imports Microsoft.VisualBasic
Imports System.Windows.Forms
Imports OneStream.Shared.Common
Imports OneStream.Shared.Wcf
Imports OneStream.Shared.Engine
Imports OneStream.Shared.Database
Imports OneStream.Stage.Engine
Imports OneStream.Stage.Database
Imports OneStream.Finance.Engine
Imports OneStream.Finance.Database

Namespace OneStream.BusinessRule.Finance.SharedFinanceFunctions
    Public Class MainClass

        Public Function Main(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As
FinanceRulesApi, ByVal args As FinanceRulesArgs) As Object
            Try
                Return Nothing
            Catch ex As Exception
                Throw ErrorHandler.LogWrite(si, New XFException(si, ex))
            End Try
        End Function

        Public Function Test(ByVal si As SessionInfo, ByVal api As FinanceRulesApi, ByVal args As
FinanceRulesArgs) As String
            Try
                Return "This is the result of my Test function!"
            Catch ex As Exception
                Throw ErrorHandler.LogWrite(si, New XFException(si, ex))
            End Try
        End Function
    End Class
End Namespace
```

## Conditionally Apply Formulas to Entities and Consolidation Members

Conditionally apply formulas to run only when needed. For example, calculate headcount only at Base Entities and Local currency and the consolidation engine will do the rest.

```
If (api.cons.IsLocalCurrencyForEntity And (Not api.entity.HasChildren)) Then
    api.data.calculate("A#10999=A#25999-A#25986")
End If
```

# Focus Consolidation to Calculate Only When Needed

There are up to seven calculation operations per Entity in the Consolidation process. Add Conditional Statements to formulas to limit which Consolidation calculation processes will run for a particular formula.

### Base Entities Only

```
If Not api.entity.HasChildren Then
```

### Local Currency Only

```
If Api.Cons.IsLocalCurrencyforEntity Then
```

### Translated Currency Only

```
If api.Cons.IsForeignCurrencyForEntity Then
```

### Parent-Child Relationships

If api.Cons.IsForeignCurrencyForEntity Then returns True if the current calculated Consolidation Member also depends on the Parent Entity (i.e. OwnerPreAdj, Share, Elimination, OwnerPostAdj, Top). If there are two different Parent Entities for the same Entity, then there are two different sets of numbers stored for those Consolidation Members.

### At Specific Level of Consolidation

```
If (Not api.Entity.HasChildren()) And (api.Pov.Cons.Name.XFEqualsIgnoreCase("Elimination"))  
Then would run if the Entity is a parent and also if the member of the Consolidation dimension  
being processed in the Data Unit is Elimination. Note that this is the preferred function to use  
rather than the formerly supported api.Pov.Cons.ScriptName.
```

# Formulas for Calculation Drill Down

To drill down on calculated Members, a formula must be entered in the Formula for Calculation Drill Down property. This allows drilling to occur on calculated Account, Flow, User Defined or Scenario Members.

Drill down can occur on data cells copied from one Scenario to another via formula or Data Management Sequence. Before displaying the drill results, every cell's Formula for Calculation DrillDown Scenario Property is executed. The result determines whether the Scenario Member will appear as drillable or not. Therefore, use If Statements in the formula to narrow in on the cell's Storage Type and/or the POV Members associated with the data copy, so cells do not appear drillable when they are not. The example below copies data from the Actual Scenario Type to Budget:

## About the Financial Model

---

```
Dim result As New DrillDownFormulaResult()
If args.DrillDownArgs.RequestedDataCell.CellStatus.StorageType = DataCellStorageType.Calculation Then
    'Use this to drill down to data that was copied using a Scenario Formula.
    result.Explanation = "Formula Definition: Actual = Budget"
    result.SourceDataCells.Add("Cb#Houston:E#Houston:S#Actual")
Else If args.DrillDownArgs.RequestedDataCell.CellStatus.StorageType = DataCellStorageType.Input Then
    'Use this to drill down to data that was copied using Data Management.
    'result.Explanation = "Data Management Defintion: Actual = Budget"
    'result.SourceDataCells.Add("Cb#Houston:E#Houston:S#Actual")
End If
Return Result
```

## Global Entity for Driver Storage

A Global Entity can be created to store information that is unrelated to the company's data such as switches, drivers, or values. To do this within the application, create the Entity and set the Is Consolidated property to False. The Consolidation process will then skip this Entity allowing companies to use this for grouping purposes only.

## Avoiding Data Explosion

In some cases, many data intersections unintentionally explode in the Cube with unintended values. This can happen with multi-dimensional data structures.

For example, an application has one Entity, four Accounts, 100 Products (UD1) and 100 Customers (UD2). The user has entered three numbers in the system.

(A#Sales:U1#Prod1:U2#Cust1=22.0, A#Sales:U1#Prod5:U2#Cust1=33.0,  
A#Sales:U1#Prod7:U2#Cust100=55.0).

The following cases explain what happens when writing the following Formulas.

### Case 1: No Explosion

Use the following statement to set the value of a data intersection equal to another or a constant value.

```
api.Data.Calculate("A#Cash1 = A#RestrictedCash")
```

Cash1 Account will equal what was in the Restricted Cash Account. The only UD1 and UD2 Members populated would be None Members because it is not typical to delineate Cash by Product, or Customer.

### Case 2: No Explosion

```
api.Data.Calculate("A#Profit:U2#Cust1 = A#Sales * 1.05")
```

Two of the three numbers are multiplied by 1.05 and copied from the Sales Account to the Profit Account which are only the source Sales numbers for Customer 1.

### Case 3: Some Explosion

```
api.Data.Calculate("A#Profit = A#Sales:U2#Cust1")
```

The user specified a Customer for source data but did not specify which customer to use when writing to the destination. This means answers are written to every base-level member of the Customer dimension, resulting in some data explosion. The Profit account has a number for every Customer (UD2) using the same Product (UD1) member as the source numbers. Profit account will have 300 numbers, but because there were three different source products, there will be some data explosion.

Not specifying a Member for a Dimension is the same as specifying all for a Member. If All is specified on the left only, data explosion occurs. Some level of data explosion occurs if:

- The right side of an equation identifies a specific member or amount.
- The left side of an equation specifies All for one or more dimensions.

### Case 4: Large Scale Explosion When Setting to a Constant

```
api.Data.Calculate("A#Profit = 2.0")
```

Profit will have 1000 numbers stored that are all Products by all Customers, causing large scale explosion. However, this example is overly simple. If all Dimensions are used, trillions of numbers may be stored. A constant is the same as not specifying All for every Dimension on the right side of the equation.

## Writing Formulas to Avoid Data Explosion

If a member is specified for a dimension on the right side of the equation, explicitly specify a value for that Dimension on the left side.

The examples below are the same as Cases 3 and 4 from above, but this time the formulas are written to avoid data explosion.

### Case 3: Specify a target on the left side of the equation

`api.Data.Calculate("A#Profit = A#Sales:U2#Cust1")` would be better written as `api.Data.Calculate("A#Profit:U2#Cust1 = A#Sales:U2#Cust1")` if that is what is intended. If the user intended on copying Cust1's Sales figures to every Customer intersection under the Profit Account, the formula would need to be written as follows:

```
api.Data.Calculate("A#Profit:U2#All = A#Sales:U2#Cust1")
```

### Case 4:

`api.Data.Calculate("A#Profit = 2.0")` will fail since the constant of 2.0 implies All Members from each Dimension. In order for this Formula to execute, it needs to be `api.Data.Calculate("A#Profit:F#All:O#All:I#All:U1#All:U2#All:U3#All:U4#All:U5#All:U6#All:U7#All:U8#All = 2.0")`, although the user may choose to do otherwise.

## Avoiding Data Explosion in Stored Formulas

When writing stored calculations, the Member Script equations in `api.Data.Calculate` end up reading or calculating one or more Data Buffers for the purpose of saving a resulting Data Buffer. As described earlier, thousands of Data Cells could be processed using a seemingly simple one-line formula.

Data Explosion can occur when a formula is inadvertently written to read or calculate a Data Buffer and then copy all Data Cells in that Data Buffer multiple times to every base-level Member of a Dimension using the resulting Data Buffer (often causing hundreds of thousands of new numbers to be saved). We only apply calculation to intersections where data exists. Fortunately, OneStream protects the user from writing formulas that could result in data explosion, but the concepts are important to understand because it is possible to circumvent those protections.

Consider the formula from an earlier example:

```
api.Data.Calculate("A#Sales1 = A#Sales2 + A#Sales3")
```

Now, the formula is changed, so it only copies the sales data for specific customers with UD2 being the Customer Dimension.

```
api.Data.Calculate("A#Sales1:U2#None = A#Sales2:U2#CustomerX + A#Sales3:U2#CustomerY")
```

The above formula will not result in data explosion because there is the same level of detail (i.e., the same Dimensions) specified in the destination as in every source operand. The example now reads the Sales2 data for CustomerX, adds it to the Sales3 data for CustomerY, and saves the results in the Sales1 account and the UD2 None Member.

The following formula is written to cause Data Explosion:

## About the Financial Model

---

```
api.Data.Calculate("A#Sales1 = A#Sales2:U2#CustomerX + A#Sales3:U2#CustomerY")
```

The UD2 Dimension is specified for the source operands which is the right-hand side of the equals sign, but UD2 is not specified for the destination which is the left-hand side of the equals sign.

When processing this formula, OneStream will read the two Data Buffers and add them together correctly as before. However, it then needs to assign the combined Data Buffer to the Sales1 Account. However, the system cannot use U2#CustomerX or U2#CustomerY because Data Cells were created by adding those together and it cannot arbitrarily choose one over the other. The system could also have defaulted to use the U2#None Member, but history suggests that this type of rule is more often written in error, and the customer did not intend the results to be stored in the U2#None Member. From a maintenance perspective, OneStream feels that it is better to explicitly specify U2#None if that is the intended destination Member.

If something like this were to happen, OneStream will provide an error message notifying the user that data explosion will occur when trying to execute the above formula. Otherwise, the formula will copy the source Data Buffer to every base-level UD2 Member because #All is the default setting for each unspecified Member.

To circumvent the error message and force data explosion (please do not do this), explicitly specify U2#All in the destination as shown below. This should be avoided and #All should never be used in Member Scripts for stored formulas. However, the capability is provided for extremely rare circumstances where that functionality was relied upon using an older product. In this case, the consultant carefully analyzed the quantity of data and metadata settings to ensure the data explosion resulted in a manageable number of Data Cells.

**CAUTION:** This causes Data Explosion! Do not ever use #All explicitly in stored Member Formulas.

```
api.Data.Calculate("A#Sales1:U2#All = A#Sales2:U2#CustomerX + A#Sales3:U2#CustomerY")
```

## Key Functions

The list below contains the most commonly used functions; however, this is not the complete list of all available functions. Download the OneStream API Overview Guide and OneStream API Details and Database Documentation from MarketPlace for detailed Business Rule engine background, an API guide and information on each database.

## Account

Name	Function	Description
Account Type	<pre>Public Function GetAccountType (MemberId As Integer) As AccountType  myAccountType = api.Account.GetAccountType (MemberId)</pre>	Retrieves the Account type for the Member.
Get Cell Account Type	GetCellLevelAccountType	Retrieves the Account type of the data cell based on its Account and flow settings.
Get Formula Type	<pre>Public Function GetFormulaType (MemberId As Integer) As FormulaType  myFormulaType = api.Account.GetFormulaType (MemberId)</pre>	Returns the Formula Type if the Account is calculated.
Get Plug Account	<pre>Public Function GetPlugAccount (MemberId As Integer) As Member myMember = api.Account.GetPlugAccount (MemberId)</pre>	Retrieves the plug Account.

## Consolidation

Name	Function	Description
Calculate	<pre>Public Sub Calculate(Formula As String, Optional onBeforeSetDataBuffer As BeforeSetDataBufferDelegate, Optional userState As Object, Optional arg0 As String, Optional arg1 As String, Optional arg2 As String, Optional arg3 As String, Optional arg4 As String, Optional arg5 As String, Optional arg6 As String, Optional arg7 As String)</pre> <pre>api.Data.Calculate(Formula, onBeforeSetDataBuffer, userState, arg0, arg1, arg2, arg3, arg4, arg5, arg6, arg7)</pre>	Executes a calculation for a specifically qualified Point of View.
Execute Default Elimination	<pre>Public Sub ExecuteDefaultElimination()</pre> <pre>api.ExecuteDefaultElimination()</pre>	Puts data into the Elimination Member of the Consolidation Dimension.
Second Pass Eliminations	<p>Boolean argument for use in FinanceFunctionType.Calculate rules</p> <pre>Dim bValue As Boolean = args.CalculateArgs.IsSecondPassEliminationCalc</pre>	Used to calculate Data Units where Entity members are sibling members. Used to ensure source Entities are fully calculated at Eliminations
Execute Default Share	<pre>Public Sub ExecuteDefaultShare()</pre> <pre>api.ExecuteDefaultShare()</pre>	Puts data into the Share Member of the Consolidation Dimension.

## About the Financial Model

---

Name	Function	Description
Execute Default Translation	<pre>Public Sub ExecuteDefaultTranslation (fxRuleTypeRevenueExp As FxRuleType, rateForRevenueExp As Decimal, fxRuleTypeAssetLiab As FxRuleType, rateForAssetLiab As Decimal)  api.ExecuteDefaultTranslation (fxRuleTypeRevenueExp, rateForRevenueExp, fxRuleTypeAssetLiab, rateForAssetLiab) or api.ExecuteDefaultTranslation()</pre>	Calculates translation by bypassing consolidation and Business Rules.
Is Consolidated	<pre>Public Function IsConsolidated(MemberId As Integer) As TriStateBool  myTriStateBool = api.Account.IsConsolidated (MemberId)</pre>	Reveals if the Account is consolidated.
Percent Consolidated	<pre>Public Function PercentConsolidation(Optional EntityId As Integer, Optional ParentId As Integer, Optional varyByScenarioTypeld As Integer, Optional varyByTimeld As Integer) As Decimal  myDecimal = api.Entity.PercentConsolidation (EntityId, ParentId, varyByScenarioTypeld, varyByTimeld)</pre>	The percent that an Entity contributes to a Relationship.
Translated Currency Member	<pre>Public Function IsForeignCurrencyForEntity (Optional EntityId As Integer, Optional consId As Integer) As Boolean  myBoolean = api.Cons.IsForeignCurrencyForEntity (EntityId, consId)</pre>	Returns if the Consolidation Dimension Member is a translated currency Member. Used in translation rules.

## Data

Name	Function	Description
Allocation	Use Journals for allocations	Allocates data across Dimensions (Entities, User Defined Dimensions, Accounts, etc.) with configurable weighting, all through Journals that can be previewed, are generated, are posted and can be unposted.
Convert Data Buffer	api.Data.ConvertDataBuffer	Modifies Dimension Members for the cells in a Data Buffer using mapping.
Convert Data Buffer Extended Members	api.Data.ConvertDataBufferExtendedMembers	Automatically aggregates the data for extended Members in order to create data cells for Parent Members that are Base-Level Members in the destination Dimensions. This is used when copying data from a source Data Buffer created in another Cube or Scenario where one or more Dimensions have been extended.

## About the Financial Model

---

Name	Function	Description
Get Data Buffer (working with Data Units)	<pre>Public Function GetDataBuffer (scriptMethodType As DataApiScriptMethodType, sourceDataBufferScript As String, expressionDestinationInfo As ExpressionDestinationInfo) As DataBuffer  myDataBuffer = api.Data.GetDataBuffer (scriptMethodType, sourceDataBufferScript, expressionDestinationInfo)</pre>	Retrieves a Data Unit's values during a consolidation, calculation, or translation.
Get Data Buffer Using Formula	<pre>Dim myDataBufer As DataBuffer = api.Data.GetDataBufferUsingFormula ("A#Sales-A#Costs")</pre>	Use an entire math expression to calculate a final data buffer.
Get Stored and Dynamically Calculated Values	<pre>Public Function GetDataCell(Formula As String, Optional arg0 As String, Optional arg1 As String, Optional arg2 As String, Optional arg3 As String, Optional arg4 As String, Optional arg5 As String, Optional arg6 As String, Optional arg7 As String) As DataCell  myDataCell = api.Data.GetDataCell (Formula, arg0, arg1, arg2, arg3, arg4, arg5, arg6, arg7)</pre>	Retrieves the data contained in an intersection of the Cube. If this is a dynamically calculated value, the calculation will be run on the fly before returning the value.

## About the Financial Model

---

Name	Function	Description
Has Cell Notes or Data Attachments	api.Data.GetDataCell(Formula, Parameters).CellStatus.HasAttachment	Indicates whether the data cell has notes or file attachments. (Boolean) To retrieve the data attachment, query the appropriate Member of the View Dimension (Annotation, AuditComment, Footnote, VarianceExplanation, Assumptions).
Has No Data	api.Data.GetDataCell(Formula, Parameters).CellStatus.IsNoData	Indicates if the cell has no data. (Boolean)
Has Valid Data	api.Data.GetDataCell (Formula,Parameters). CellStatus.DataIsTimeLogicDerived	Indicates if the data in a cell contains valid data that was derived from time logic. (Boolean)
Get Data Buffer for Custom Share Calculation	<p>Public Function GetDataBufferForCustomShareCalculation (Optional cubeld As Integer, Optional entityId As Integer, Optional ParentId As Integer, Optional scenarioId As Integer, Optional timeld As Integer, Optional viewId As Integer) As DataBuffer</p> <p>myDataBuffer = api.Data. GetDataBufferForCustomShareCalculation (cubeld, entityId, ParentId, scenarioId, timeld, viewId)</p>	Use this function to assist in Custom Consolidations

## About the Financial Model

---

Name	Function	Description
Get Data Buffer for Custom Elim Calculation	<pre>Public Function GetDataBufferForForCustomElimCalculation (Optional includelCNone As Boolean, Optional includelCPartners As Boolean, Optional combineImportFormsAndAdjConsolidatedInto Elim As Boolean, Optional cubeld As Integer, Optional entityId As Integer, Optional ParentId As Integer, Optional scenarioid As Integer, Optional timeld As Integer, Optional viewId As Integer) As DataBuffer  myDataBuffer = api.Data. GetDataBufferForForCustomElimCalculation (includelCNone, includelCPartners, combineImportFormsAndAdjConsolidatedInto Elim, cubeld, entityId, ParentId, scenarioid, timeld, viewId)</pre>	Use this function to assist in Custom Calculations
Rounding	Use the VB.NET function for rounding: Round(expression[, numdecimalplaces])	Controls the level of rounding that is used.
Set and Clear Data	<pre>Public Sub SetDataCell(Memberscript As String, amount As Decimal, isNoData As Boolean)  api.Data.SetDataCell(Memberscript, amount, isNoData)</pre>	Sets data to a certain value.

## About the Financial Model

---

Name	Function	Description
Show Cell Text	Data Attachment Members in the View Dimension: Annotation, Assumptions, AuditComment, Footnote and VarianceExplanation	Use these View Members to display data attachment text within a Cube View, which can be used to edit text or display on Reports.
Variance	$((A-B)/Abs(B))$	Returns a Variance, but does not consider Account Type
BetterWorse Difference	Revenue Accounts: $((A-B/Abs(B)))$ Expense Accounts: $((-1*A)-(-1*B))$	Returns a Variance based on the Account Type.
VariancePercent	$((A-B) / Abs(B)) * 100$	Returns a Variance Percent but does not consider Account Type.
BetterWorse Percent	Revenue Accounts: $((A-B) / Abs(B)) * 100$ Expense Accounts: $(((-1 * A) - (-1 * B))/Abs(B)) * 100$	Returns a Variance Percentage based on the Account Type.
Unbalanced Math: Add, Subtract, Multiply, and Divide	Api.Data.Calculate("A#TargetAccount = AddUnbalanced (A#60000,A#41000:O#Top, O#Top)")	Required to perform math using two Data Buffers where additional dimensionality must be specified for the second Data Buffer.

## Entity

Name	Function	Description
Default Currency	Public Function GetLocalCurrency(Optional EntityId As Integer) As Currency  myCurrency = api.Entity.GetLocalCurrency(EntityId)	Retrieves the assigned Currency for the Entity or Parent.
Is Descendant	Public Function IsDescendent(dimPk As DimPk, ancestorMemberId As Integer, descendantMemberId As Integer, dimDisplayOptions As DimDisplayOptions) As Boolean  myBoolean = api.Members.IsDescendent(dimPk, ancestorMemberId, descendantMemberId, dimDisplayOptions)	Returns if the Member is a Descendant of another Member. (Boolean)
Is Intercompany	Public Function IsIC(Optional EntityId As Integer) As Boolean  myBoolean = api.Entity.IsIC(EntityId)	Returns if the Entity or Account Member is an intercompany Member.

## Flow

Name	Function	Description
Switch Sign	Public Function SwitchSign(MemberId As Integer) As Boolean  myBoolean = api.Flow.SwitchSign(MemberId)	Flow Dimension only. Responds as to whether credits are switched to debits for the specified Member for Revenue / Expense Accounts. (Boolean)

## About the Financial Model

---

Name	Function	Description
Switch Type	Public Function SwitchType(MemberId As Integer) As Boolean  myBoolean = api.Flow.SwitchType (MemberId)	Flow Dimension only. Responds as to whether Account types are switched for the current or specified Member. This can drive translating this Member by a different FX Rate Type. (Boolean)

## FX

Name	Function	Description
Currency Type	Public Function GetDefaultCurrencyId(Optional Cuboid As Integer) As Integer  myInteger = api.Cubes.GetDefaultCurrencyId(Cuboid) or Public Function GetCurrency(currencyName As String) As Currency  myCurrency = api.Cons.GetCurrency(currencyName)	Retrieves the currency type for the Cube or the Consolidation Dimension Member.
Current Exchange Rate	Public Function GetStoredFxRate(fxRateType As FxRateType, Optional timeld As Integer, Optional sourceCurrencyId As Integer, Optional destCurrencyId As Integer) As FxRate  myFxRate = api.FxRates.GetStoredFxRate (fxRateType, timeld, sourceCurrencyId, destCurrencyId)	Retrieves the current exchange rate for the specified Entity.

## About the Financial Model

---

Name	Function	Description
Exchange Rate Calculated	<pre>Public Function GetCalculatedFxRate(fxRateType As FxRateType, timeld As Integer) As Decimal  myDecimal = api.FxRates.GetCalculatedFxRate(fxRateType, timeld)</pre>	Calculates the exchange rate from the default currency to another.
Get FX Rate Type for Asset / Liability Accounts	<pre>Public Function GetFxRateTypeForAssetLiability(Optional Cubeld As Integer, Optional Scenarioid As Integer) As FxRateType  myFxRateType = api.FxRates.GetFxRateTypeForAssetLiability(Cubeld, Scenarioid)</pre>	Retrieves the default Rate Type for Asset and Liability Accounts in this Cube or Scenario (overrides Cube value).
Get FX Rate Type for Revenue/Expense Accounts	<pre>Public Function GetFxRateTypeForRevenueExp(Optional Cubeld As Integer, Optional Scenarioid As Integer) As FxRateType  myFxRateType = api.FxRates.GetFxRateTypeForRevenueExp(Cubeld, Scenarioid)</pre>	Retrieves the default Rate Type for Revenue and Expense Accounts in this Cube or Scenario (overrides Cube value).

## About the Financial Model

---

Name	Function	Description
Get FX Rule Type for Asset / Liability Accounts	<p>Public Function GetFxRuleTypeForAssetLiability (Optional Cubeld As Integer, Optional ScenarioId As Integer) As FxRuleType</p> <p>myFxRuleType = api.FxRates.GetFxRuleTypeForAssetLiability(Cubeld, ScenarioId)</p>	Retrieves the default translation Rule Type for Asset and Liability Accounts in this Cube or Scenario (overrides Cube value).
Get FX Rule Type for Revenue/Expense Accounts	<p>Public Function GetFxRuleTypeForRevenueExp (Optional Cubeld As Integer, Optional ScenarioId As Integer) As FxRuleType</p> <p>myFxRuleType = api.FxRates.GetFxRuleTypeForRevenueExp(Cubeld, ScenarioId)</p>	Retrieves the default translation Rule Type for Revenue and Expense Accounts in this Cube or Scenario (overrides Cube value).
Translate	<p>Public Sub Translate(sourceDataBufferScript As String, destDataBufferScript As String, fxRuleType As FxRuleType, rate As Decimal)</p> <p>api.Data.Translate(sourceDataBufferScript, destDataBufferScript, fxRuleType, rate)</p>	Performs the translation method assigned to the Cube or Scenario (Periodic or Direct).

## Journals

Name	Function	Description
Journal Postings Allowed	<pre>Public Function AllowAdjustments(Optional EntityId As Integer, Optional varyByScenarioTypeld As Integer, Optional varyByTimeld As Integer) As Boolean  myBoolean = api.Entity.AllowAdjustments(EntityId, varyByScenarioTypeld, varyByTimeld)</pre>	Results determine if Journal postings are allowed for the Member.
Journal Postings from Children Allowed	<pre>Public Function AllowAdjustmentsFromChildren (Optional EntityId As Integer, Optional varyByScenarioTypeld As Integer, Optional varyByTimeld As Integer) As Boolean  myBoolean = api.Entity.AllowAdjustmentsFromChildren(EntityId, varyByScenarioTypeld, varyByTimeld)</pre>	Results determine if Journal postings from children are allowed for this Member.

## Member

Name	Function	Description
Base Members	HasChildren = false, e.g. A#Root.Children (HasChildren=False)	Determines if the Member is a base Member.
Get Member ID	<pre>api.Members.GetMember(dimTypeld, MemberName).MemberPk.MemberID or  api.POVDimension.Memberpk.Memberid</pre>	Retrieves the name for the selected Member.

## About the Financial Model

---

Name	Function	Description
Get Member Name	api.Members.GetMember(dimTypeld, MemberName).Name	Retrieves the name for the selected Member.
Member Lists	See section of documentation on Creating Member Lists	Retrieves the Members from a named list stored in a Business Rule.
Member Name	api.POView.AccountDim, api.POView.EntityDim, etc	Retrieves the Member name.
Member Name or ID	Get Member name from Member ID: Public Function GetMember(dimTypeld As Integer, MemberId As Integer) As Member  myMember = api.Members.GetMember(dimTypeld, MemberId)  Get Member ID from Member name: Public Function GetMember(dimTypeld As Integer, MemberName As String) As Member  myMember = api.Members.GetMember(dimTypeld, MemberName)	Retrieves the Member for the specified ID number or name.
Top Member	api.Account.GetTopMemberForDimType (AccountMemberId, dimTypeForTopMember)	Retrieves the top Member of the selected Dimension.

## About the Financial Model

---

Name	Function	Description
Is Base	<p>Public Function IsBase(dimPk As DimPk, ancestorMemberId As Integer, baseMemberId As Integer, Optional dimDisplayOptions As DimDisplayOptions) As Boolean</p> <p>myBoolean = api.Members.IsBase(dimPk, ancestorMemberId, baseMemberId, dimDisplayOptions)</p>	Determines whether the POV Member is the Base of a defined Member.
Get First Common Parent	<p>Public Function GetFirstCommonParent(dimPk As DimPk, topMostMemberId As Integer, MemberIdA As Integer, MemberIdB As Integer, Optional dimDisplayOptions As DimDisplayOptions) As Member</p> <p>myMember = api.Members.GetFirstCommonParent(dimPk, topMostMemberId, MemberIdA, MemberIdB, dimDisplayOptions)</p>	Returns the first common Parent between multiple Members.

## Scenario

Name	Function	Description
Scenario Consolidation View	<p>Public Function GetConsolidationView(Optional Scenarioid As Integer) As ViewMember</p> <p>myViewMember = api.Scenario.GetConsolidationView(Scenarioid)</p>	Determines if the Scenario's Consolidation View is set to YTD or Periodic.

## About the Financial Model

---

Name	Function	Description
Scenario Default View	<pre>Public Function GetDefaultView (Optional Scenarioid As Integer) As ViewMember  myViewMember = api.Scenario.GetDefaultView (Scenarioid)</pre>	Retrieves the Scenario's Default View.
Scenario Input Frequency	<pre>Public Function GetInputFrequency (Optional Scenarioid As Integer) As Frequency  myFrequency = api.Scenario.GetInputFrequency (Scenarioid)</pre>	Retrieves the Scenario's Input Frequency.

## Status

Name	Function	Description
Calc Status	<pre>Public Function GetCalcStatus(Optional Cuboid As Integer, Optional EntityId As Integer, Optional ParentId As Integer, Optional consId As Integer, Optional Scenarioid As Integer, Optional timId As Integer) As CalcStatus  myCalcStatus = api.CalcStatus.GetCalcStatus(Cuboid, EntityId, ParentId, consId, Scenarioid, timId)</pre>	Retrieves the calculation status for the data intersection.
Impact Calc Status	<pre>Public Sub ImpactCalcStatus(Optional Cuboid As Integer, Optional EntityId As Integer, Optional ParentId As Integer, Optional consId As Integer, Optional Scenarioid As Integer, Optional timId As Integer)  api.CalcStatus.ImpactCalcStatus(Cuboid, EntityId, ParentId, consId, Scenarioid, timId)</pre>	Changes the status of the specified Data Unit to impact.

## Time

Name	Function	Description
MemberId for the Month to which a Week belongs	Dim timeIdForMonth As Integer = BRApi.Finance.Time.ConvertIdToClosestIdUsingAnotherFrequency (si, timeIdForWeek, Frequency.Monthly)	Weekly applications only: This determines to which month a specific week belongs. Used within a Finance Business Rule.

## Functions

Name	Function	Description
Days Sales Outstanding	Dim cell As DataCell = api.Functions.GetDSODataCell (acctsReceivableMember, salesMember)	Calculates Days Sales Outstanding (see below).
Dynamic Simulation of Consolidation	api.Functions.GetEntityAggregationDataCell (memberScript as string, Optional useStoredAmountsWhereCalcStatusIsOK as Boolean, Optional fxRateTypeRevenueExpOverride as String, Optional fxRuleTypeRevenueExpOverride as String, Optional fxRateTypeAssetLiabOverride as String, Optional fxRuleTypeAssetLiabOverride as String)	Pseudo/approximate consolidation of a data cell (see Dynamic Simulation of Consolidation)

Name	Function	Description
GetStage or GetCustom	Download the OneStream API Overview Guide from MarketPlace for examples:  GetStageBlendTextUsingCurrentPOV GetStageBlendText GetStageBlendNumberUsingCurrentPOV GetStageBlendNumber GetStageBlendDataTableUsingCurrentPOV GetStageBlendDataTable GetCustomBlendDataTableUsingCurrentPOV GetCustomBlendDataTable	Several functions that retrieve text and calculate values by reading values from the OneStream Stage or a custom relational table.

## Examples of Key Functions in Use

### Calculate Data

The most common function used is `api.data.calculate`, which sets the value of one or more values (left side of Formula) equal to another (right side). A final argument (optional) can be added as True or False as to whether to use a data cell Storage Type of Durable. Durable data will not be cleared automatically when a Data Unit is re-calculated. It can only be cleared by calling `api.Data.ClearCalculatedData` with the `clearDurableCalculatedData` boolean property set to True.

```
api.Data.Calculate(formula, isDurableCalculatedData)
```

For example, the following Stat Account is used to calculate Total Cost of Sales for three months and is not set to be a Durable storage method (optional argument):

```
api.data.calculate("A#TOT_COS_LAST3:V#YTD = A#TOT_COS:V#Periodic + A#TOT_COS:T#POVPrior1:V#Periodic +  
A#TOT_COS:  
T#POVPrior2:V#Periodic", False)
```

An alternative to this overloaded function is to provide Member Filters (all optional) that can be used to filter the results before saving them to the target to affect fewer intersections, such as only to be applied to certain Flow members:

## About the Financial Model

---

```
api.Data.Calculate(formula, accountFilter, flowFilter, originFilter, icFilter, ud1Filter, ud2Filter,  
ud3Filter, ud4Filter,  
ud5Filter, ud6Filter, ud7Filter, ud8Filter, onEvalDataBuffer, userState, isDurableCalculatedData)
```

Another alternative allows the use of arguments to be applied:

```
api.Data.Calculate(Formula, onBeforeSetDataBuffer, userState, arg0, arg1, arg2, arg3, arg4, arg5, arg6,  
arg7)
```

## Clear Calculated Data

Clearing calculated data is performed when calculated Members need to be reset to NoData. This will result in the Data Units requiring a calculation. Note that the final argument must be true to clear any data with a Storage Type of Durable. See Calculate Data for how to set a calculation to store data as Durable.

```
api.Data.ClearCalculatedData(dataBufferScript, clearCalculateData, clearTranslationData,  
clearConsolidationData,  
clearDurableCalculatedData)
```

```
api.Data.ClearCalculatedData("A#[cash deposits]:ud1#Production", true, true, true, false)
```

## DataBuffer

When setting a value equal to another value, the item on the left side of the expression is the value being set, and the item on the right side is the value being queried or calculated to set the left side. Example: F#BeginBalance = F#EndingBalance.T#POVPrior1 would set the beginning balance in the Flow Dimension to the prior period's ending balance. In a Business Rule, the DestinationInfo is the left side of the equation while a GetDataBuffer is the right side of the equation.

```
Dim destinationInfo As ExpressionDestinationInfo = api.Data.GetExpressionDestinationInfo  
    ("A#EBITDA:UD1#Tires")  
Dim sales As DataBuffer = api.Data.GetDataBuffer("A#Sales:UD1#Tires", destinationInfo)  
Dim operatingExpenses As DataBuffer = api.Data.GetDataBuffer("A#OperatingExpenses:UD1#Tires",  
    destinationInfo)  
Dim ebitda As DataBuffer = (sales - operatingExpenses)  
api.Data.SetDataBuffer(ebitda, destinationInfo)
```

This translates to the following equation:

```
A#EBITDA:UD1#Tires=A#Sales:UD1#Tires - A#OperatingExpenses:UD1#Tires
```

# GetDataBufferUsingFormula

Use an entire math expression to calculate a final data buffer.

Api.Data.GetDataBufferUsingFormula can perform the same data buffer math as api.Data.Calculate, but the result is assigned to a variable where api.Data.Calculate saves the calculated data.

## Example

```
Dim myDataBuffer As DataBuffer = api.Data.GetDataBufferUsingFormula ("A#Sales-A#Costs")
```

Loop over the contents of myDataBuffer to conditionally change each data cell.

## Formula Variables

There is additional capability to using Formula Variables to achieve the same level of flexibility and integration as using Evals. After creating a data buffer variable, name it as a Formula Variable and reference it inside api.Data.Calculate or other calls to api.Data.GetDataBufferUsingFormula. This provides flexibility and can improve performance because the Data Buffer is calculated once and the variable is re-used multiple times.

## Example

```
Dim myDataBuffer As DataBuffer = api.Data.GetDataBufferUsingFormula ("A#Sales-A#Costs")
api.Data.FormulaVariables.SetDataBufferVariable("myDataBuffer",myDataBuffer,False)
api.Data.Calculate("A#Profit=A#5000 + $myDataBuffer")
```

Use api.Data.FormulaVariables.SetDataBufferVariable to name the data buffer. Pass in any name followed by the data buffer variable. Enter a True/False value for Uses Indexes to Optimize Repeat Filtering. Using True will re-use the same data buffer using FilterMembers and improve performance. After naming the data buffer, use a dollar sign and the name when referencing it in a script.

### FilterMembers

Use this inside of an api.Data.Calculate or api.Data.GetDataBufferUsingFormula script.

## Example

```
Dim myDataBuffer As DataBuffer
Api.Data.GetDataBufferUsingFormula("FilterMembers(A#All,A#6000,[A#[3600].Base])")
```

## About the Financial Model

---

Change a data buffer and only include numbers for the specified Dimensions. The first parameter is the starting data buffer. This can be a variable name or an entire math equation in parentheses. There can be as many parameters as needed to specify Member Filters and different Member Filters can be used for multiple Dimension types. The resulting filtered data buffer will only contain numbers that match the Members in the filters.

### **RemoveMembers**

This uses the same syntax as FilterMembers, but it takes the data cells away for the specified Members instead of keeping them.

### **Retrieving Member Names from IDs**

This retrieves the Member names from the IDs when looping over the cells in a data buffer.

## **Example**

```
For Each sourceCell As DataBufferCell In myDataBuffer.DataBufferCells.Values
    Dim accountName As String = sourceCell.DataBufferCellPk.GetAccountName(api)
    Dim ud1Name As String = sourceCell.DataBufferCellPk.GetUD1Name(api)
    If ud1Name = "None" Then BRApi.ErrorLog.LogError(si, "UD1"& ud1Name, Nothing)
Next
```

### **Logging Contents of a Data Buffer**

Log the contents of a Data Buffer to the Error Log when writing Business Rules in order to make corrections and troubleshoot along the way.

### **Example**

```
myDataBuffer.LogDataBuffer(api,"MyDataBufferOutput",1000)
```

The third Parameters (1000) indicates the maximum number of cells to include in the log and displays what is in the data buffer.

### **Example api**

```
api.LogError(XFErrorLevel.Information, "MyDataBuffer As a CSV String For Excel",
myDataBuffer.GetCSVString(api, False, 1000))
```

The false Parameter specifies whether to include Member IDs in the output. Member names are always included.

## **Remove Functions in Formulas**

### **Remove Zeros**

The RemoveZeros function evaluates a source data buffer and removes all cells where the amount is zero whether it is flagged as a NoData cell or not. Therefore, it is not necessary to use both RemoveZeros and RemoveNoData together since RemoveZeros handles both situations. This can be identified on individual cells as the Cell Amount under Cell Status.

## About the Financial Model

---

This function is important to use for performance purposes. Use the RemoveZeros function in calculations where there is a substantial amount of No Data or 0.00 cells in Data Units. This can be determined by looking at the Data Unit Statistics when right-clicking on a cell in a Cube View.

General	
Total Number of Stored Records	1637
NODATA Status	
Number of NODATA Cells	4
Number of Zero Cells	13
Number of Real Cells	1617
Number of Derived Cells	3

Example using api.Data.Calculate:

```
'Run for base Entities and local currency only
If ((Not api.Entity.HasChildren()) And (api.Cons.IsLocalCurrencyForEntity())) Then
    api.Data.Calculate("A#CashCalc=RemoveZeros(A#10000)")
End If
```

Example using GetDataBufferUsingFormula:

```
If ((Not api.Entity.HasChildren()) And (api.Cons.IsLocalCurrencyForEntity())) Then

    'Get Data Buffer by using GetDataBufferUsingFormula
    Dim dataBufferExample As DataBuffer = api.Data.GetDataBufferUsingFormula("RemoveZeros(A#100000)")
    'Set Data Buffer Variable to pass into api.Data.Calculate formula. Can be used for multiple instances of api.Data.Calculate
    'Create a unique name to name the Data Buffer as a Formula Variable
    api.Data.FormulaVariables.SetDataBufferVariable("dataBufferExample", dataBufferExample, False)
    'Pass Formula Variable into api.Data.Calculate using a $
    api.Data.Calculate("A#CashCalc = $dataBufferExample")

End If
```

### Remove NoData

The RemoveNoData function evaluates a source data buffer and removes data cells that have a cell amount of NoData.

This function is important to use for performance purposes. Use the RemoveNoData function in calculations where there is a substantial number of cells with a Cell Status of NoData in Data Units. This can be determined by looking at the Data Unit Statistics when right-clicking on a cell in a Cube View.

## About the Financial Model

---

General	
Total Number of Stored Records	1637
NODATA Status	
Number of NODATA Cells	4
Number of Zero Cells	13
Number of Real Cells	1617
Number of Derived Cells	3

Example using api.Data.Calculate:

```
'Run for base Entities and local currency only
If ((Not api.Entity.HasChildren()) And (api.Cons.IsLocalCurrencyForEntity())) Then
    api.Data.Calculate("A#CashCalc=RemoveNoData(A#10000)")
End If
```

Example using GetDataBufferUsingFormula:

```
If ((Not api.Entity.HasChildren()) And (api.Cons.IsLocalCurrencyForEntity())) Then

    'Get Data Buffer by using GetDataBufferUsingFormula to do the math
    Dim dataBufferExample As DataBuffer = api.Data.GetDataBufferUsingFormula("RemoveNoData(A#10000)")
    'Set Data Buffer Variable to pass into api.Data.Calculate formula. Can be used for multiple instances of api.Data.Calculate
    'Create a unique name to name the Data Buffer as a Formula Variable
    api.Data.FormulaVariables.SetDataBufferVariable("dataBufferExample", dataBufferExample, False)
    'Pass variable into api.Data.Calculate using a $
    api.Data.Calculate("A#CashCalc = $dataBufferExample")

End If
```

Performance Note: In the calculation performance testing for a single Data Unit, the use of RemoveZeros and RemoveNoData using api.Data.Calculate and/or GetDataBufferUsingFormula rendered a significant performance advantage. This is testing the before and after calculation time of a single formula change for a Data Unit. Times may vary from Data Unit to Data Unit and Application to Application. However, the use of RemoveZeros and RemoveNoData is highly recommended in formulas where Cube and Dimensionality designs lend to sparse data models. This is not limited to sparse data models as this can be identified in dense data models as well.

## Dynamic Calc Using GetDataCell

A Dynamic Calc returns an Account value on demand without storing the data in the Cube. These Members cannot be referenced by other Members in calculating their values. Dynamic Calcs are best used in Members required for reporting only. One ideal use is the GetDataCell function in ratio-style statistical Accounts (e.g. Current Ratio).

```
Return api.Data.GetDataCell("Divide(A#[CurrentAssets], A#[CurrentLiabilities])")
```

## Dynamic Filters and Aliases

Create Dynamic Member Filters to assign to rows or columns within Cube Views. A common example of a Variance column that subtracts one Member from another is shown below. Adding the :Name("Membername") option at the end gives the ability to assign a display name to the header of the Cube View. The double quotes in the Name() function are optional.

```
Cube View Filter: GetDataCell("Variance(T#POV,T#Prior12)":Name("Variance"))
```

## Error Traps

```
Try
    if api.POView.Cons.name =api.POView.GetEntityCurrency() then
        api.Data.Calculate("A#Cash1=A#[Restricted Cash] + 70000")
    End If
catch ex as exception
    api.LogError(ex)
End Try
```

## Math Functions

Use the math functions built into VB.NET.

Examples:

Absolute Value - math.abs()

Rounding - math.round()

Several other supported math functions and examples are listed here:

<http://msdn.microsoft.com/en-us/library/thc0a116%28v=VS.90%29.aspx>

## Member List Functions

When a Cube filter is run and the filter value does not match a Member, the system looks in all Business Rules for the MemberListProperty property to find a matching Member list name. Users should create MemberLists in Business Rules that are not assigned to Cubes because the entire Business Rule will be run if the system finds a matching Member list name. Reference MemberLists like this: E#Root.PartnerList.

The following example first declares the two MemberListHeader names of Sample Member List and PartnerList. It then addresses each MemberList. Sample Member List shows a way to get the children of an Entity. E#Texas.[Sample Member List] is just another way of saying E#Texas.Children. In the PartnerList example, E#Root.PartnerList will generate a list that includes Paris and Nice.

```
Case Is = FinanceFunctionType.MemberListHeaders
' Additional logic that defines the names of the two custom Member Lists
    Dim myMemberListHeaders = New List(Of MemberListHeader)()
    myMemberListHeaders.Add(new MemberListHeader("Sample Member List"))
    myMemberListHeaders.Add(new MemberListHeader("PartnerList"))
Return myMemberListHeaders

Case Is = FinanceFunctionType.MemberList
    If args.MemberListArgs.MemberListName = "Sample Member List" Then
        Dim myMemberListHeader = new MemberListHeader(args.MemberListArgs.MemberListName)
        Dim myMembers = new List(Of Member)()
        Dim myMemberList = New MemberList(myMemberListHeader,
            myMembers)
        myMembers.AddRange(api.Members.GetChildren(args.MemberListArgs.DimPk,
            args.MemberListArgs.TopMember.MemberPk.MemberId, args.MemberListArgs.DimDisplayOptions))
        Return myMemberList
    Else If args.MemberListArgs.MemberListName = "PartnerList" Then
        Dim myMemberListHeader = new MemberListHeader(args.MemberListArgs.MemberListName)
        Dim myMembers = new List(Of Member)()
        Dim myMemberList = New MemberList(myMemberListHeader, myMembers)
        myMembers.AddRange(api.Members.GetBaseMembers(args.MemberListArgs.DimPk,
            api.Members.GetMember(args.MemberListArgs.DimPk.dimtypeid, "Paris").MemberPk.MemberId,
            args.MemberListArgs.DimDisplayOptions))
        myMembers.AddRange(api.Members.GetBaseMembers(args.MemberListArgs.DimPk,
            api.Members.GetMember(args.MemberListArgs.DimPk.dimtypeid, "Nice").MemberPk.MemberId,
            args.MemberListArgs.DimDisplayOptions))
        Return myMemberList
    End If
```

## POV Object

In Business Rules, only Data Unit Dimensions are valid in the POV object, not Account or User Defined Members, so the code below only works in Business Rules:

## About the Financial Model

---

```
Dim AcctID as Integer=api.POView.Account.MemberPk.MemberID
```

```
Dim AcctID as Integer=api.POView.GetDataCellPk.AccountID
```

## Retrieve Time Varying Property of Relationship's % Consolidated Entity Attribute

```
'Get Entity ID, Parent Entity ID, Scenario ID, Scenario Type ID and Time ID
dim EntityID as integer= api.POView.Entity.Memberpk.Memberid
dim ParentID as integer= api.POView.Parent.Memberpk.Memberid
dim ScenarioID as integer=api.POView.Scenario.Memberpk.Memberid
dim ScenarioTypeID as integer = api.Scenario.GetScenarioType(ScenarioID).id
dim TimeID as integer = api.Time.GetIdFromName("2013M1")
'Get time varying node % consolidation property value for current Scenario POV and 2013M1
dim myDecimal as decimal= api.Entity.PercentConsolidation(EntityId, ParentId, ScenarioTypeId, TimeId)
'Log the time varying property
api.logerror(xfErrorLevel.Information,myDecimal & " " & api.POView.Entity.name & "." & api.POView.Parent.name)
```

**TIP:** API.POView.Parent object is only relevant for relationship level Consolidation Members

## SetDataCell

Use this to set a value equal to another value. All Members on the right side must be specified.

```
api.Data.SetDataCell(MemberScript, amount, isNoData)
api.Data.SetDataCell("A#[Restricted Cash]:0#Forms:F#None:IC#None:U1#None:U2#None:U3#None:U4#None:
U5#None:U6#None:U7#None:
U8#None", 50, False)
```

## Translate

The api.data.translate function is the same as api.data.calculate, but aggregates AdjInput data into the AdjConsolidated Member.

```
'GetParentCurrency only returns a value when running a translate.
If api.Parameters.FunctionType = FinanceFunctionType.Translate Then if api.POView.Cons.Name =
    api.POView.GetParentCurrency.Name Then
        api.Data.Translate("A#[Restricted Cash]=A#[Restricted Cash]:C#[USD]*10")end if
End If
```

# Unbalanced Math

The Unbalanced math functions are required when performing math with two Data Buffers where the second Data Buffer needs to specify additional dimensionality. The term Unbalanced is used because the script for the second Data Buffer can represent a different set of Dimensions from the other Data Buffer in the api.Data.Calculate text. These functions prevent data explosion.

In the examples below, the first two parameters represent the first and second Data Buffers on which to perform the function. The third parameter represents the Members to use from the second Data Buffer when performing math with every intersection in the first Data Buffer. The math favors the intersections in the first Data Buffer without creating additional intersections.

It is key that the dimensionality of the Target (left side of the equation) matches the dimensionality of the first data buffer on the right side of the equation (argument 1).

### AddUnbalanced

```
api.Data.Calculate("A#TargetAccount = AddUnbalanced(A#OperatingSales, A#DriverAccount:U2#Global,  
U2#Global)")
```

### SubtractUnbalanced

```
api.Data.Calculate("A#TargetAccount = SubtractUnbalanced(A#OperatingSales, A#DriverAccount:U2#Global,  
U2#Global)")
```

### DivideUnbalanced

```
api.Data.Calculate("A#TargetAccount = DivideUnbalanced (A#OperatingSales, A#DriverAccount:U2#Global,  
U2#Global)")
```

### MultiplyUnbalanced

```
api.Data.Calculate("A#TargetAccount = MultiplyUnbalanced (A#OperatingSales, A#DriverAccount:U2#Global,  
U2#Global)")
```

Consider this example. A#OperatingSales has 100 stored records in January for a single Entity.

Because A#OperatingSales has a total of 100 stored values, A#TargetAccount will end up with 100 stored numbers and the amounts would be the values from A#OperatingSales plus/minus/multiplied/divided by whatever was found at A#DriverAccount:U2#Global for each of those 100 intersections.

This means that if there was no data in A#OperatingSales:U2#Widgets, then even though the UD2 Dimension is unspecified in the target and in the first Data Buffer expression, no record would be created, hence avoiding data explosion. The most common use case would be applying a driver for some of the Dimensions.

### ConvertUnbalanced

This function is related to the Unbalanced Math functions (see Unbalanced Math later in this section) and used to convert a data buffer so that it is balanced with an api.Data.Calculate script where unbalanced math does not apply. This is necessary when using an api.Data.GetDataBufferUsingFormula to calculate a data buffer where the script was not balanced to match a formula in another script where a data buffer variable needs to be used.

In the example below, a myDataBuffer was created to have data for all stored accounts, but the subsequent api.Data.Calculate scripts expects each operand to use a specific account. The ConvertUnbalanced function filters the data buffer to only include the specified account name and it also converts the data buffer to make it balanced and consistent with the destination. The same data buffer can be re-used multiple times.

#### Example

```
Dim myDataBuffer As DataBuffer = api.Data.GetDataBufferUsingFormula("A#All")
api.Data.FormulaVariables.SetDataBufferVariable("myDataBuffer ", myDataBuffer, True)
api.Data.Calculate("A#6050 = ConvertUnbalanced($myDataBuffer, A#6000) +
ConvertUnbalanced($myDataBuffer, A#3000)")
```

## Finance Function Types

Use Finance Function Types, except for Custom Calculate, to run logic in certain calculation sequences during the calculation sequence while a Function Type is in process. There are thirteen Function Types:

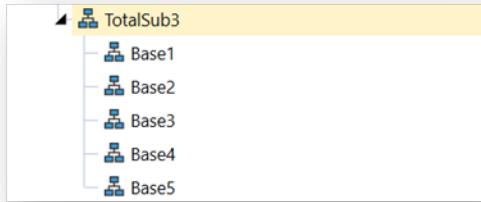
### Calculate

Additional logic during calculation of Entity, Consolidation Scenario and Time. This sets the value of one or more values (left side of Formula) equal to another (right side). It then executes a calculation for a specifically qualified Point of View. This is the most common function used.

There are situations where the Entity being processed must access another Entity's data. In situations involving pulling Consolidation dimension Elimination results from other Entities as siblings, the multi-thread processing of the calculations requires an additional function to ensure the calculations are complete. Below is an example of this sibling relationship. The Entities Base1, Base2, etc. are siblings that would be calculated simultaneously during a consolidation:

## About the Financial Model

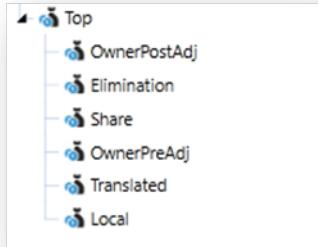
---



For this purpose, the Calculate Finance Function Type supports the argument `IsSecondPassEliminationCalc`.

```
Case Is = FinanceFunctionType.Calculate  
If (api.Pov.Cons.MemberId = DimConstants.ConsElimination) And (args.CalculateArgs.IsSecondPassEliminationCalc) Then
```

During the Calculate process, this function allows Business Rules to execute after the Sibling Entities have calculated results to the Consolidation Elimination member. Below is a reference to the Consolidation dimension:



Once all the sibling Entity members are calculated to Elimination, the Business Rules within the `IsSecondPassEliminationCalc` will be executed.

### Translate

Additional logic that uses custom translation.

### FXRate

Custom logic used to determine Foreign Exchange rates for any intersection.

### Consolidate Share

Additional logic used during the custom calculation of the Share Member.

### Consolidate Elimination

Additional logic used during the custom calculation of the Elimination Member.

### Custom Calculate

A CustomCalculate Finance Function Type can be used in order to execute a single year custom calculation via a Dashboard Parameter Component Server Task Action. This is considered a partial calculation and does not store the calculated data or run the calculation during a consolidation. Running a custom calculation from a Dashboard will impact calculation status for the affected data unit even if the data does not change. See Parameter Components in "Presenting Data With Books, Cube Views and Dashboards" on page 547 for more details on how to assign this type of Finance Rule to a Dashboard. See Data Management for details on creating this type of Data Management Step. See Parameter Components for more information on passing arguments to a Custom Calculate function.

Example:

```
Select Case api.FunctionType
  Case Is = FinanceFunctionType.CustomCalculate
    If args.CustomCalculateArgs.FunctionName.XFEqualsIgnoreCase("FunctionName") Then
      Me.CalcTest(si, globals, api, args)
      api.Data.Calculate("A#TFS2903 = A#10000 + A#69000")
    End If
  End Select
```

### Calculation Drill Down Member Formula

Provides custom drill down results.

### Conditional Input Rule

Conditional Input Rules make data cells read-only. While the settings for this can be done directly on the Cube, using a Conditional Input Business Rule offers more flexibility and still allows the use of the Cube settings. This rule can return the following: ConditionalInputResultType.Default, ConditionalInputResultType.NoInput, ConditionalInputResultType.NoInputAllowCellDetail, and ConditionalInputResultType.NoCellDetailAllowInput.

The following Business Rule example will make all cells for the Account 6000 read-only. This should be added to a Business Rule attached to a Cube.

```
Case Is = FinanceFunctionType.ConditionalInput
  If api.Pov.Account.Name.XFEqualsIgnoreCase("6000") Then
    Return ConditionalInputResultType.NoInput
  End If
  Return ConditionalInputResultType.Default
```

### Confirmation Rule

Special logic that runs with Confirmation Rules.

## About the Financial Model

---

### Data Cell

Named GetDataCell calculations that can be reused such as a Better/Worse calculation in Cube Views.

### Dynamic Calc Account

Special logic to use in Dynamic Calc members.

### Member List and Member List Headers

A custom list of members for use in Cube Views and other areas. See Commonly Used Member Filter Functions in "Cubes" on page 328 for more details on using custom lists in a Cube View.

Select Case api.FunctionType is the expression used when a certain process needs to be isolated and run special logic. See examples below:

```
Select Case api.FunctionType
Case Is = FinanceFunctionType.Calculate
    ' Additional logic to run with every calculation.
    api.Data.Calculate("A#DACash1= A#Cash4 + 1")
    api.Data.Calculate("A#2150:F#Movement = V#Periodic:A#5750:F#None") if api.metaData.Cons.name ="USD"
        then
            api.Data.Calculate("A#Cash1=A#[Restricted Cash]")
End If
Case Is = FinanceFunctionType.Translate
    ' Additional logic to run with every translation. If api.ExecuteDefaultTranslation is not
    included, then the standard translation process will not run.
    api.ExecuteDefaultTranslation
    dim Cubeid as integer = api.Cubes.GetCubeInfo("GolfStream").Cube.Cubeid
dim fxRule as fxRuleType=api.Cubes.GetFxRuleTypeForAssetLiability(CubeId)
api.Data.Translate("A#[Restricted Cash]:C#[USD]","A#[Restricted Cash]",fxRule,.5)
Case Is = FinanceFunctionType.ConsolidateShare then
    ' Additional logic to run when the Share Member of the Consolidation Dimension is being calculated.
    api.ExecuteDefaultShare()
    api.data.calculate("A#[Restricted Cash]=A#[Restricted Cash]*.8")
Case Is = FinanceFunctionType.ConsolidateElimination then
    ' Additional logic to run while the Elimination Member of the Consolidation Dimension is being
    calculated.
    api.ExecuteDefaultElimination
    api.Data.Calculate("A#[IC AR]=A#[IC AR] * 1.1")
Case Is = FinanceFunctionType.MemberListHeaders
    ' Additional logic that defines the names of custom Member Lists
    Dim myMemberListHeaders = New List(Of MemberListHeader)myMemberListHeaders.Add(new MemberListHeader
    ("Sample Member List"))

    Return myMemberListHeaders
Case Is = FinanceFunctionType.MemberList
    ' Additional logic that defines the Members within custom Member List that are included in Business
    Rules.

See the "MemberList Functions" section in the proceeding pages.
'Example: "A#Root.[Sample MemberList]"
If args.MemberListArgs.MemberListName = "Sample Member List" Then Dim myMemberListHeader =
```

## About the Financial Model

---

```
new MemberListHeader(args.MemberListArgs.MemberListName)
    Dim myMembers = new List(Of Member)()
    Dim myMemberList = New MemberList(myMemberListHeader, myMembers)
    myMembers.AddRange(api.Members.GetChildren(
        args.MemberListArgs.DimPk,
        args.MemberListArgs.TopMember.MemberPk.MemberId, args.MemberListArgs.DimDisplayOptions))
    Return myMemberList
End If
End Select
```

### Ultimate Ownership

This function calculates and stores Ultimate Ownership results for every ancestor/child Entity relationship. The current Entity being calculated is the ancestor Entity, and when storing the results, the Members in the Intercompany Dimension are used to represent each descendant Entity. This function assumes the source Direct Ownership numbers are weights and are typically entered by a user (e.g., numbers between 0.0 and 1.0).

C#OwnerPreAdj:O#AdjInput is using an account that does not consolidate and accepts manual entry for adjustments in those Parent/Child relationships. It cascades the user-entered amounts up the Entity tree and multiplies them when a Parent owns part of a Child and that Child owns part of a Grandchild.

sourceDirectOwnershipMemberscript represents the data cells where the user-entered Direct Ownership amounts are found for Parent/Child Entity combinations.

destUltimateOwnershipMemberscript represents the intersection for the resulting calculations. All results are stored using the current Data Unit being calculated.

This function is intended to run only for the Entity's Local Consolidation Member, and it uses IC Members to store the results for each descendant Entity.

Example: Use the following as a Member Formula in the UltimateOwnership account:

```
If api.Cons.IsLocalCurrencyForEntity() Then
    api.Data.CalculateUltimateOwnership("C#OwnerPreAdj:A#DirectOwnership:O#AdjInput",
                                         "A#UltimateOwnership:O#Forms", 1.0)
End If
```

Here are the steps to set up Ownership entry and calculations in an application. This functionality is intended to be used in a separate Control Cube accessed via the main Financial Cube:

## Create Two Accounts

### DirectOwnership

This is for the user to enter Direct Ownership for each Parent/Child Entity combination.

AccountType = BalanceRecurring, IsConsolidated = False, Adjustment Type = Data Entry

## About the Financial Model

---

### **UltimateOwnership**

This stores the calculated results in Parent Entities using IC Members for every descendant Entity.

AccountType = BalanceRecurring, FormulaType = Formula Pass 1, Allow Input = False, IsConsolidated = False, Is IC Account = Conditional

Formula:

```
If api.Cons.IsLocalCurrencyForEntity() Then  
    api.Data.CalculateUltimateOwnership("C#OwnerPreAdj:A#DirectOwnership:0#AdjInput",  
                                         "A#UltimateOwnership:0#Forms", 1.0)  
End If
```

## Create a Cube View for entering Direct Ownership Amounts

### **Cube View POV**

Consolidation = OwnerPreAdj, Origin = AdjInput, Flow, IC and UDs = None

#### **Column**

A#DirectOwnership

#### **Row**

E#Root.TreeDescendants

## Create a Cube View for Running a Consolidation and Viewing the Ultimate Ownership Results

### **Cube View POV**

Consolidation = Local, Account = UltimateOwnership, Origin = Forms, Flow IC and UDs = None

#### **Column**

IC#ICEntities.Base

#### **Row**

E#Root.TreeDescendants

## Consolidation

Consolidation is the process of taking base level Entities and aggregating them up a hierarchy to their Parent. The hierarchy and add efficiencies where applicable, meaning some siblings may be able to run in parallel when they are processed. Calculations and translations are also run during this process.

## About the Financial Model

---

Consolidations can be launched from the Process step in Workflow, Cube Views or Forms. By triggering a consolidation, the existing POV is run for that time period and is consolidated for the respective period along with any of the prior periods for that year. For example, if a consolidation is triggered for June, it will consolidate January through June.

The following chart explains how the Consolidation Dimension works.

---

TOP - Contributes to the Parent Entity's LOCAL Member

---

Sum (SHARE + ELIMINATION + OWNERPOSTADJ)

---

OWNERPOSTADJ (Execute Data Unit Calculation Sequences, FinanceFunctionType = Calculate)

Stores Journal entries that are needed after the Share and Elimination calculations have been performed for an Entity and Parent.

---

ELIMINATION (Execute DUCS, FinanceFunctionType = Calculate)

---

Elimination algorithms (Use default or implement FinanceFunctionType.ConsolidateElimination in a BR)

Start with SHARE Data Unit and eliminate data cells where IC is not None. If a common Parent Entity was reached for both the Entity and the IC Entity, generate an offset amount and a plug Account amount.

---

SHARE (Execute DUCS, FinanceFunctionType = Calculate, only if not using default calc-on-the-fly)

Default Share is calculated on-the-fly

---

Sum (TRANSLATED + OWNERPREADJ) \* % Consolidation

---

Share algorithms (Use default or stored, or implement FinanceFunctionType.ConsolidateShare in a BR)

---

OWNERPREADJ (Execute DUCS, FinanceFunctionType = Calculate)

## About the Financial Model

---

Stores Journal entries that are needed before the Share and Elimination calculations have been performed. This is a Relationship-level Member, so the data is for a specific Parent Entity.

---

### TRANSLATED (Execute DUCS, FinanceFunctionType = Calculate)

---

The “Translated” Member is a shortcut to the Consolidation Member that matches the Parent Entity’s default currency (e.g., USD). It stores data that was generated by multiplying local currency data by FX rates, and then executing the DUCS.

---

### Translation algorithms (Use default or implement FinanceFunctionType.Translate in a BR)

---

Determine FX Rates and multiply to create a translated Data Unit from the local currency’s Data Unit.

---

### LOCAL (Execute DUCS, FinanceFunctionType = Calculate)

---

The “Local” Member is a shortcut to the Consolidation Member for the Entity’s default currency (e.g., EUR). It stores input data in base Entities, or data that was consolidated by combining data from the TOP consolidation Member from lower-level child Entities.

---

## Example of a Consolidation

Entity	Currency
GolfStream	USD
Clubs	USD
Frankfurt	Euro
Houston	USD

## About the Financial Model

---

Clubs + GolfStream Relationship	Clubs	Interco AR	<b>Top</b> OwnerPostAdj Elimination <b>Share</b> OwnerPreAdj <b>Translated</b>	70	
	Clubs	Interco AR		0	
	Clubs	Interco AR		0	
	Clubs	Interco AR		70	
	Clubs	Interco AR		0	
	Clubs	Interco AR		70	
	Clubs	Interco AR		Pulled from Frankfurt Child Top (& other child entities)	
Frankfurt + Clubs Relationship	Frankfurt	Interco AR	<b>Top</b> OwnerPostAdj Elimination <b>Share</b> OwnerPreAdj <b>Translated (USD)</b>	70	Sum of Share + Elim + OwnerPostAdj
	Frankfurt	Interco AR		0	Calculation Sequence
	Frankfurt	Interco AR		-50	Elimination: Consolidation Logic
	Frankfurt	Interco AR		120	Share: Consolidation Logic (at 100%)
	Frankfurt	Interco AR		0	Calculation Sequence
	Frankfurt	Interco AR		120	Translation Logic then Calculation Sequence
	Frankfurt	Interco AR	Local (EUR)	100	Calculation Sequence

## Launching a Consolidation

In Workflow Profiles, set up Calculation Definitions to process the appropriate calculation, translation, or consolidation type. Right-click on the appropriate cell in a Cube View or Form to view Process options:



### Calculate

Runs calculations at the Entity level within the Local Member of the Consolidation Dimension without translating or consolidating.

Calculate

Force Calculate

Calculate with Logging

Force Calculate with Logging



### Translate

Runs the Calculate step above at the Entity and then translates data within the Translated Member of the Consolidation Dimension for each applicable Relationship.

Translate

Force Translate

Translate with Logging

Force Translate with Logging

## About the Financial Model

---



### Consolidate

Runs the Calculate and Translate steps and then completes the calculations required all the way up the Consolidation Dimension.

Consolidate

Force Consolidate

Consolidate with Logging

Force Consolidate with Logging

Force menu items such as Force Consolidate will run as if every cell included is marked as requiring calculation, translation or consolidation.

Consolidate and Force Consolidate check and determine a Parent Member's calculation status and all children of the Parent before consolidating any data. The difference is Consolidate checks Calculation Status, and if the status is OK, it accepts it and continues the consolidation process. Force Consolidate runs as if every Member needs to be consolidated regardless of its actual Calculation Status and does not bother querying Calculation Status. While they both perform optimally, there are some cases where one performs better than the other. See the examples below to learn more about when to use Force Consolidate vs. Consolidate.

## Calculation Status State 1: Actual Scenario, Loading for December



In this case Consolidate will perform better than Force Consolidate because every month prior to December has an OK Calculation Status. This means the data for that month has not changed since the last consolidation and only the month of December needs to be consolidated. If a Force Consolidate was used, every calculation would be performed again whether it is necessary or not, therefore taking longer in the consolidation process.

# Calculation Status State 2: Plan Scenario, Loading for 12 Months

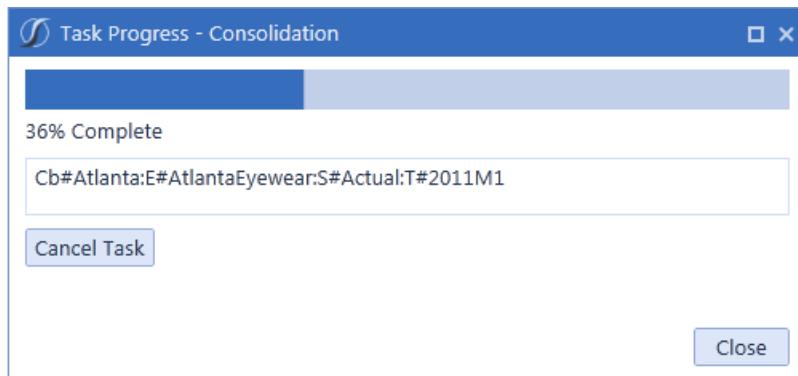


In this case, Force Consolidate will perform better than Consolidate because every month needs to be consolidated. If a Consolidate was used, the system would needlessly check each Calculation Status before calculating each period. A Force Consolidate will calculate all periods regardless. See **Calculation Status** for details on the status codes.

**NOTE:** Forced calculations that run on base entities from open periods will not impact the status of Parent Entities for closed periods if the data is in an OK state. If a metadata change occurred and entities are in an OK, MC state, all periods and entities will recalculate, regardless of Workflow Open/Close state.

The Logging items (such as Force Translation with Logging) trigger additional detailed logging

which can be viewed in the Task Activity  area. Drill into a log to see the length of time and details about every calculation. A progress window displays Consolidations.



## Calculation Status

If the existing data set for a POV changes, the calculation status is updated accordingly. A timestamp table is used, and servers are synchronized to ensure the calculation status is always accurate.

## About the Financial Model

---

To explore calculation status, build a Cube View where the View Member in the Point of View is set to CS, the columns are the time periods, and the rows are set to the desired Entity structure. For example:

The screenshot shows a Cube View interface with a tree view on the left and a grid on the right. The grid has three columns representing time periods: Jan 2011, Feb 2011, and Mar 2011. The rows represent different entities, each with a small icon and a label. The status code is displayed in each cell of the grid.

	Jan 2011	Feb 2011	Mar 2011
All Orgs	CN	CN	CN
Total GolfStream	CN	CN	CN
Clubs	CN	CN	CN
NA Clubs	CN	CN	CN
Canada Clubs	OK,MC	OK,MC	OK,MC
Montreal	OK,MC	OK,MC	OK,MC
Quebec City	OK,MC	OK,MC	OK,MC
US Clubs	CN	CN	CN
Augusta	OK,MC	OK,MC	OK,MC
Carlsbad	OK,MC	OK,MC	OK,MC
Houston	CN	CN	CN
Houston Heights	OK	OK	OK
South Houston	OK,MC	OK,MC	OK,MC
Europe Clubs	CN	CN	CN
Frankfurt	OK,MC	OK,MC	OK,MC
Golf Balls	OK,MC	OK,MC	OK,MC

These status codes indicate what action to perform:

- **OK:** The data for this intersection has not changed since the last calculation.
- **OK, NA:** Calculation not required. No data.
- **OK, MC:** The intersection was calculated but metadata changed due to modifications to artifacts such as Business Rules associated with this Cube, formulas and FX rates. This is a clue that if the calculation is run again, the results may not be the same.
- **CA:** Calculate data since an import was run or data was entered.
- **TR:** Translate.
- **CN:** Consolidate because data for a child Entity has changed.
- **CA, CN:** Calculate and consolidate.

- **CA, TR:** Calculate and translate.
- **TR, CN:** Translate and consolidate.
- **CA, TR, CN:** Calculate, translate and consolidate.

## Currency Translation

Currency can be converted from one currency to another. This process utilizes the defined FX Rates in the FX Rates portion of the product. See Foreign Exchange Rates in "Cubes" on page 328 for additional details.

The base Entity can be converted to the Parent Entity's currency if required. The currency translation is run as needed based on the configuration of the Parent / child currencies as stored in the Entity Dimension. This feature can be run independently of a consolidation if required. Right-click in the appropriate cell and the translation option will display.



The Consolidation Dimension is shown above. For example, if an Entity's currency is Euro, and data is written to the EUR Member, that value is also displayed in the Local Member. The Local Member is a pointer to the appropriate local currency Member where the data is stored. If that European Entity in the screen shot above is consolidated to a company in the UK that has GBP as their Currency setting under Entities, when the European Entity is consolidated into the UK Parent, the European Entity's Translated Member will reference the translated value which is stored under the GBP Member under Currencies.

Calculation Status determines when data was entered. If the Local currency's Calculation Status is CA because the data has not been calculated yet, then the foreign currencies would also need to be translated and calculated resulting in a TR calc status. If a calculation is done on the Local Currency in order to make its calc status OK, the foreign currencies are still going to be TR because that data has not been translated or calculated yet. Also, if a foreign currency has an OK calculation status, and a foreign currency journal is entered, that currency then becomes TR.

## How the Origin Dimension Works with the Consolidation Dimension

The Origin and Consolidation Dimensions are separate but work well together. Origin primarily calls out how a number originated. This section explains how each Member of the Origin Dimension interacts with Members of the Consolidation Dimension.

The consolidation process starts at Local, which is the same data as one of the currencies under the Currencies Member (based on the Entity's default currency setting). If the Entity is base-level, the data can be loaded into the Local Consolidation Member using the Import Origin Member, and data can be typed into the Local Consolidation Member using the Forms Origin Member. If it is a Parent Entity, the Local Consolidation Member is read only (except for Journals) as it represents data that has been rolled up from child Entities (using the Import, Forms, and the AdjConsolidated Origin Members).

Regardless of whether the Entity is base-level or a Parent, Journals can be entered using the Local, Translated, OwnerPreAdj, OwnerPostAdj, and Any Currency Consolidation Member. For all the Consolidation Members, Journals are always posted to the AdjInput Origin Member. As data is rolled up from child Entities into a Parent Entity during the consolidation process, the AdjConsolidated Origin Member will contain the combined values from the child Entities' Journals.

## Imports and Forms

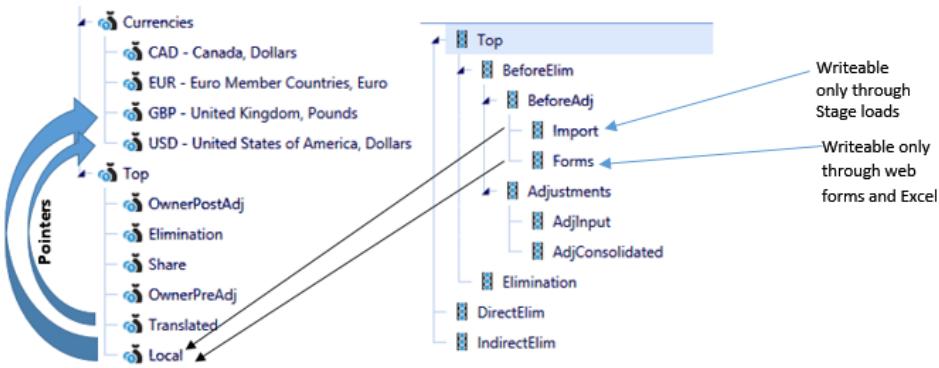
When data is imported through Workflow to the Data Stage, it is happening through the Import Member in the Origin Dimension. When data is entered in web or Excel Forms, it is happening in the Forms Member in the Origin Dimension (with one exception discussed under Adjustments). The Import and Forms Members are coupled with the Local Member of the Consolidation Dimension.

Consolidation Dimension

Origin Dimension

## About the Financial Model

---



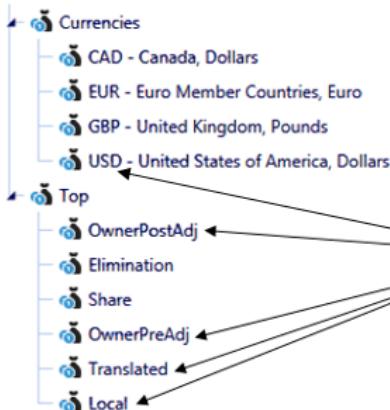
## Adjustments

Adjustments are created either through Journal entries, or in special instances, through Forms that do their input as a Journal would. Both do their input into the AdjInput Member of the Origin Dimension. Adjustments can be made to the following Members of the Consolidation Dimension: Local, Translated, OwnerPreAdj and OwnerPostAdj.

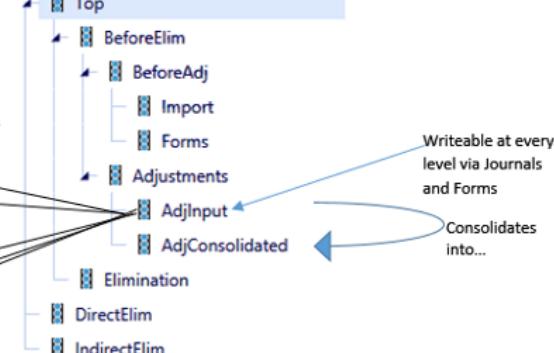
When a consolidation is run, the AdjInput entries in child Entities are consolidated into the AdjConsolidated Members in the Parent.

Users can drill down into the Adjustments Member in a Parent to show adjustments made in both the Parent and child Entities.

Consolidation Dimension



Origin Dimension

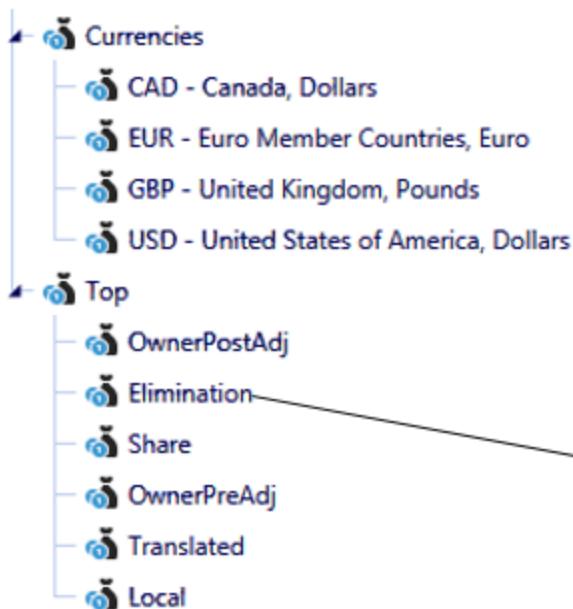


## Eliminations

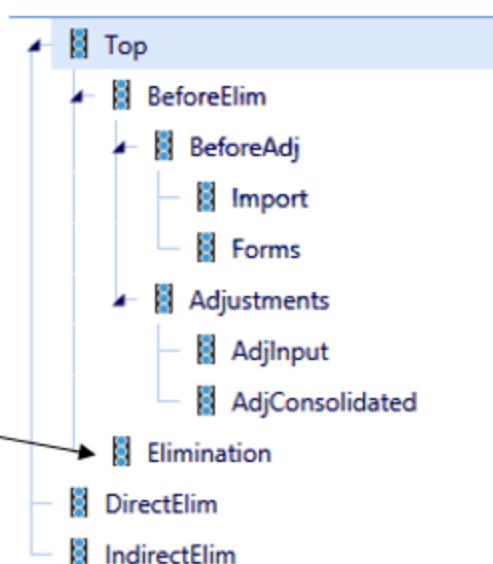
When eliminations are calculated, the Elimination Member of both the Consolidation and Origin Dimensions is updated. The primary purpose of the Elimination Member within the Origin Dimension is to allow visibility from the top down without those figures getting lost during the consolidation process.

Users can drill down into the O#Elimination Member in a Parent to show the elimination entries made in both the Parent and child Entities.

Consolidation Dimension



Origin Dimension

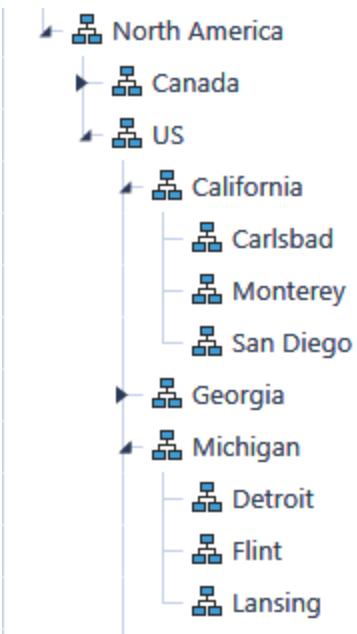


## Intercompany Elimination

Intercompany Elimination is the process of cancelling out account balances for intercompany partners for intercompany accounts with any unresolved balance being placed in a Plug Account.

## About the Financial Model

---



The Entity structure above belongs to GolfStream, a fictitious golf manufacturer. If the Detroit Entity sold golf club shafts to Monterey who assembles the final club product, this would be an intercompany transaction. The following prerequisites must exist for the transaction to eliminate.

- Monterey and Detroit must have their Is IC Entity property set to True.
- The Accounts Intercompany Receivables and Intercompany Payables must be set with the Is IC Account property set to True and the Plug Account pointed to a third account.
- The intercompany entries must properly note the intercompany partner in the IC Dimension Member. For example, Detroit would book an entry to Intercompany Receivables and the IC Member for that entry would be Monterey.

Intercompany Eliminations occur once the values roll up to a common Parent. As the consolidation begins, Detroit consolidates its values to Michigan and Monterey consolidates its values to California. An elimination does not occur because they have not yet consolidated their values to a common Parent. The elimination occurs when Michigan and California are consolidated into the US common Parent. The two intercompany values will be eliminated at this level with any discrepancies being posted to the Plug Account.

In another example in Workflow, Houston Heights and South Houston trade with Carlsbad, Dallas, and Montreal. In the screen shot below, South Houston is shown in green because it is balanced within an acceptable tolerance of \$1. The \$0.59 discrepancy is booked to the related Plug Account.

## About the Financial Model

---

IC Matching Summary [20300]						
Status	Entity Name	Unbalanced (USD)	Partners	Pairings	Balanced	Unbalanced
Red	Houston Height	-229.15	3	3	1	2
Green	South Houston	0.00	2	2	2	0

( Houston Heights ) IC Matching Partner Detail						( Montreal - Houston Heights ) Status
Pairing Group	Account	Report Curr.	Entity Curr.	Partner Curr.		
▼ Carlsbad						✓ 20200 - IC Payables
▲ Dallas	-500.00 USD					● Disputed (Partner Status)
Houston Heights - Dallas	20200 - IC Payables	40,000.00 USD	40,000.00 USD	40,000.00 USD		Administrator
Dallas - Houston Height	11200 - IC Receivables	39,500.00 USD	39,500.00 USD	39,500.00 USD		10/2/2011 6:18 PM
	Difference	-500.00 USD	-500.00 USD	-500.00 USD		Some items being returned due to damage. Please issue credit memo for \$275.60 CAD.
▲ Montreal	270.85 USD					
Houston Heights - Mont	11200 - IC Receivables	4,000.00 USD	4,000.00 USD	4,075.60 CAD		
Montreal - Houston Heig	20200 - IC Payables	3,729.15 USD	3,729.15 USD	3,800.00 CAD		
	Difference	270.85 USD	270.85 USD	275.60 CAD		

Houston Heights is shown in red because there is a \$229.51 discrepancy. Details on the discrepancy are shown in the lower part of the screen. Houston Heights' discrepancy with Montreal is shown at the bottom in the application reporting currency (USD), Houston Heights' currency (USD), and the partner currency for Montreal (CAD). Through the right click menu, leave a status and description that each partner can see. As these values roll up the Entity structure, the Parent Entities can also see this detail.

See Workflow Profiles in "Workflow" on page 444 for more details on how to set up Intercompany Matching via Workflow.

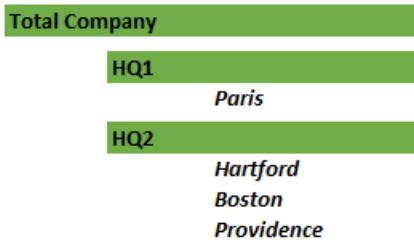
**NOTE:** Using Extensible Dimensionality to extend Accounts used within the Intercompany Matching process is not a recommended practice.

## Direct and Indirect Eliminations

The Origin Dimension contains the Direct and Indirect members to simplify reporting on Eliminations. The Direct member returns the results of Eliminations that occur from transactions, removing the direct children of a parent Entity. Indirect returns the total Eliminations that occurred outside the direct children of a parent Entity.

## About the Financial Model

---



In the example, transactions that occurred between the HQ2 Entities eliminate at HQ2. Transactions between members of HQ1 and HQ2, such as a transaction between Paris and Hartford, eliminate at the first common parent, Total Company.

Reporting on the results at the Total Company level, Direct returns results that occurred between the HQ1 and HQ2 groups. Indirect at Total Company would allow reporting on eliminations outside its direct children, HQ2 eliminations.

Direct and Indirect									
	Paris USD BeforeElim	Hartford USD BeforeElim	Boston USD BeforeElim	Providence USD BeforeElim	Total Company USD BeforeElim	Total Company USD DirectElim	Total Company USD IndirectElim	Total Company USD Top	
ICRec	0.00	5,400.00	2,000.00	1,200.00	8,600.00	-4,600.00	-4,000.00	0.00	
ICPay	5,040.00	0.00	0.00	4,000.00	9,040.00	-5,040.00	-4,000.00	0.00	
ICDiff	0.00	0.00	0.00	0.00	0.00	440.00	0.00	440.00	

## Custom Consolidation

If the standard consolidation or translation logic does not meet your project requirements, deploy custom Business Rules. The first step in this process is defining the Cube's properties settings. See "Cubes" on page 328 in Calculation section for Consolidation Algorithm Type and Translation Algorithm Type settings. The default setting is Standard but you can change this to Custom for additional flexibility.

Next, write a Finance Business Rule. See:

- "Application Tools" on page 727 in the Business Rule section on applying Business Rule logic by Finance Function Type to discover what you can customize.

- "Formulas" on page 25 for common customizations.
- The API Overview Guide.

Reference the Finance Business Rule under any of the Business Rule 1-8 properties for each Cube on which to use the logic. Also see Data Unit Calculation Sequences (DUCS).

## Equity Pickup

OneStream supports Equity Pickup calculations using three different properties located under the Entity's settings all of which can vary by Scenario Type. These settings were designed to implement Equity Pickup using normal Business Rules and formulas.

See Entity Dimension in "Cubes" on page 328 for definitions of each setting.

## Equity Pickup Example

The Entity hierarchy example below illustrates how Equity Pickup works in OneStream and discusses how each Entity's currency and Equity Pickup settings affect calculation and consolidation.

- Clubs (USD)
- Holding (EUR)
- Houston (USD)
- Carlsbad (USD)
- Frankfurt (EUR)

In the example above, the formulas for Holding need to read calculated data from the other sibling Entities. Therefore, the Sibling Consolidation Pass property for Holding would be set to Pass 2 causing calculation to occur on all the other sibling Entities before the Holding Entity is calculated. This allows the formulas for Holding to correctly read calculated data from Houston, Carlsbad, and Frankfurt. For Entities not involved in Equity Pickup, the (Use Default), or Pass 1 settings for Sibling Consolidation Pass causes all sibling Entities to be calculated at the same time.

Holding is using a different local currency than its Parent Clubs, but only wants to read data using the EUR currency. In this situation, the Auto Translation Currencies setting for Houston and Carlsbad needs to be set to EUR in order to have them automatically translate to EUR when Clubs is consolidated. Normally, all the sibling Entities translate to the Parent Entity's local currency, which in this case is USD, however this setting tells the engine to translate Houston and Carlsbad to EUR as well during the consolidation. Once the consolidation is complete, Holding's formulas, which are calculated in Pass 2, can read data from E#Houston:C#EUR, E#Carlsbad:C#EUR, and E#Frankfurt:C#EUR.

Sibling Repeat Calculation Pass is designed for circular ownership and may not be used as often as the other Equity Pickup settings. If this is used, it causes the Entity's calculation to be repeated after all the Sibling Calculation Passes have been completed. For example, if there was another Entity in the structure above named Holding2, it would be set to use a Sibling Calculation Pass of Pass 3. This would cause its normal calculation to occur after Holding and allow Holding2 to read calculated data from Holding. Holding could also use a Sibling Repeat Calculation Pass causing it to be recalculated. In that repeat calculation, Holding could then read calculated data from Holding2 resulting in circular ownership. When writing formulas, use api.Args.CalculateArgs.HasRepeatCalc and api.Args.CalculateArgs.IsRepeatCalc to determine if the engine is currently running the repeat calculation.

## Entity Aggregation

Entity Aggregation provides the speed and flexibility required for Budgeting, Planning and Forecasting. Unlike consolidation processes, entity aggregation is simpler and faster because it does not roll up financial data - heavily driven by financial and accounting rules - to a parent level for reporting.

Consider this: "Consolidations are usually crafted to satisfy internal management and external regulatory agency reporting requirements. The most common, effective way to understand the core requirements of a consolidation system is to begin with the end in mind and look at the reports produced by the legacy (or current) system. These usually involve an Income Statement (Profit & Loss), a Balance Sheet, and a Cash Flow Statement" – OneStream Architect Factory

## How Entity Aggregation Works

Entity Aggregation bypasses most statutory financial and accounting rules to quickly consolidate / aggregate data the entity dimension for fast, what if scenario modeling. Entity Aggregation uses the Entity and the Consolidation dimension (Aggregated member) to aggregate data with minimal rules.

## About the Financial Model

---

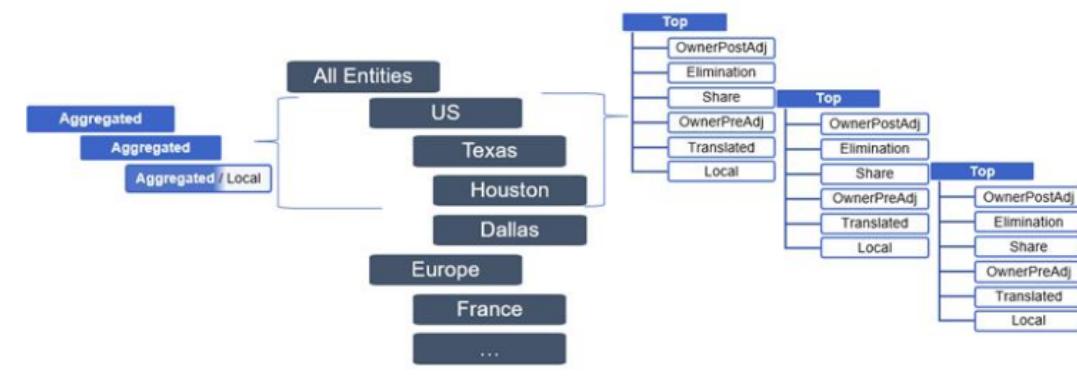
For base level entities, the Aggregated member displays the data that is stored in the “Local” member.

For parent entities, the Aggregated member stores the results of the Entity Aggregation process that occurs on its children.

Aggregation Algorithm:

- Execute chart logic (business rules and member formulas) on the Local Consolidation member for all base entities.
- Execute these steps recursively for each Parent and its direct children, from lower-level entities to the parent entities:
  - For each child:
    - Translate stored data in memory.
    - Calculate the share amount in memory
  - Add the data cells from each child in memory.
  - Store the results in the Aggregated member for the parent entity.

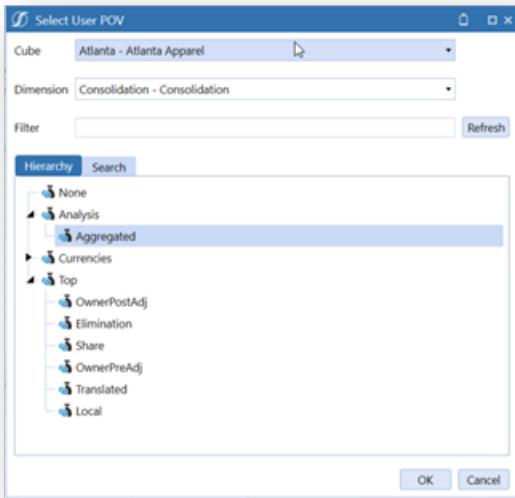
## Example of an Entity Aggregation



## About the Financial Model

---

### Aggregated Dimension



## About the Financial Model

---

### Example of Cube View Using C#Aggregated Dimension

The screenshot shows the Microsoft Analysis Services Designer interface. At the top, there are tabs for 'Designer' (selected) and 'Advanced'. Below this is a navigation pane with icons for 'POV', 'General Settings', 'Report Header', and 'Rows and Columns'. The main workspace displays a table structure with one row ('Row1') and one column ('Col1'). The value in the cell is '9,999.99'. Below the table, there are three tabs: 'Member Filters', 'Formatting', 'Data', 'Sharing', and 'Row Overrides'. Under 'Row Overrides', the 'Row' is set to 'Row1'. The 'Entity' filter is set to 'E#[North America].TreeDescendantsInclusive'. There are four levels of member filters: Level 2, Level 3, and Level 4, all labeled '(Not Used)'. At the bottom of the workspace is a 'Report Footer' section.

- Rows
  - Entity: E#[North America].TreeDescendantsInclusive
- Columns
  - Consolidation: C#Aggregated, C#Aggregated:V#CalcStatus, C#Local, C#Local:V#CalcStatus
  - Time: T#2018M1, T#2018M12

## About the Financial Model

---

	Aggregated Jan 2018	Aggregated Jan 2018	Local Jan 2018	Local Jan 2018
North America	888,005.77 USD	OK	888,005.77 USD	OK
Canada	340,000.00 CAD	OK	340,000.00 CAD	OK
Ontario	5,000.00 CAD	OK	5,000.00 CAD	OK
Ottawa	5,000.00 CAD	OK	5,000.00 CAD	OK
Toronto		OK		OK
Quebec	335,000.00 CAD	OK	335,000.00 CAD	OK
Montreal	85,000.00 CAD	OK	85,000.00 CAD	OK
Quebec City	250,000.00 CAD	OK	250,000.00 CAD	OK
US	550,000.00 USD	OK	550,000.00 USD	OK
California		OK		OK
Carlsbad		OK		OK
Monterey		OK		OK
San Diego		OK		OK
Georgia		OK		OK
Atlanta		OK		OK
AtlantaClothing		OK		OK
AtlantaFootwear		OK		OK
AtlantaEyewear		OK		OK
Augusta		OK		OK
Roswell		OK		OK
Michigan	550,000.00 USD	OK	550,000.00 USD	OK
Detroit	50,000.00 USD	OK	50,000.00 USD	OK
Flint	500,000.00 USD	OK	500,000.00 USD	OK
Lansing		OK		OK
New York		OK		OK
Texas	0.00 USD	OK	0.00 USD	OK
Austin	0.00 USD	OK	0.00 USD	OK
Dallas		OK		OK
Houston	0.00 USD	OK	0.00 USD	OK
Houston Heights	0.00 USD	OK	0.00 USD	OK
South Houston		OK		OK

Aggregate Jan 2018, by right-clicking on the number and pick Consolidate to Aggregate numbers. Context aware algorithms are used recognize the Aggregated member.

Notes:

- No Parent Journal Adjustments
- No Eliminations
- Business Rules (execute at the base level only)
- Select different Consolidation member (C#Aggregated) to run Consolidation on and pick the numbers that were loaded.

## Launching an Aggregation

This is the same process as launching a Consolidation using context aware algorithms once the Aggregated member of the Consolidation (C#Aggregated) is selected. For example, right-click on the appropriate cell in a Cube View or Form to view the Process by clicking Consolidate.

## About the Financial Model

---



### Consolidate

- Consolidate
- Force Consolidate
- Consolidate with Logging
- Force Consolidate with Logging

Aggregate Jan 2018, by right-clicking on the number and picking Consolidate to Aggregate numbers. Context aware algorithms are used recognize the Aggregated member.

The screenshot shows the Data Explorer interface with a tree view of geographical entities on the left and a grid of financial data on the right. A context menu is open over a cell in the grid, with 'Consolidate' highlighted. A secondary dialog, 'Edit PCV Information', is open over the same cell, showing the 'CMaggregated' tab selected. The dialog displays properties like 'Name' (CMaggregated), 'Description' (C#Aggregated), and 'Properties' (Member Dimension).

## Entity Aggregation Input

Entity aggregation input lets you submit data for base member values through the aggregated member. Use the “Allow Input Into the Aggregated Consolidation Member” property to enable input. Values entered into an aggregated cell are saved to the respective local currency member (C#USD, C#CAD, and so forth).

**IMPORTANT:** Data entered through journals cannot be entered into the aggregated member.

This feature is available in versions 7.4 and higher.

## Enable Entity Aggregation Input

You must first enable aggregation input on the scenario you are using.

1. On the Application tab, click **Dimensions**.
2. Expand **Scenario Dimensions**.
3. Expand the appropriate dimension, then select **Scenarios**.
4. On the Members tab, expand **Hierarchy** and click on the aggregated member to select it.
5. Click the **Member Properties** tab.
6. Under Settings, set **Allow Input into the Aggregated Consolidation Member** to TRUE.
7. Click **Save**.

## Working With Hybrid Scenarios

Hybrid scenario data improves query performance if you analyze smaller data sets from large data units that contain high volumes of account-level detail for a cube such as scenario and entity. This enhances query performance when:

- Different business units require smaller subsets of the data unit because they only need to report on specific accounts, cost centers, products and so on.
- Business processes require a higher level view of data for budget or planning.

In these cases, data is filtered to focus on business and consumption needs for more efficient data analysis.

Hybrid data uses data from a source scenario member and displays the results in a target scenario member. You may need multiple target scenarios based on how you use hybrid source data in your application.

For information about associated properties, see "Hybrid Scenarios" on page 351. Also see:

- "Data Bindings" on the next page
- "Data Filters" on page 111

## About the Financial Model

---

- "Share Data From the Source Scenario" on the next page
- "Copy Data From the Source Scenario" on page 112

## Data Bindings

Data bindings determine if data is shared or copied from the source scenario member to the target scenario member. Sharing data is best if you need to analyze smaller data sets from a large data unit.

The data results are dynamic and reference source data. This data is not stored to the target scenario and is read-only. Shared scenarios also share the source scenario's calculation status and indicate source data changes. Standard calculations run from a shared scenario also run on the source scenario member.

Use Copy Data to compare 'What If' Scenarios, for budget versioning or forecasting. By default a data copy occurs if a standard calculation - associated with the following - is run on the target scenario:

- Cube view or Quick View grid
- Calculation definitions
- Calculate data management step

The **Calculate Data Management Step** offers more control over the hybrid source data process. Set **ExecuteCopyAfterCalc Scenario** to **False** and enable the copy execution on the Data Management Step. If enabled, calculations run on the scenario, but the data copy only occurs if the calculation runs from the Data Management Step.

General (Step)	
Name	HybridSourceCopy
Description	
Data Management Group	HybridSourceCopy
Step Type	
Calculation	
Calculation Type	Calculate
Execute Scenario Hybrid Source Data Copy	True

Calculations follow the standard sequence and store the data as "calculated". If multiple calculations run, the previously calculated or copied data is cleared. To preserve existing data, set the data as "durable" on the target scenario.

## About the Financial Model

---

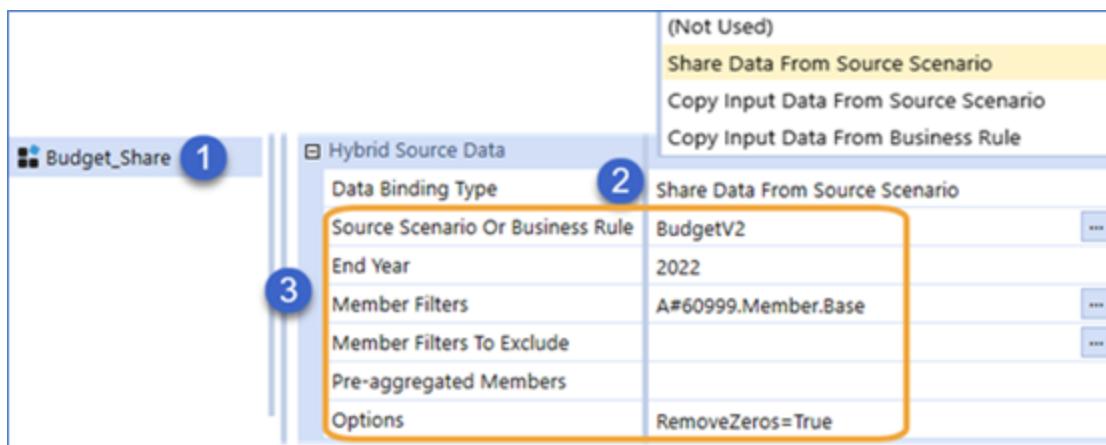
Once the target scenario copy finishes, you can modify and adjust data, but the target scenario is still bound to the source. The data copy occurs whenever the scenario is calculated based on the scenario setup. Hybrid copies can be chained, so results in a target scenario can be used as the source in another.

## Data Filters

Hybrid source data generates unique data views using different filters, so you can best focus your data modeling and analysis.

Pre-aggregated members provide a summarized source data view by defining the source parent members and target base members. Members are filtered before aggregation and cached in RAM. Then the smaller data set is brought in and Share or Copy the information over to the target accounts.

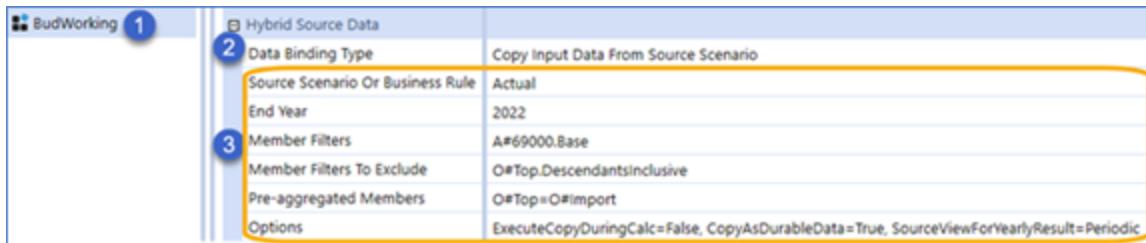
## Share Data From the Source Scenario



1. Create a target scenario called Budget\_Share.
2. Specify a Data Binding Type: Share Data from Source.
3. Select source scenario and data settings:
  - Source scenario: BudgetV2
  - Data binding ends in 2022

This shares the base accounts for Account 60999. Any zeros in the source data are removed.

# Copy Data From the Source Scenario



1. Create a target scenario called BudWorking.
2. Specify a Data Binding Type: Copy Input Data from Source Scenario.
3. Select the source scenario and data settings:
  - Source scenario: Actual
  - Data binding ends in 2022

This copies the base accounts of Account 69000 and excludes Origin dimension members. Data is copied from the top of Origin in the source scenario to the Import member of Origin in the target scenario. The calculation occurs only if run from the Data Management Step, and the copied data is stored as durable.

# Navigation

You can access OneStream through a web browser or via OneStream Windows App through an application shortcut deployed via a OneStream website, or from a version installed directly on a computer. These options are available for both administrators and end users.

The layout is intuitive and easy to use. OnePlace will highlight all the major touch points as well as provide information on the navigation tips.

**TIP:** Hover over any of the icons and a tool tip will temporarily display.

## About the OneStream Windows Application

Using the Windows App is helpful because:

- The Windows app automatically updates whenever the application server version is updated.
- You do not need admin rights to download or use the application.
- It offers robust spreadsheet functionality, so you may not need to install the Excel Add-in.

To launch the Windows App:

1. Click the launch icon in the upper right corner of the window from any 4.2 or greater web-based instance.
2. If prompted, click **Run**.

You can also launch the Windows App via Microsoft Edge with the URL associated with the latest version of the server. First, enable Edge Chromium support for Click Once.

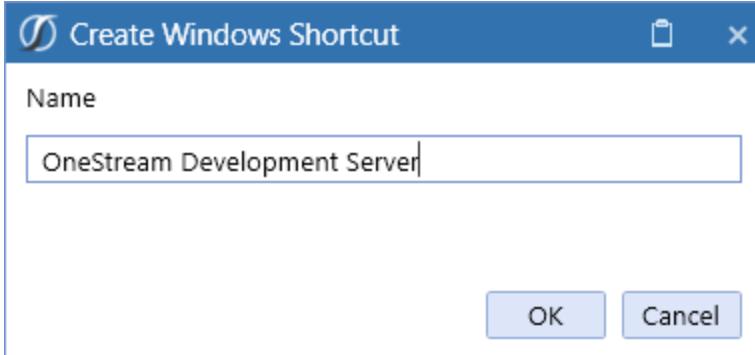
See <https://docs.microsoft.com/en-us/deployedge/microsoft-edge-policies#clickonceenabled>.

To create a desktop shortcut:

## Navigation

---

1. Click . The Create Windows Shortcut window displays.



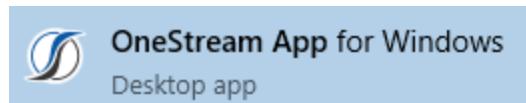
2. Modify the shortcut name as needed and click **OK**.



This icon can now be used to launch the Windows App from your desktop.

## Launching the OneStream Windows App

The Windows App is installed locally and can access different servers. To launch the app, click, **Start** and locate the OneStream Desktop app.



## Defining Server Connections

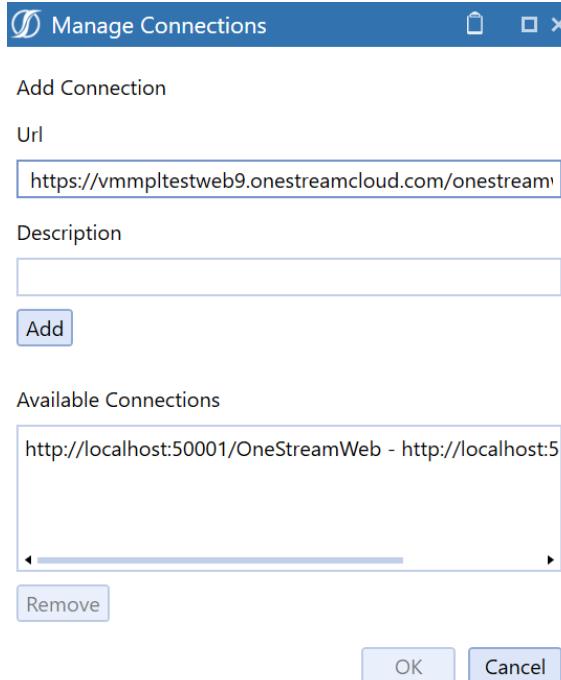
Before you can log in and open an application, you must define server connections:

## Navigation

---

1. On the **Logon** page, click  in **Server Address** to add the URL of the server to connect to.

The Manage Connections window displays.



2. Enter the server URL and a description.
3. Click **Add** to include the server in the list of **Available Connections**.
4. In **Available Connections**, select the server and click **OK**.

## Logging In

If you are in a OneStream-hosted environment, see the *Identity and Access Management Guide* for information about authentication with OneStream IdentityServer.

If you are in a self-hosted environment, follow these steps to log into OneStream.

1. On the OneStream **Logon** screen, for **Server Address**, specify the URL or client connection. See [Defining Server Connections](#).

## Navigation

---

2. Click the **Connect** button.

**NOTE:** The Logon screen will look different depending on how your environment is configured (native authentication, external identity provider, or both native authentication and an external identity provider).

### Native Authentication Only

The screenshot shows the ONESTREAM Logon window version 8.0.0.15015. The window title is "Logon (8.0.0.15015)". The main area is titled "Authentication". It includes fields for "Server Address" (http://localhost:50001/onestreamweb), "User Name", "Password", and a "Remember User" checkbox. There are "Logon" and "Logoff" buttons. Below these are dropdown menus for "Application" and an "Open Application" button.

## Navigation

---

### External Identity Provider Only



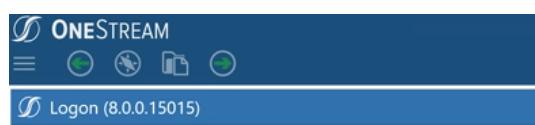
### Authentication

Server Address

...

Application

### Native Authentication and an External Identity Provider



### Authentication

Server Address

...

Or

User Name

Password

Remember User

Application

3. Follow the steps for your authentication configuration:

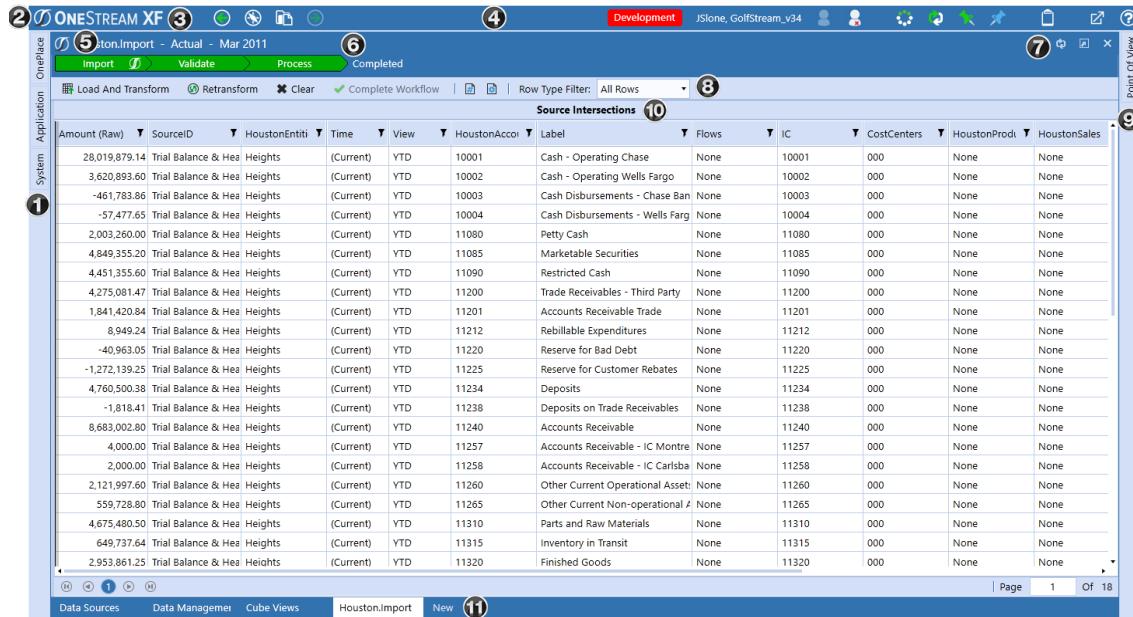
- For native authentication:
  - a. Enter your user name and password.
  - b. Click the **Logon** button.

**NOTE:** This process for native authentication is also used for authentication with MSAD and LDAP identity providers. Enter your user name and password set up in MSAD or LDAP and click the **Logon** button.

- For an external identity provider:
    - a. Click the **External Provider Sign In** button.
    - b. Enter your external identity provider login credentials.
4. On the OneStream **Logon** screen, select an application from the drop-down menu.
5. Click the **Open Application** button.

For instructions on configuring authentication in a self-hosted environment, see the *Installation and Configuration Guide*.

# OnePlace Layout

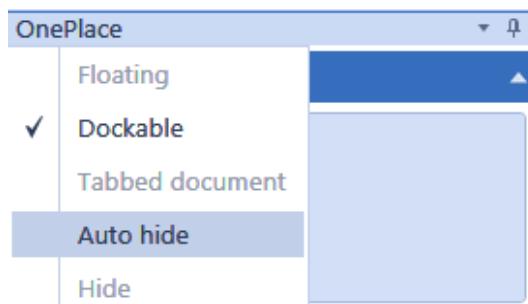


The screenshot shows the OnePlace Layout interface. At the top, there's a toolbar with icons for Import, Validate, Process, and Completed. Below the toolbar is a navigation pane with tabs for System, Application, and OnePlace. The OnePlace tab is selected. The main area contains a data grid titled "Source Intersections". The grid has columns for Amount (Raw), SourceID, HoustonEntity, Time, View, HoustonAcco, Label, Flows, IC, CostCenters, HoustonProd, and HoustonSales. The data grid displays numerous rows of financial data. At the bottom of the interface, there are buttons for Data Sources, Data Management, Cube Views, and a New button.

1. Navigation Pane - This section covers the three tabs that are available: System, Application and OnePlace.

**NOTE:** Tab visibility is determined by users' security settings.

Each bar can be displayed by pinning it to the screen, or by Auto Hide. Additional details can be found further down in this section.



## Navigation

---

2. Home - Click the large OneStream icon to navigate to the user's set home screen. See Page Setting Options below for more information on setting home screens.
3. Title Bar - This displays the OneStream Logo.
4. Application Tray



### Hamburger Menu (Navigation Pane)

Clicking (not hovering) this icon will hide or unhide the Application, System, OnePlace Tabs on the left-hand side of the OnePlace screen. If the Navigate Pane is unpinned, clicking (not hovering) this icon will hide the Application, System, OnePlace Tabs.



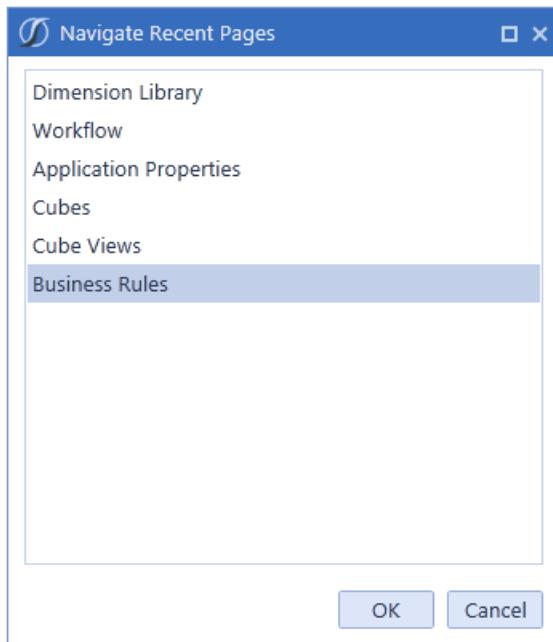
### Back

This will navigate back to the last open screen and continue to navigate back to each previously opened screen.



### Navigate Recent Pages

This opens a navigation dialog allowing the user to select and go back to a recent page.



### File Explorer

## Navigation

---

This opens the File Explorer dialog allowing users to access public folders, documents and the File Share.



### Forward

This works with the Back icon and navigates forward to the screen a user was previously using.

Development

### Environment Name

This is a customized environment name which can be made different across environments (e.g. Development, Test and Production). Specify an environment name and color in the application's application server configuration file. See Installation and Configuration.

JStone, GolfStream\_v34

User ID and Application



### Logon/Logoff Icon

Upon selecting the Logoff icon, the user will be prompted to End Session or Change Application. End Session logs the user off and removes the saved password from the logon screen. Change Application keeps the user logged in and allows him/her to select a new application from the drop-down screen.



### Task Activity

This displays all tasks performed within the application. See Task Activity in "Logging" on page 890 for more details on this feature.



### Refresh Application

Refreshes the Application and checks the first open tab. If it is an Application tab, the view will change to that tab. If it is not an Application tab, the view will stay on the selected tab but will change the main active tab to Application.



### Pin or Unpin Navigation Pane

This will hide or unhide the Application, System, OnePlace Tabs on the left-hand side of the OnePlace screen.



### Pin or Unpin POV Pane

This will hide or unhide the POV on the right side of the OnePlace screen.

## Navigation

---



### Clipboard

Drag and drop items such as data cells, text, rule scripts to the clipboard in order to reuse them in other areas. Users can store up to ten items on a clipboard.



### Launch OneStream OneStream Windows App

Click this to launch the application as a OneStream Windows App. See [Launching OneStream Windows App](#) for more details on this feature.



This opens OneStream documentation for Platform and MarketPlace.



### Hamburger Menu (POV Pane)

Clicking (not hovering) this icon will hide or unhide the POV on the right side of the OnePlace screen. If the Navigate Pane is unpinned, clicking (not hovering) this icon will hide the POV on the right-hand side of the OnePlace screen.

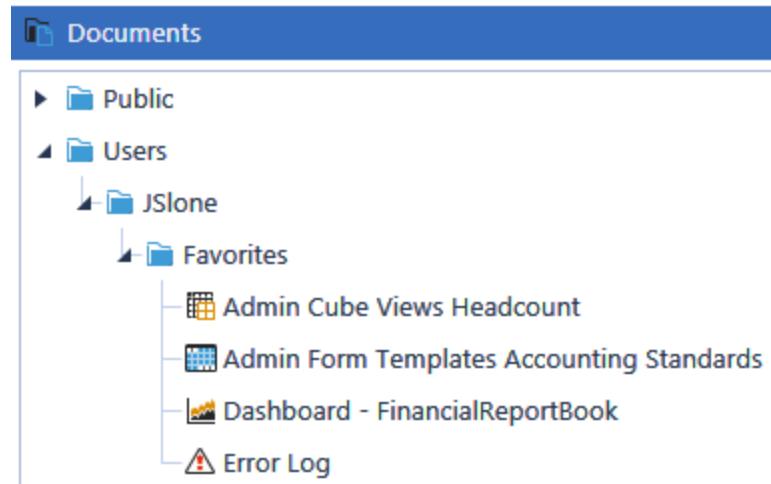
#### 5. Page Setting Options

Click the small OneStream icon to use the following page settings:

- a. Refresh Page: This will refresh the current page.
- b. Close Page: This will close the current page.
- c. Create Shortcut: This will create a shortcut for the current page and store it in the user's Favorites Folder. When this shortcut is selected from the user's folder, it will navigate to that page. This can be used for specific Cube Views, Dashboards, or Application/System pages.

## Navigation

---



- d. Set Current Page as Home Page: This setting controls the default settings for both the page display as well as the pinning of the Navigation and Point-of-View panels when you sign on. Click the OneStream icon in the application tray in order to navigate to the home page from any other screen. This functionality works in both the browser and the OneStream Windows App versions. Changes made here within one environment carry over to the other.

This also controls the Pinning of the Navigation Bar and the POV Bar as well.

This generates a UserAppSettings.XML file that has the following pin options:

```
<SLHomePagePinNavPane>TrueValue</SLHomePagePinNavPane>
<SLHomePagePinPovPane>FalseValue</SLHomePagePinPovPane>
```

These control if the user's navigation bar is pinned by default when logging into the application.

1. Clear Home Page Setting

This will remove the current home page setting. This functionality works in both the browser and the OneStream Windows App versions. Changes made here within one environment carry over to the other.

2. Save Home Page Setting As Default For New Users

This will save a home page as the default home page for any new user logging in for the first time. This functionality works in both the browser and the OneStream Windows App versions. Changes made here within one environment carry over to the other.

3. Close All Pages

This closes all open pages.

## Navigation

---

### 4. Close All Pages Except Current Page

This will close all open pages except the one currently displayed.

### 5. Workflow Bar

This section displays exactly where the user is in the Workflow process. Based on the Workflow Profile, this can be configured as a Certifier or as a Data Loader. The example above is configured as a Data Loader on the Validate task of the Workflow. The color green indicates a completed task, blue indicates incomplete tasks. The white OneStream icon indicates what task is currently in view.

### 6. Page Refresh

This section covers the local refresh and the ability to close a page.



#### Refresh Page

This refreshes the active page.



#### Toggle Page Size

This changes the page size in order to fit the screen.



#### Close Page

This closes the active page.

### 7. Toolbar

Similar to the Workflow bar, this displays the items for a Data Loader or for a Reviewer / Certifier. The example below is configured as a Data Loader during the Import task.



### 8. Context Pane

This bar is where the Point of View is set. This is an important concept because it determines to which Dimensions the users will have access for this application view. Additional details can be found further down in this section.

### 9. Grid

This displays the active window's contents for the functionality being executed.

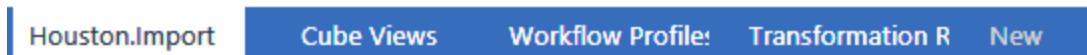
## Navigation

---

- Pages: This allows the ability to navigate through the pages either by choosing the page, by clicking on the next page, or the first / last page.



- Tabs: Now that the tab is open, simply click on one of them to save time in navigating. Clicking the New tab allows two or more like tabs to be opened. For example, two Cube Views tabs can be opened at once.



### 10. Pages

**NOTE:** Right-click on any of these opened tabs for more page setting options.

The following icons are only located in the OneStream Windows App.



#### Create Windows Shortcut

This creates a desktop shortcut for the application. See "About the OneStream Windows Application" on page 113.



#### Zoom Options

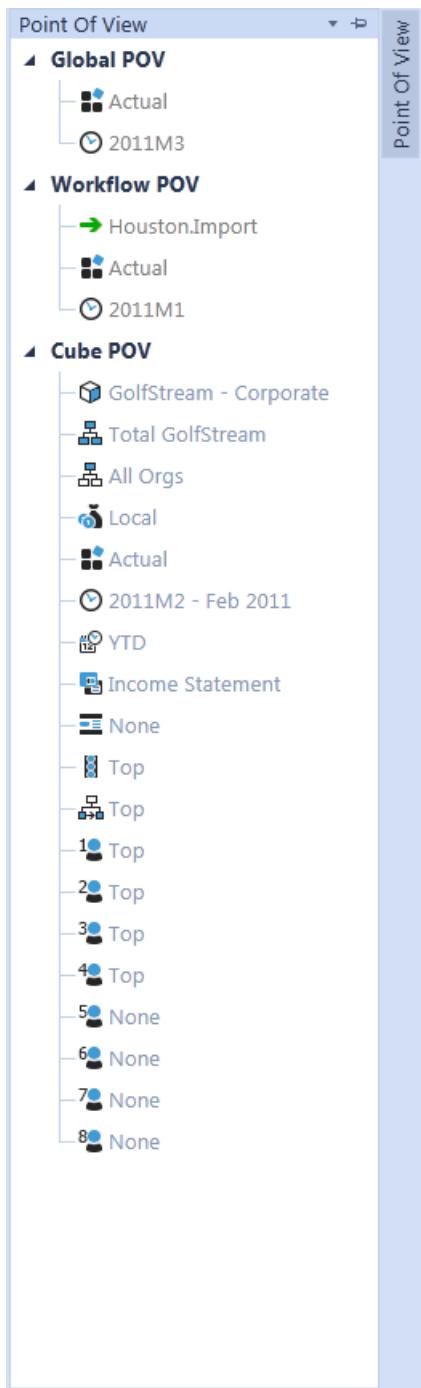
This controls the zoom settings when working in the OneStream Windows App.

## Point of View

The Point of View is located on the right side of the application. This tab can be docked by clicking the pin button, otherwise, it will disappear when clicking anywhere in the main page.

## Navigation

---



There are three primary sections defined under the Point of View.

## Navigation

---

### Global POV

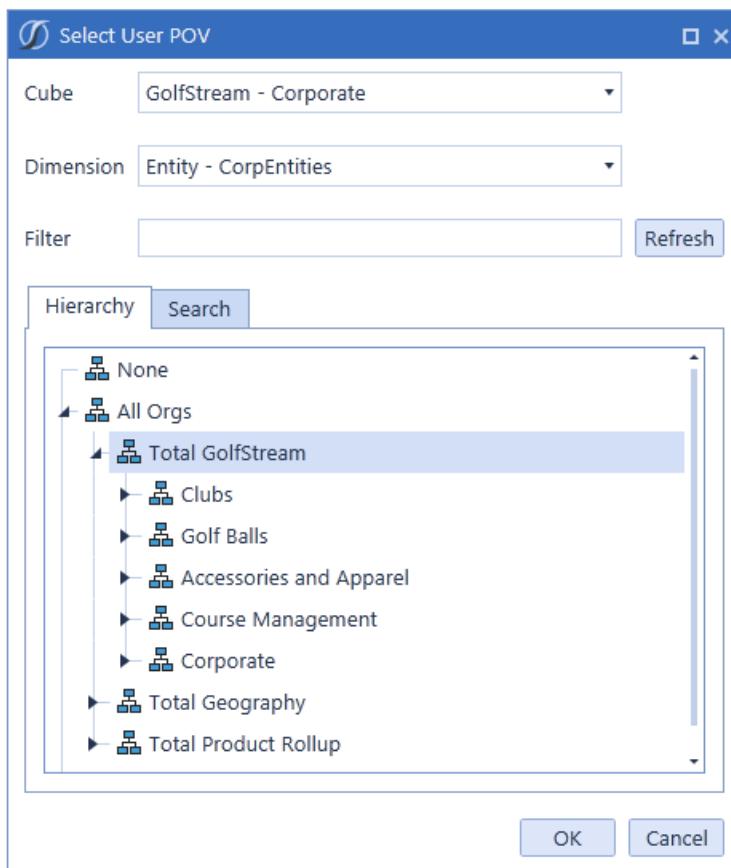
This is the Point of View for the whole application. This is set by the administrator and will not be active for the end user to update. This includes Scenario and Time.

### Workflow POV

This has the same configuration as the Workflow area of the OnePlace Tab. It will display the active Point of View; however it will not be active for the end user to update. This includes Workflow, Scenario, and Time. The Time displayed is based on the Time Dimension Profile associated with the Cube assigned to the Workflow Profile.

### Cube POV

This is active and available to be updated by the end user. Each Dimension will need to be set based on the information or activity a user needs to perform. Hover over any of the Dimensions and a tool tip will display the Dimension type. To update a Dimension, select one and a Select User POV box will appear. This box will give the user an opportunity to pick the Cube, Dimension, and the ability to apply a Member Filter and search.



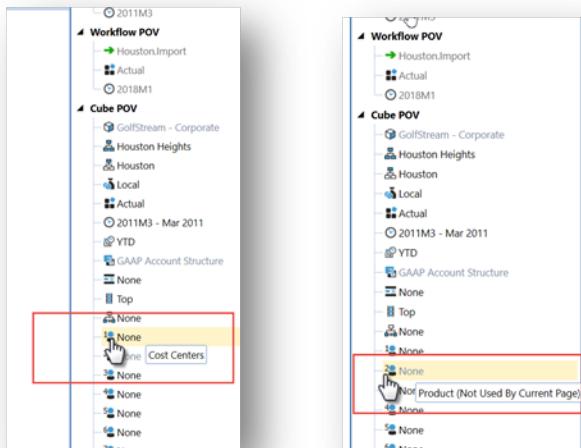
## Navigation

---

**TIP:** Save a commonly used POV by right clicking on the Cube POV and selecting Save Cube POV to Favorites. This saves the POV under Application|Documents|Users|(User Name)|Favorites allowing it to be used on any Cube View, grid, or Dashboard.

## User Defined Description – Point of View

The Point-of-View panel supports the new User Defined Descriptions. Hovering over a selectable point-of-view member will display the defined description. Dimensions which are fixed, not selectable, will display the defined description and append “Not Used by Current Page”. See Application Properties and then User Defined Dimensions (Descriptions) for more information.



## Navigating Mobile

### Mobile Toolbar



Click this to display or hide the OneStream Mobile Menu.



Click this to display or hide the OneStream Mobile POV context pane. The Mobile POV is derived from the OneStream POV. The Mobile POV is interactive and can be changed to view Cube Views or Workflow Status.

## OneStream Mobile Menu

### Home

Select Home in order to navigate to the pre-set home screen. See Settings below for more details on setting a customized home screen.

### Presentation

**NOTE:** OneStream application security applies to all Dashboards, Cube Views and Documents in Mobile. If users do not have access to particular reports in the application, they will not have access to them in the Mobile web interface.

### Dashboards

Select this to run Mobile Dashboards. Dashboards display several different views of OneStream data. These Mobile specific Dashboards are designed in the Application, but can be viewed from Mobile devices such as cell phones and tablets. See Application Dashboards in "Presenting Data With Books, Cube Views and Dashboards" on page 547.

### Cube Views

Select this to view any Cube View currently saved in OnePlace in the application. If users can change certain POV Members in the application, they can control those same Members in their Mobile POV in order to have multiple views of their Cube View data.

### Documents

Select this to launch and view the files saved in the application File Explorer. These documents could include items such as Extensible Documents or Excel analysis workbooks.

**NOTE:** Users cannot launch any type of process from the Mobile web interface. Processes include things such as running Business Rules, Calculations, Data Management Steps or Sequences, etc. Mobile Dashboards and Cube Views are read-only.

# Status

### Workflow

Select this to get the current Workflow Status of the Workflow set in the Application. The Workflow Status provides details on each Workflow Input Type and shows the percentage OK, In Process, Not Started, or Error. To view a different Workflow, click and select a new Workflow in the Workflow POV. Navigate back to the Workflow Status page to view the new Workflow's details. Workflow security applies to OneStream Mobile.

Workflow Status

- ▼ Houston
  - Process Cube Started
  - (no channel)
  - % OK : 25 % IP : 75 % NS : 0 % ERR : 0
- Houston.Import
  - Import Completed
  - Standard
  - % OK : 0 % IP : 0 % NS : 0 % ERR : 0
- Houston.Sales Detail
  - Completed
  - Standard
  - % OK : 0 % IP : 0 % NS : 0 % ERR : 0
- Houston.Forms
  - Input Forms Started
  - Standard
  - % OK : 0 % IP : 0 % NS : 0 % ERR : 0
- Houston.Journals
  - Input Journals Started
  - Standard
  - % OK : 0 % IP : 0 % NS : 0 % ERR : 0

### Servers

Select this for a read-only view of the OneStream Application Servers' Status. This provides details about each application server such as the Environment, Web Server Name and Connection Status.

## Navigation

---

### Activity

Select this to see all the task activities in the application. End Users can view their own tasks and administrators can view every task performed by any user.

## Settings

### Set Current Page as Home Page

Click this to make the current Mobile view the Mobile web interface's home page. This can be any of the displays under Presentation or Status. After a home page is set, click Home to be redirected to that page.

### Clear Home Page

Select this to clear the current home page.

### Set Page as Default for Users

Select this to make the current view the default home page for any new user.

# Collecting Data

OneStream can connect to and import data from any external system using direct database connections to the external system. Data can be collected from delimited data sources, forms, and Excel files using XFSetCell or Cube views. Connector Business Rules define the connection, data result sets, and drill-back option capabilities of an external data connection.

In this section you will learn about the various methods for collecting data.

## Data Sources

### Delimited Data Source

A Delimited Data Source with a separate column for Debit and Credit for the Amount Dimension is accomplished by using a Parser Complex Expression, or a Business Rule. First, assign the Debit column as the Amount. Next, create a Parser Complex Expression if this is a one-time occurrence, or create a Business Rule if it can be applied to numerous Data Sources, and assign it to the Amount Column. In the Complex Expression or Business Rule, check the value returned for the Debit. If it is empty, or 0.00, refer to the credit value with the method outlined below.

Assessing the credit value is possible in OneStream because the transformation engine provides an array list of all fields in the line. In the example below, Debit is column six, and Credit is column 7.

#### **Dim creditValue as String = api.Parser.DelimitedParsedValues(6)**

This is a zero based list, so index six is equal to column seven in the Data Source. Using this expression would only return the credit value, so apply the statement above to a broader Complex Expression.

### Connector Data Source

OneStream can connect and import data from any external system using direct database connection to the external system. This means data can be imported and processed all the way through the Workflow certification process without ever having to use a source system extract file.

## Connector Business Rule

A Connector Business Rule defines the connection, data result sets, and drill-back option capabilities of an external data connection. A Connector functions as a Business Rule called by a Data Source and reveals what information is required from an external system. See Business Rules in "Application Tools" on page 727 for an example of this rule.

## Connector Data Source

Fields from the external data query results are mapped to Dimensions creating a processing behavior similar to the behavior of a Delimited File. Using this mapping process enables a Connector Data Source to use all the same built processing capabilities available with file-based Data Sources. This capability enables the design of an external data Connector to be entirely focused on connecting to and reading data from an external source instead of focusing on integrating complex business logic. The specific business logic can be added to the Data Source Dimensions in the form of a Complex Expression or Business Rule. This design methodology will help with writing the Connector Business Rule in a way that requires very little maintenance by business users.

## Connector Information Request Types

### **GetFieldList**

This is called by the Data Source designer screen when the user selects a Connector Data Source or one of its defined Dimensions. A list of available fields in the external Data Source will be visible as a list of Vb.Net Strings [List(Of String)] is requested.

### **GetData**

This is called by the Import Workflow task when the Load and Transform button is clicked. The execution of a data query(s) that retrieves the row values for the chosen Workflow Unit is requested.

### **Fields**

The field names returned by this query must match the field names returned by the GetFieldList request.

### **Where Clause**

Typically the active Workflow Unit Time and Scenario values are converted to equivalent criteria values for the Time or Scenario of the external system.

### **Data Volume**

Consider loading summarized data rather than full transaction system data replication because drill back is provided for more detailed values.

### **GetDrillBackTypes**

Drill Back types can deliver results based on the different visualization types. This is called when a user double-clicks or right-clicks and selects Drill Back from a row in the source data load or drill down screens. A set of supported drill-back options to present to the end user as a list of DrillBackTypeInfo objects [List(Of DrillBackTypeInfo)] is requested. Drill Back types provide the Connector designer with the power to provide the end user with a menu list of drill back options.

### **DataGrid**

This presents a grid of data rows to the end user.

### **TextMessage**

This presents a text message to the end user.

### **WebUrl**

This presents a website or custom HTML web content to the end user.

### **WebUrlPopOutDefaultBrowser**

Opens a website or custom HTML web content in an external browser. From the Stage Import data grid, right-click on a data record, and select Drill Back. A dialog presents a menu of pre-configured Drill-back options. When you choose WebUrlPopOutDefaultBrowser, a standard browser session is launched, and you go to a web page based on variables.

### **FileViewer**

This presents file contents to the end user from one of three locations.

### **FileShareFile**

A file located in a folder in the OneStream File Share.

### **AppDBFile**

A file stored in an application database.

### **SysDBFile**

A file stored in a framework (System) database.

### **GetDrillBack**

This is called when a user selects a specific Drill Back type presented by the GetDrillBackTypes request. When this action is executed, the Business Rule arguments will contain a reference to the DrillBackTypeInfo object the user selected which allows the Connector designer to determine how to get proper information to display for the DrillBackTypeInfo.

## **Connector Integration Prerequisites**

The following items provide an overview of the major technology components involved in integrating external systems with deployment.

### Determine Source System Inventory

The first step in integrating various source systems is to determine all the ones needed. This includes:

Source System Location & Identification

### Database Type and Source System

Oracle, SQL, DB2, Syteline, Newstar, Lawson, PeopleSoft, Access, MAS500, etc.

**NOTE:** The requirement for Oracle Database integrations is that all Oracle Source System TNS Profile details need to be in place on each of the OneStream application servers.

### Data Query Method

Detailed Data Query, Data View, Stored Procedure, etc.

### Source System Drill Back Criteria (if required)

Detailed Data Query, Data View, Stored Procedure, etc.

### Source System Direct Access Credentials

A read-only type of access needs to be granted for the user account because the data from these external systems will be read. The read-only access should be granted against the production instance of the data source as the data queries will be used to tie out data and do not present any risk to the source system themselves.

### Source System 64-bit Client Data Provider

OneStream is a Microsoft .NET application with a 64-bit architecture. To communicate with any source system, a 64-bit source system client data provider needs to be available and installed on each OneStream application server. The source system's client data provider is what gives the ability to make an OLEDB or ODBC connection to the system.

### Determine Connection String

A connection string specifies information about a data source and the means of connecting to it. It is passed in code to an underlying driver or provider to initiate the connection. Whilst commonly used for a database connection, the data source could also be a spreadsheet or text file. The connection string may include attributes such as the name of the driver, server and database, as well as security information such as user name and password.

# Create a Connection String from the OneStream Application Server

1. Right-click the Desktop icon of the OneStream application server and select **New > Text Document**.
2. Name the document and change the file extension from txt to udl.



This creates a Data Link File to assist in the formation of the source system connection string.

3. Determine the DB Provider that the GL Source System is using (e.g. SQL, Oracle, etc.).
4. Determine the server name where the data resides for the GL Source System.
5. Determine the user name and password used to connect to the server for the GL Source System.
6. Determine the database name on the server where the GL Source System data resides.
7. Save the completed UDL file and then rename the extension back to txt from udl.
8. Open the text file to see the connection string provided.

A screenshot of a Windows Notepad window titled 'Test.txt - Notepad'. The menu bar includes File, Edit, Format, View, and Help. The window contains the following text:

```
[oledb]
; Everything after this line is an OLE DB initstring
Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Users\mbaranowski\Desktop\Database11.accdb;Mode=Read|Share Deny None;Persist Security Info=False
```

## Example Connection Strings

### SQL Server

Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=DBName;Data Source=SQLSERVERNAME

### ORACLE (11i or R12)

Provider=OraOLEDB.Oracle.1;Password=<xxxxx>;Persist Security Info=True;User ID=<username>;Data Source=frepro.world

## Collecting Data

---

### DB2

Provider=IBMDA400.DataSource.1;Password=<xxxxx>;Persist Security Info=True;User ID=OSuser;Data Source=HUTCH400;Use SQL Packages=True

### MS Access

Provider=Microsoft.ACE.OLEDB.12.0;Data Source=\\UNCFileShare\\DB1.accdb;Mode=Read|Share Deny None;Persist Security Info=False

## Determine the Data Query Method

To extract data from any source system, the data query method and facility need to be determined. Data can be queried through a SQL Query, a SQL View, or Stored Procedure. OneStream executes this request against the source system using the defined source system connection string and processes the returned results within OneStream.

For example, if directly pulling in Trial Balance Data is required, then the detailed query that currently makes up the existing Trial Balance Report would be necessary for OneStream to pull the same data.

### SQL Query

A SQL Query can be broken down into numerous elements, each beginning with a keyword. Although it is not necessary, a common convention is to write these keywords in all capital letters. The standard sections of a SQL Query are made up of the following four elements:

SELECT  
FROM  
WHERE  
ORDER BY

The example below is a SQL Query used to pull Trial Balance Data from several different tables in an Oracle Database:

```
SELECT
GL_SETS_OF_BOOKS.NAME
,GL_BALANCES.ACTUAL_FLAG
,GL_BALANCES.PERIOD_NAME
,GL_BALANCES.PERIOD_NUM
,GL_BALANCES.PERIOD_YEAR
,GL_CODE_COMBINATIONS.CODE_COMBINATION_ID
,GL_CODE_COMBINATIONS.SEGMENT1
,GL_CODE_COMBINATIONS.SEGMENT2
,GL_CODE_COMBINATIONS.SEGMENT3
,GL_CODE_COMBINATIONS.SEGMENT4
,GL_CODE_COMBINATIONS.SEGMENT5
,GL_CODE_COMBINATIONS.SEGMENT6
,GL_CODE_COMBINATIONS.SEGMENT7
,GL_CODE_COMBINATIONS.SEGMENT8
,GL_CODE_COMBINATIONS.SEGMENT9
```

## Collecting Data

---

```
,GL_CODE_COMBINATIONS SEGMENT10
, SUM( NVL(GL_BALANCES.BEGIN_BALANCE_DR,0) - NVL(GL_BALANCES.BEGIN_BALANCE_CR,0)) "OPEN BAL"
, NVL(GL_BALANCES.PERIOD_NET_DR,0) "DEBIT"
, NVL(GL_BALANCES.PERIOD_NET_CR,0) "CREDIT"
, SUM( NVL(GL_BALANCES.PERIOD_NET_DR,0) - NVL(GL_BALANCES.PERIOD_NET_CR,0)) "NET MOVEMENT"
, SUM(( NVL(GL_BALANCES.PERIOD_NET_DR,0) + NVL(GL_BALANCES.BEGIN_BALANCE_DR,0))) - SUM
    (NVL(GL_BALANCES.PERIOD_NET_CR,0)+NVL(GL_BALANCES.BEGIN_BALANCE_CR,0)) "CLOSE BAL"
,GL_BALANCES.CURRENCY_CODE
,GL_BALANCES.TRANSLATED_FLAG
,GL_BALANCES.TEMPLATE_ID
,FND_FLEX_VALUES_VL.FLEX_VALUE
,FND_FLEX_VALUES_VL.DESCRIPTION
,FND_FLEX_VALUES_VL.FLEX_VALUE_SET_ID

FROM
GL_BALANCES,
GL_CODE_COMBINATIONS,
GL_SETS_OF_BOOKS,
FND_FLEX_VALUES_VL

WHERE GL_CODE_COMBINATIONS.CODE_COMBINATION_ID = GL_BALANCES.CODE_COMBINATION_ID
AND GL_BALANCES.ACTUAL_FLAG = 'A'
AND GL_BALANCES.CURRENCY_CODE = GL_SETS_OF_BOOKS.CURRENCY_CODE
AND GL_BALANCES.LEDGER_ID = GL_SETS_OF_BOOKS.SET_OF_BOOKS_ID
AND GL_BALANCES TEMPLATE_ID IS NULL
AND GL_BALANCES.PERIOD_NAME = 'Jul-14'
AND FND_FLEX_VALUES_VL.FLEX_VALUE = GL_CODE_COMBINATIONS.SEGMENT4
AND FND_FLEX_VALUES_VL.FLEX_VALUE_SET_ID = '101432874'
AND GL_CODE_COMBINATIONS.SEGMENT2 IN (2050, 2100, 2200, 2300, 2400, 2500)

GROUP BY GL_SETS_OF_BOOKS.NAME
,GL_BALANCES.ACTUAL_FLAG
,GL_BALANCES.PERIOD_NAME
,GL_BALANCES.PERIOD_NUM
,GL_BALANCES.PERIOD_YEAR
,GL_CODE_COMBINATIONS.CODE_COMBINATION_ID
,GL_CODE_COMBINATIONS.SEGMENT1
,GL_CODE_COMBINATIONS.SEGMENT2
,GL_CODE_COMBINATIONS.SEGMENT3
,GL_CODE_COMBINATIONS.SEGMENT4
,GL_CODE_COMBINATIONS.SEGMENTS5
,GL_CODE_COMBINATIONS.SEGMENT6
,GL_CODE_COMBINATIONS.SEGMENT7
,GL_CODE_COMBINATIONS.SEGMENT8
,GL_CODE_COMBINATIONS.SEGMENT9
,GL_CODE_COMBINATIONS.SEGMENT10
,NVL(GL_BALANCES.PERIOD_NET_DR,0)
,NVL(GL_BALANCES.PERIOD_NET_CR,0)
,GL_BALANCES.CURRENCY_CODE
,GL_BALANCES.TRANSLATED_FLAG
,GL_BALANCES.TEMPLATE_ID
,FND_FLEX_VALUES_VL.FLEX_VALUE
,FND_FLEX_VALUES_VL.DESCRIPTION
,FND_FLEX_VALUES_VL.FLEX_VALUE_SET_ID
```

## Collecting Data

---

```
HAVING SUM(( NVL(GL_BALANCES.PERIOD_NET_DR,0) + NVL(GL_BALANCES.BEGIN_BALANCE_DR,0))) -  
SUM(NVL(GL_BALANCES.PERIOD_NET_CR,0)+NVL(GL_BALANCES.BEGIN_BALANCE_CR,0)) <> 0
```

### SQL View

In many cases, creating a SQL View of data to provide information to OneStream is a more preferred option and typically simplifies the complexity of the query.

In the example below, the customer can combine several data tables required in the source system, and present the data in one View for OneStream to query:

```
SELECT  
SEGMENT1 As Entity  
SEGMENT2 As Establishment  
SEGMENT3 As France_Account  
SEGMENT4 As US_Account  
SEGMENTS5 As Cost_Center  
SEGMENT6 As Family  
SEGMENT7 As Product_Line  
SEGMENT8 As Interco  
SEGMENT9 As Future  
PERIOD_YEAR As Year  
PERIOD_MONTH As Month  
CURRENCY_CODE As Currency_Code  
CLOSE_NET_BALANCE As Net_Balance  
SET_OF_BOOKS_ID As Set_Of_Books_ID  
FROM APPS.XXSWM_ONESTREAM_GL_BALANCES
```

### Stored Procedure

The example below is a SQL Stored Procedure used to pull Trial Balance Data from several different tables in a SQL Database.

In this example, the Entity, Year, and Period are passed to the Stored Procedure:

```
spGLCalcTrialBalance 'ASCC', '2013', 6
```

### Apply Connection String to XFAppServerConfig.xml File

When the connection string is created, then the database connections can be centralized in the Server Configuration under the App Server Configuration File. Under Databases, click on (Collection) for Database Server Connections and the Database Server Connections will appear. The string will then be placed in the Connection String under Connection String Settings. The name of the connection string will be used as part of the source connector.

## Defining External Data Connections

Application Server Configuration File  
Creating Named External ODBC / OLEDB Connection

## Collecting Data

---

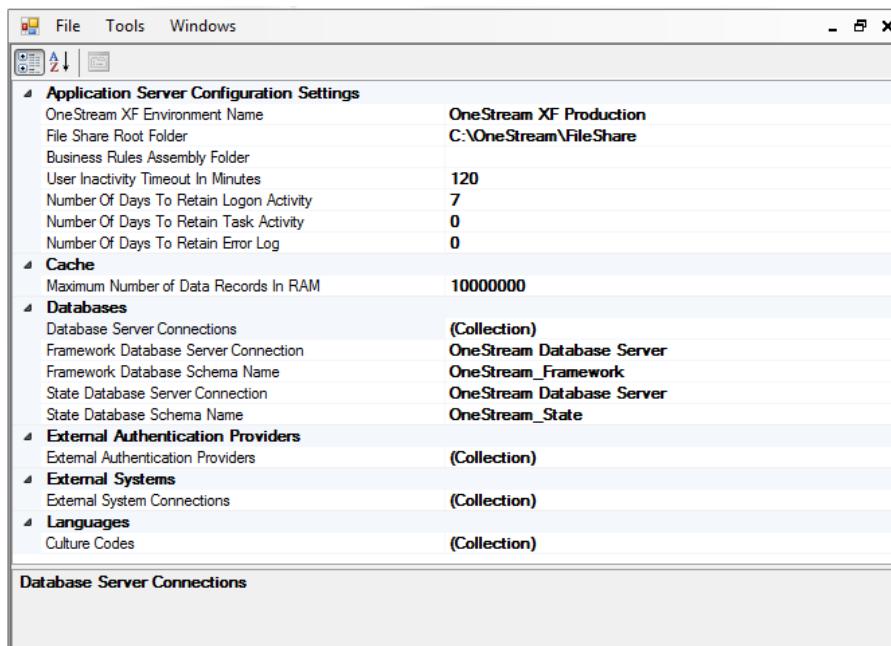
### Step 1: Required ODBC/OLEDB Connection Software

Any client ODBC/OLEDB drivers must be installed on each application server for the OneStream application to make a connection to the external database. This way the administrator knows what type of database engine contains which Data Source.

### Step 2: Creating the Connection String

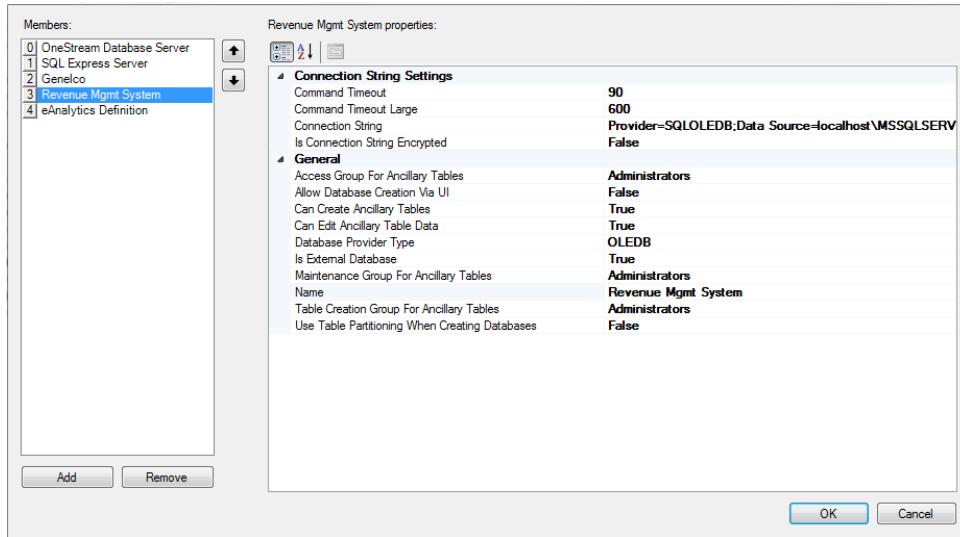
The application server configuration file must be modified to add a named external database connection that can be used by the Connector Business Rule and custom reports.

Example of the Server Configuration Utility:



## Collecting Data

---



### Step 3: Creating an External Database Test Query

The best way to prototype the queries needed to create a Connector Business Rule is to create a set of Dashboard Data Adapters to be used as a test bed.

As a best practice, create a new Dashboard Maintenance Unit named EXS The Connector Name. The prefix EXS stands for External System and will provide administrators with an immediate understanding of the Maintenance Unit's contents. The three steps below explain how to create this.

#### Step 1

Create a new Data Adapter for each type of query needed to prototype (GetFieldList, SelectData, Drill Back, etc.)

Example of a Data Adapter being used to get all fields in the source table of the external database connection:

## Collecting Data

---

General (Data Adapter)	
Name	InvoiceDocumentDetailFieldList
Description	List Of Fields in Source Table
Maintenance Unit	External System (Revenue Mgmt Houston)
Data Source	
Command Type	SQL
Database Location	External
External Database Connection	Revenue Mgmt System
SQL Query	Select Top(1) * From InvoiceDocumentDetail
Results Table Name	FieldNameList

### Step 2

Click  in the Dashboard administration toolbar to test the query.

### Step 3

Evaluate the results of the query. The Data Adapter test only returns a small subset of rows from the query, but it specifies the actual number of rows that will return during an actual query execution.

 Data Preview - InvoiceDocumentDetailFieldList

Original Query

```
Select Top(1) * From InvoiceDocumentDetail
```

Substituted Query

```
Select Top(1) * From InvoiceDocumentDetail
```

Data Table

FieldNameList

TransID	PlantCode	CustId	CustName	InvNo	InvYear	InvMonth
e5142bed-1ad2-4463-bc44-00012dc85fdc	H200	DE1475	Double Eagle	I17-H200-DE1475-2011M1	2011	M1

Total Number Of Rows: 1

# Building Data Connectors

ODBC / OLEDB Connectors

### **GetFieldList**

Select Query against the external database. There will be a manual list of strings returned for each field.

### **GetData**

The selected statement should match GetFieldList. Add criteria for Scenario and Time and map the OneStream Workflow Unit Scenario and Time values to corresponding values in the source system as a Where Clause criteria value.

### **GetDrillBackTypes**

This shows the set of drill back options provided to the user.

### **GetDrillBack**

This executes the selected drill back type for the current source data row.

### **Custom API Connectors**

Uses OneStream's External Server Technology. IIS needs to be recycled on all application servers and followed by all web servers after adding the external named connection.

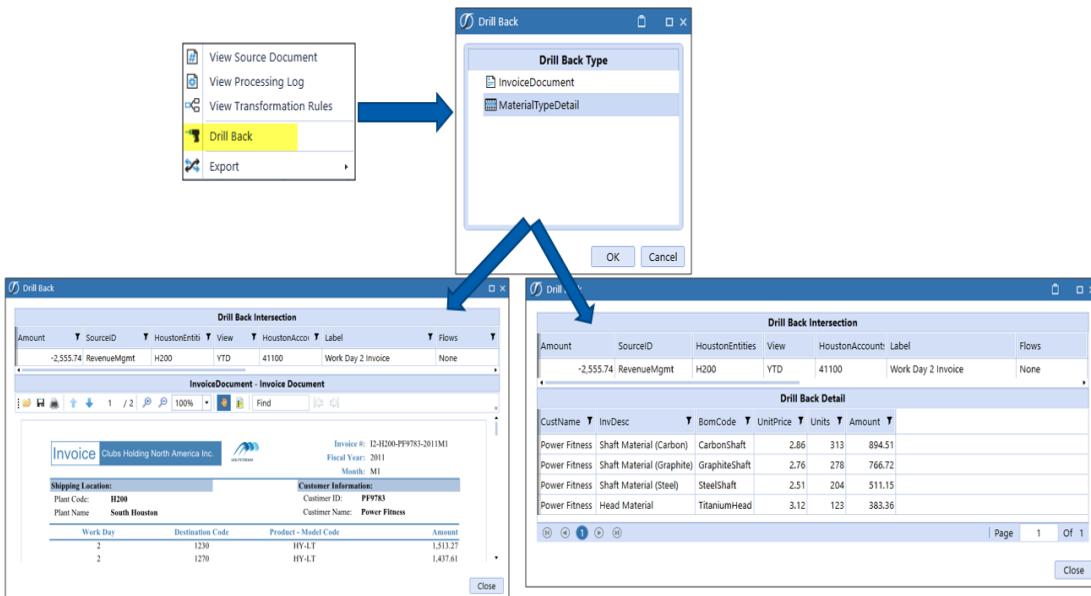
## Drill Back

Using a SQL connector allows a user to drill back to a source system and show detailed records from a document, PDF, website. The Connector Data Source, configured by the author, provides a menu of data viewing options such as Year to Date, Month to Date, Invoice Documents or Material Type Detail. Utilizing this feature can reduce the amount of data imported into the Financial Model by allowing analysis to occur at the source system.

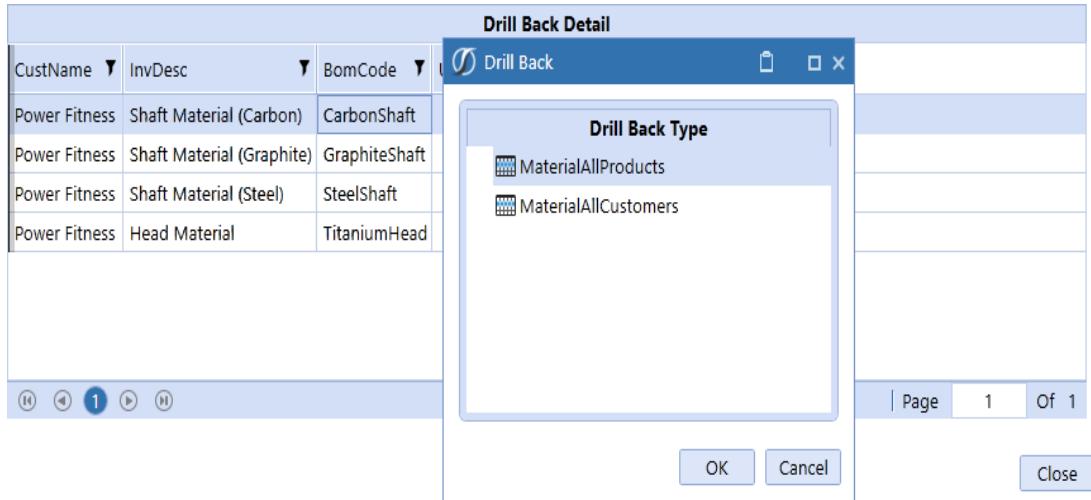
### **Viewing Data**

After data is loaded into the Stage, a user can right-click on a data row and select Drill Back. This will bring up the pre-configured options from which the user can choose.

## Collecting Data



If more detail is needed, another level of Drill Back can be performed. This is configured in the Connector Business Rule and can drill back and around source systems. These nested drill paths can provide as much detail as an application requires.



# Key API, Args, or BRAPI Examples

```

--> 19  Namespace OneStream.BusinessRule.Connector.RevenueMgmtHouston
20  <--> 20  Public Class MainClass
21  <--> 21  Public Function Main(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Transformer, ByVal args As ConnectorArgs) As Object
22  Try
23      'Get the query information
24      Dim connectionString As String = GetConnectionString(si, globals, api)
25
26      'Get the Field name list or load the data
27      Select Case args.ActionType
28          Case Is = ConnectorActionTypes.GetFieldList
29              'Return Field Name List
30              Dim fieldListSQL As String = GetFieldListSQL(si, globals, api)
31              Return api.Parser.GetFieldNameListForSQLQuery(si, DbProviderType.OLEDB, connectionString, true, fieldListSQL, false)
32
33          Case Is = ConnectorActionTypes.GetData
34              'Process Data
35              Dim sourceDataSQL As String = GetSourceDataSQL(si, globals, api)
36              api.Parser.ProcessSQLQuery(si, DbProviderType.OLEDB, connectionString, true, sourceDataSQL, false, api.ProcessInfo)
37              Return Nothing
38
39          Case Is = ConnectorActionTypes.GetDrillBackTypes
40              'Return the list of Drill Types (Options) to present to the end user
41              Return Me.GetDrillBackTypeList(si, globals, api, args)
42
43          Case Is = ConnectorActionTypes.GetDrillBack
44              'Process the specific Drill-Back type
45              Return Me.GetDrillBack(si, globals, api, args, args.DrillBackType.DisplayType, connectionString)
46      End Select
47
48      Catch ex As Exception
49          Throw ErrorHandler.LogError(si, New XFEException(si, ex))
50      End Try
51  End Function
52

53  'Create a Connection string to the External Database
54  Private Function GetConnectionString(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Transformer) As String
55      Try
56
57          'Connection String Method
58          '-----
59          Dim connection As New Text.StringBuilder'
60          connection.Append("Provider=SQLOLEDB.1;")
61          connection.Append("Data Source=LocalHost\MSSQLSERVER2008;")
62          connection.Append("Initial Catalog=SampleData;")
63          connection.Append("Integrated Security=SSPI")
64          Return connection.ToString
65
66          'Named External Connection
67          '-----
68          Return "Revenue Mgmt System"
69
70      Catch ex As Exception
71          Throw ErrorHandler.LogError(si, New XFEException(si, ex))
72      End Try
73  End Function
74

75  'Create the field list SQL Statement
76  Private Function GetFieldListSQL(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Transformer) As String
77      Try
78          'Create the SQL Statement
79          Dim sql As New Text.StringBuilder
80
81          sql.Append("SELECT Top(1)")
82          sql.Append("TransID, PlantCode, CustId, CustName, InvNo, InvYear, InvMonth, InvDesc, GLAccount, WorkDay, ProdModel, BomCode, UnitPrice, Units, Amount, DestinationCode")
83          sql.Append("FROM InvoiceDocumentDetail ")
84
85          Return sql.ToString
86
87      Catch ex As Exception
88          Throw ErrorHandler.LogError(si, New XFEException(si, ex))
89      End Try
90  End Function

```

## Collecting Data

```

'Create the data load SQL Statement
Private Function GetSQLStatement(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Transformer) As String
    Try
        'Create the SQL Statement
        Dim statement As New Text.StringBuilder
        Dim selectClause As New Text.StringBuilder
        Dim fromClause As New Text.StringBuilder
        Dim whereClause As New Text.StringBuilder
        Dim orderByClause As New Text.StringBuilder

        selectClause.Append("SELECT ")
        selectClause.Append("TransID, PlantCode, CustId, CustName, InvNo, InvYear, InvMonth, WorkDay, ProdModel, BomCode, UnitPrice, Units, Amount, Destina")
        fromClause.Append("FROM InvoiceDocumentDetail ")
        whereClause.Append("WHERE ")
        whereClause.Append("Get THE YEAR from the current XF workflow unit TimeKey")
        whereClause.Append("(")
        whereClause.Append("InvYear = " & TimeDimHelper.GetYearFromId(api.WorkflowUnitPk.TimeKey).ToString)
        whereClause.Append(")")

        'Get the MONTH from the current XF workflow unit TimeKey
        whereClause.Append(" And ")
        whereClause.Append("(")
        whereClause.Append("InvMonth = 'M' & TimeDimHelper.GetSubComponentsFromId(api.WorkflowUnitPk.TimeKey).Month.ToString & '')")
        whereClause.Append(")")

        'Select Plant Codes
        whereClause.Append(" And ")
        whereClause.Append("(")
        whereClause.Append("PlantCode IN('H200','H210')")
        whereClause.Append(")")

        orderByClause.Append("ORDER BY ")
        orderByClause.Append("PlantCode, Custid, WorkDay, ProdModel, DestinationCode")

        'Create the full SQL Statement
        statement.Append(selectClause.ToString)
        statement.Append(fromClause.ToString)
        statement.Append(whereClause.ToString)
        statement.Append(orderByClause.ToString)
    End Function

'Create the drill back option list
Private Function GetDrillBackTypeList(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Transformer, ByVal args As ConnectorArgs) As List(Of DrillBackTypeInfo)
    Try
        'Create the SQL Statement
        Dim drillTypes As New List(Of DrillBackTypeInfo)

        drillTypes.Add(New DrillBackTypeInfo(ConnectorDrillBackDisplayTypes.FileShareFile, New NameAndDesc("Invoice Document", "Invoice Document")))
        drillTypes.Add(New DrillBackTypeInfo(ConnectorDrillBackDisplayTypes.DataGrid, New NameAndDesc("Material Type Detail", "Material Type Detail")))

        Return drillTypes
    Catch ex As Exception
        Throw ErrorHandler.LogWrite(si, New XFEException(si, ex))
    End Try
End Function

'Execute specific drill back type
Private Function GetDrillBack@(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Transformer, ByVal args As ConnectorArgs, ByVal drillBackType As ConnectorDrillBackDisplayTypes, ByVal connectionstr As String)
    Try
        Select case drillBackType
        Case Is = ConnectorDrillBackDisplayTypes.FileSharefile
            'Show Fileshare File
            Dim drillBackSql As New DrillBackSqlInfo
            drillBackSql.DisplayType = ConnectorDrillBackDisplayTypes.Filesharefile
            drillBackSql.Info.DocumentPath = Me.getDrillbackDocPath(si, globals, api, args)
            Return drillBackSql
        Case Is = ConnectorDrillBackDisplayTypes.DataGridView
            'Return Drill Back Detail
            Dim drillBackSQL As String = GetDrillbackSQL(si, globals, api, args)
            Dim drillBackInfo As New DrillBackResultInfo
            drillBackInfo.DisplayType = ConnectorDrillBackDisplayTypes.DataGridView
            drillBackInfo.DataSource = api.Parser.GetDataTableForSQLQuery(si, 0, ProviderType.OLEDB, connectionString, True, drillBackSQL, False, args.PageSize, args.PageNumber)
            Return drillBackInfo
        Case Else
            Return Nothing
        End Select
    Catch ex As Exception
        Throw ErrorHandler.LogWrite(si, New XFEException(si, ex))
    End Try
End Function

'Creates the drill back Document Path
Private Function GetDrillbackDocPath@(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Transformer, ByVal args As ConnectorArgs) As String
    Try
        'Get the values for the source row that we are drilling back to
        Dim sourcevalues As Dictionary(Of String, Object) = api.Parser.GetFieldValuesForSourceDataRow(si, args.RowID)
        If (Not sourcevalues Is Nothing) And (sourcevalues.Count > 0) Then
            Return "Applications/GolfStream_v24/DataManagement/RevenuegmtInvoices/" & sourcevalues.Item(StageConstants.MasterDimensionNames.Attribute1).ToString & ".pdf"
        Else
            Return String.Empty
        End If
    Catch ex As Exception
        Throw ErrorHandler.LogWrite(si, New XFEException(si, ex))
    End Try
End Function

```

## Collecting Data

---

```
200     'Create the drill back SQL Statement
201     Private Function GetDrillbackSQL(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Transformer, ByVal args As ConnectorArgs) As String
202         Try
203             'Get the values for the source row that we are drilling back to
204             Dim sourcevalues As Dictionary(Of String, Object) = api.Parser.GetFieldValuesForSourceDataRow(si, args.RowID)
205             If (Not sourcevalues Is Nothing) And (sourcevalues.Count > 0) Then
206
207                 Dim statement As New Text.StringBuilder
208                 Dim selectClause As New Text.StringBuilder
209                 Dim fromClause As New Text.StringBuilder
210                 Dim whereClause As New Text.StringBuilder
211                 Dim orderByClause As New Text.StringBuilder
212
213                 'Create the SQL Statement
214                 selectClause.Append("SELECT ")
215                 selectClause.Append("CustName, InvDesc, BomCode, UnitPrice, Units, Amount ")
216
217                 fromClause.Append("FROM InvoiceMaterialDetail ")
218
219                 whereClause.Append("WHERE ")
220                 'Get the YEAR from the source record
221                 whereClause.Append("(")
222                 whereClause.Append("InvYear = " & TimeDimHelper.GetYearFromId(sourcevalues.Item(StageTableFields.StageSourceData.DimworkFlowTimeKey).ToString))
223                 whereClause.Append(")")
224
225                 'Get the MONTH from the source record
226                 whereClause.Append(" And ")
227                 whereClause.Append("(")
228                 whereClause.Append("InvMonth = 'M" & TimeDimHelper.GetSubComponentsFromId(sourcevalues.Item(StageTableFields.StageSourceData.DimworkflowTimeKey)).Month.ToString & "'")
229                 whereClause.Append(")")
230
231                 whereClause.Append(" And ")
232                 whereClause.Append("(")
233                 whereClause.Append("PlantCode = '" & sourcevalues.Item(StageConstants.MasterDimensionNames.Entity).ToString & "'")
234                 whereClause.Append(")")
235
236                 whereClause.Append(" And ")
237                 whereClause.Append("(")
238                 whereClause.Append("InvNo = '" & sourcevalues.item(Stageconstants.MasterDimensionNames.Attribute1).ToString & "'")
239                 whereClause.Append(")")
240
241                 whereClause.Append(" And ")
242                 whereClause.Append("(")
243                 whereClause.Append("ProdModel = '" & sourcevalues.item(Stageconstants.MasterDimensionNames.UO2).ToString & "'")
244                 whereClause.Append(")")
245
246
247                 whereClause.Append(" And ")
248                 whereClause.Append("(")
249                 whereClause.Append("DestinationCode = '" & sourcevalues.item(StageConstants.MasterDimensionNames.UO3).ToString & "'")
250                 whereClause.Append(")")
251
252                 whereClause.Append(" And ")
253                 whereClause.Append("(")
254                 whereClause.Append("CustID = '" & sourcevalues.item(StageConstants.MasterDimensionNames.UD4).ToString & "'")
255                 whereClause.Append(")")
256
257                 orderByClause.Append("ORDER BY ")
258                 orderByClause.Append("BomCode")
259
260                 'Create the full SQL Statement
261                 statement.Append(selectClause.ToString)
262                 statement.Append(fromClause.ToString)
263                 If args.ClientFilterRequest.Length > 0 Then
264                     statement.Append(whereClause.ToString)
265                     statement.Append(" And ")
266                     statement.Append(args.ClientFilterRequest)
267                 Else
268                     statement.Append(whereClause.ToString)
269                 End If
270                 If args.ClientSortRequest.Length > 0 Then
271                     statement.Append(args.ClientSortRequest)
272                 Else
273                     statement.Append(orderByClause.ToString)
274                 End If
275                 ErrorHandler.LogMessage(si, statement.ToString)
276                 Return statement.ToString
277             Else
278                 Return String.Empty
279             End If
280         Catch ex As Exception
281             Throw ErrorHandler.LogError(si, New XException(si, ex))
282         End Try
283     End Class
284 End Namespace
285
```

```
Namespace OneStream.BusinessRule.Connector.RevenueMgmtHouston
    Public Class MainClass
        Public Function Main(ByVal si As SessionInfo, ByVal globals As BRGlobals,
ByVal api As Transformer, ByVal args As ConnectorArgs) As Object
            Try
                'Get the query information

```

## Collecting Data

---

```
Dim connectionString As String = GetConnectionString(si, globals, api)

'Get the Field name list or load the data
Select Case args.ActionType
    Case Is = ConnectorActionTypes.GetFieldList
        'Return Field Name List
        Dim fieldListSQL As String = GetFieldListSQL(si, globals, api)
        Return api.Parser.GetFieldNameListForSQLQuery(si, DbProviderType.OLEDB,
connectionString, true, fieldListSQL, false)

    Case Is = ConnectorActionTypes.GetData
        'Process Data
        Dim sourceDataSQL As String = GetSourceDataSQL(si, globals, api)
        api.Parser.ProcessSQLQuery(si, DbProviderType.OLEDB, connectionString, true,
sourceDataSQL, false, api.ProcessInfo)
        Return Nothing

    Case Is = ConnectorActionTypes.GetDrillBackTypes
        'Return the list of Drill Types (Options) to present to the end user
        Return Me.GetDrillBackTypeList(si, globals, api, args)

    Case Is = ConnectorActionTypes.GetDrillBack
        'Process the specific Drill-Back type
        Return Me.GetDrillBack(si, globals, api, args, args.DrillBackType.DisplayType,
connectionString)
End Select

Catch ex As Exception
    Throw ErrorHandler.LogError(si, New XFException(si, ex))
End Try
End Function

'Create a Connection string to the External Database
Private Function GetConnectionString(ByVal si As SessionInfo, ByVal globals As BRGlobals,
    ByVal api As Transformer) As String
    Try

        'Connection String Method
        '-----
        Dim connection As New Text.StringBuilder'
        connection.Append("Provider=SQLOLEDB.1;")
        connection.Append("Data Source=LocalHost\MSSQLSERVER2008;")
        connection.Append("Initial Catalog=SampleData;")
        connection.Append("Integrated Security=SSPI")
        Return connection.ToString

        'Named External Connection
        '-----
        Return "Revenue Mgmt System"

    Catch ex As Exception
        Throw ErrorHandler.LogError(si, New XFException(si, ex))
    End Try
End Function
```

## Collecting Data

---

```
'Create the field list SQL Statement
Private Function GetFieldListSQL(ByVal si As SessionInfo, ByVal globals As BRGlobals,
    ByVal api As Transformer) As String
    Try
        'Create the SQL Statement
        Dim sql As New Text.StringBuilder

        sql.Append("SELECT Top(1)")
        sql.Append("TransID, PlantCode, CustId, CustName, InvNo,
InvYear, InvMonth, InvDesc, GLAccount, WorkDay, ProdModel, BomCode, UnitPrice, Units,
Amount, DestinationCode ")
        sql.Append("FROM InvoiceDocumentDetail ")

        Return sql.ToString

    Catch ex As Exception
        Throw ErrorHandler.LogError(si, New XFException(si, ex))
    End Try
End Function

'Create the data load SQL Statement
Private Function GetSourceDataSQL(ByVal si As SessionInfo, ByVal globals As BRGlobals,
    ByVal api As Transformer) As String
    Try
        'Create the SQL Statement
        Dim statement As New Text.StringBuilder
        Dim selectClause As New Text.StringBuilder
        Dim fromClause As New Text.StringBuilder
        Dim whereClause As New Text.StringBuilder
        Dim orderByClause As New Text.StringBuilder

        selectClause.Append("SELECT ")
        selectClause.Append("TransID, PlantCode, CustId, CustName,
InvNo, InvYear, InvMonth, InvDesc, GLAccount, WorkDay, ProdModel, BomCode, UnitPrice, Units, Amount,
DestinationCode ")

        fromClause.Append("FROM InvoiceDocumentDetail ")

        whereClause.Append("WHERE ")
        'Get the YEAR from the current XF Workflow Unit TimeKey
        whereClause.Append("(")
        whereClause.Append("InvYear = " & TimeDimHelper.GetYearFromId
(api.WorkflowUnitPk.TimeKey).ToString)
        whereClause.Append(")")

        'Get the MONTH from the current XF Workflow Unit TimeKey
        whereClause.Append(" And ")
        whereClause.Append("(")
        whereClause.Append("InvMonth = 'M" & TimeDimHelper.GetSubComponentsFromId
(api.WorkflowUnitPk.TimeKey)
.Month.ToString & "'")
        whereClause.Append(")")

    End Try
End Function
```

## Collecting Data

---

```
'Select Houston Plant Codes
whereClause.Append(" And ")
whereClause.Append("(")
whereClause.Append("PlantCode IN('H200','H210')")"
whereClause.Append(") ")

orderByClause.Append("ORDER BY ")
orderByClause.Append("PlantCode, CustId, WorkDay, ProdModel, DestinationCode")

'Create the full SQL Statement
statement.Append(selectClause.ToString)
statement.Append(fromClause.ToString)
statement.Append(whereClause.ToString)
statement.Append(orderByClause.ToString)

Return statement.ToString

Catch ex As Exception
    Throw ErrorHandler.LogWrite(si, New XFException(si, ex))
End Try
End Function

'Create the drill back options list
Private Function GetDrillBackTypeList(ByVal si As SessionInfo,
 ByVal globals As BRGlobals, ByVal api As Transformer, ByVal args As ConnectorArgs) As List(Of
DrillBackTypeInfo)
Try
    'Create the SQL Statement
    Dim drillTypes As New List(Of DrillBackTypeInfo)

    drillTypes.Add(New DrillBackTypeInfo(ConnectorDrillBackDisplayTypes.FileShareFile,
New NameAndDesc("Invoice Document","Invoice Document")))
    drillTypes.Add(New DrillBackTypeInfo(ConnectorDrillBackDisplayTypes.DataGrid,
New NameAndDesc("Material Type Detail","Material Type Detail")))

    Return drillTypes

    Catch ex As Exception
        Throw ErrorHandler.LogWrite(si, New XFException(si, ex))
    End Try
End Function

'Execute specific drill back type
private Function GetDrillBack(ByVal si As SessionInfo, ByVal globals As BRGlobals,
ByVal api As Transformer,
ByVal args As ConnectorArgs, ByVal drillBackType As ConnectorDrillBackDisplayTypes,
ByVal connectionString as String) As DrillBackResultInfo
Try
    Select case drillBackType
        case is = ConnectorDrillBackDisplayTypes.FileShareFile
```

## Collecting Data

---

```
'Show FileShare File
Dim drillBackInfo as new DrillBackResultInfo
drillBackInfo.DisplayType = ConnectorDrillBackDisplayTypes.FileShareFile
drillBackInfo.DocumentPath = Me.GetDrillBackDocPath(si, globals, api, args)
return drillBackInfo

case is = ConnectorDrillBackDisplayTypes.DataGrid
'Return Drill Back Detail
Dim drillBackSQL As String = GetDrillBackSQL(si, globals, api, args)
Dim drillBackInfo as new DrillBackResultInfo
drillBackInfo.DisplayType = ConnectorDrillBackDisplayTypes.DataGrid
drillBackInfo.DataTable = api.Parser.GetXFDataTableForSQLQuery(si,
DbProviderType.OLEDB, connectionString, true, drillBackSQL,
false, args.PageSize, args.PageNumber)
return drillBackInfo

case else
return Nothing
End Select

Catch ex As Exception
Throw ErrorHandler.LogWrite(si, New XFException(si, ex))
End Try
End Function

'Create the drill back Document Path
Private Function GetDrillBackDocPath(ByVal si As SessionInfo,
 ByVal globals As BRGlobals, ByVal api As Transformer, ByVal args As ConnectorArgs) As String
Try
'Get the values for the source row that we are drilling back to
Dim sourceValues as Dictionary(Of string, Object) =
api.Parser.GetFieldValuesForSourceDataRow(si, args.RowID)
If (Not sourceValues Is Nothing) And (sourceValues.Count > 0) then
Return "Applications/GolfStream_v24/DataManagement/RevenueMgmtInvoices/"
& sourceValues.Item(StageConstants.MasterDimensionNames.Attribute1).ToString & ".pdf"
Else
Return String.Empty
End If
Catch ex As Exception
Throw ErrorHandler.LogWrite(si, New XFException(si, ex))
End Try
End Function

'Create the drill back SQL Statement
Private Function GetDrillBackSQL(ByVal si As SessionInfo, ByVal globals As BRGlobals,
 ByVal api As Transformer, ByVal args As ConnectorArgs) As String
Try
'Get the values for the source row that we are drilling back to
Dim sourceValues as Dictionary(Of string, Object) =
api.Parser.GetFieldValuesForSourceDataRow(si, args.RowID)
If (Not sourceValues Is Nothing) And (sourceValues.Count > 0) then
```

## Collecting Data

---

```
Dim statement As New Text.StringBuilder
Dim selectClause As New Text.StringBuilder
Dim fromClause as New Text.StringBuilder
Dim whereClause as New Text.StringBuilder
Dim orderByClause as New Text.StringBuilder

'Create the SQL Statement
selectClause.Append("SELECT ")
selectClause.Append("CustName, InvDesc, BomCode, UnitPrice, Units,Amount ")

fromClause.Append("FROM InvoiceMaterialDetail ")

whereClause.Append("WHERE ")
'Get the YEAR from the source record
whereClause.Append("(")
whereClause.Append("InvYear = " & TimeDimHelper.GetYearFromId(sourceValues.Item
(StageTableFields.StageSourceData.DimWorkflowTimeKey).ToString()))
whereClause.Append(")")

'Get the MONTH from the source record
whereClause.Append(" And ")
whereClause.Append("(")
whereClause.Append("InvMonth = 'M" & TimeDimHelper.GetSubComponentsFromId
(sourceValues.Item
(StageTableFields.StageSourceData.DimWorkflowTimeKey))
.Month.ToString & "'")
whereClause.Append(")")

whereClause.Append(" And ")
whereClause.Append("(")
whereClause.Append("PlantCode = '" & sourceValues.Item
(StageConstants.MasterDimensionNames.Entity)
.ToString & "'")
whereClause.Append(")")

whereClause.Append(" And ")
whereClause.Append("(")
whereClause.Append("InvNo = '" & sourceValues.Item
(StageConstants.MasterDimensionNames.Attribute1)
.ToString & "'")
whereClause.Append(")")

whereClause.Append(" And ")
whereClause.Append("(")
whereClause.Append("ProdModel = '" & sourceValues.Item
(StageConstants.MasterDimensionNames.UD2).ToString& "'")
whereClause.Append(")")

whereClause.Append(" And ")
whereClause.Append("(")
whereClause.Append("DestinationCode = '" & sourceValues.Item
(StageConstants.MasterDimensionNames.UD3).
ToString & "'")
```

```
whereClause.Append(")")

whereClause.Append(" And ")
whereClause.Append("(")
whereClause.Append("CustID = '" &sourceValues.Item
(StageConstants.MasterDimensionNames.UD4)
.ToString & "'")
whereClause.Append(")")

orderByClause.Append("ORDER BY ")
orderByClause.Append("BomCode")

'Create the full SQL Statement
statement.Append(selectClause.ToString)
statement.Append(fromClause.ToString)
If args.ClientFilterRequest.length > 0 then
    statement.Append(whereClause.ToString)
    statement.Append(" And ")
    statement.Append(args.ClientFilterRequest)
Else
    statement.Append(whereClause.ToString)
End If
If args.ClientSortRequest.Length > 0 then
    statement.Append(args.ClientSortRequest)
Else
    statement.Append(orderByClause.ToString)
End if
'ErrorHandler.LogMessage(si, statement.ToString)
Return statement.ToString
Else
    Return String.Empty
End If
Catch ex As Exception
    Throw ErrorHandler.LogWrite(si, New XFException(si, ex))
End Try
End Function
End Class
End Namespace
```

## Loading Cell Text as an Annotation Data Attachment

To map cell text comments from a file into OneStream, here are a few tips to get started:

1. On the Cube's Integration Tab, ensure that the TextValue field is enabled for the desired Scenario Type. This is needed to import the actual text.
2. Create a Data Source to import data for the Scenario Type.

**NOTE:** TextValue is one of the mappable Dimensions.

3. Hardcode the View Dimension to import to the Annotation Member, or one of the other View Dimension Comment Members such as VarianceExplanation.
4. Map the comment text to the TextValue Dimension.
5. There may not be an Amount to bring in but select a column that has a decimal value in each row and a comment and link the Data Source to that column. These numbers will come into the Stage but will not end up in the Cube because they are mapped to an Annotation-type View Member.

# Forms

## Forms Channel Workflow

To minimize form maintenance, Cube View and Excel XFSetCell updates are not tied to specific Forms. Association is at the Input Type level, not the individual Form level. The Forms Input Type determines if you can update data from Excel. If the Forms channel is completed, but the process is not certified, you can import data from Excel using XFSetCell or a Cube View. If a cell is updated, the Analytic Engine traces the cell by:

1. Identifying the Workflow that owns the Entity.
2. Checking the Input Child Workflow Profile.

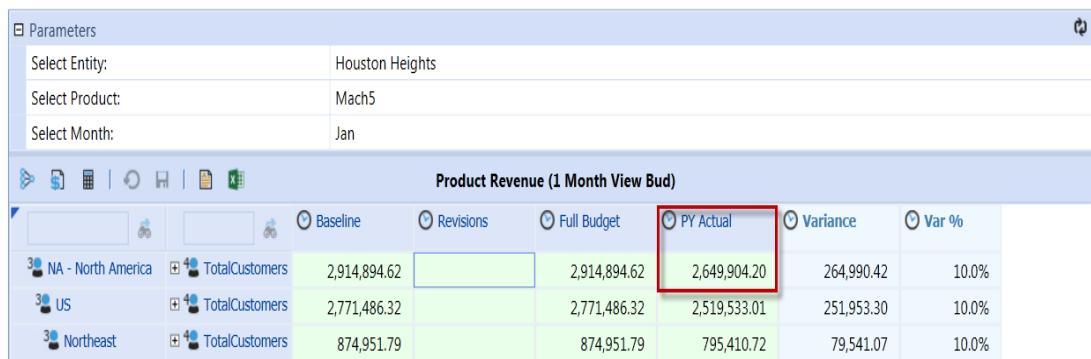
This determines if the Form Input Type is enabled for the Scenario Type. If not, the Form Input Type is disabled and cannot update cells from a Cube View in the web, Cube View, or XFSetCell function from Excel.

If Form Input Type is enabled, the Analytic Engine checks the full Workflow Status for the active Form Input Type. If the Workflow is locked or the Parent Workflow is certified, cells are not updated. If the Workflow indicates updates can occur, the Process Cube task of the Workflow and all ancestor Parent Workflows are impacted.

## Form Allocations

### Advanced Distribution

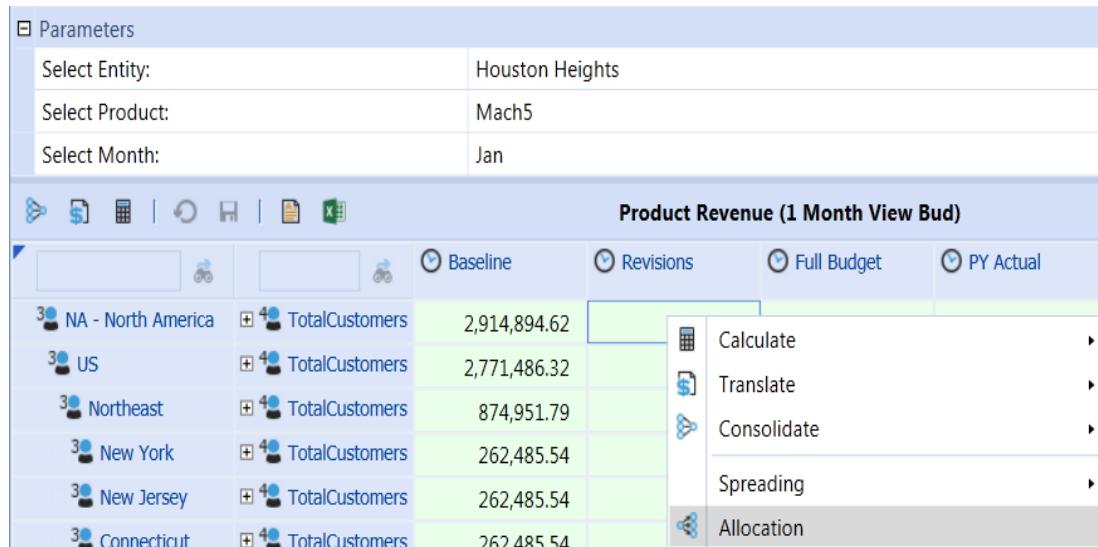
In the example below, an advanced distribution is used on a Product Revenue Form. This allocation will take the previous year's actual data, increase it by 20% and populate the current year's revenue budget revisions for all regions and customers.



			Baseline	Revisions	Full Budget	PY Actual	Variance	Var %
3 NA - North America	+ TotalCustomers	2,914,894.62		2,914,894.62		2,649,904.20	264,990.42	10.0%
3 US	+ TotalCustomers	2,771,486.32		2,771,486.32		2,519,533.01	251,953.30	10.0%
3 Northeast	+ TotalCustomers	874,951.79		874,951.79		795,410.72	79,541.07	10.0%

The allocation data is being written to a form which will then populate the Revisions column. The sum of the Baseline and Revisions will then create the new Full Budget for each Region and Customer.

Right-click the first data cell in the Revisions column and select Allocation. This helps create the Source and Destination POV.



			Baseline	Revisions	Full Budget	PY Actual
3 NA - North America	+ TotalCustomers	2,914,894.62				
3 US	+ TotalCustomers	2,771,486.32				
3 Northeast	+ TotalCustomers	874,951.79				
3 New York	+ TotalCustomers	262,485.54				
3 New Jersey	+ TotalCustomers	262,485.54				
3 Connecticut	+ TotalCustomers	262,485.54				

## Collecting Data

---

By default, the Allocation dialog will open to the last Allocation processed. Select the Allocation Type desired (e.g., Advanced).

The screenshot shows the 'Allocate' dialog box with the following configuration steps:

- General:** Allocation Type is set to Advanced.
- Source:**
  - 1 Source POV: Cb#Houston:E#[Houston Heights];C#USD:S#BudgetV1:T#2011M
  - 2 Source Amount or Calculation Script: 2649904.1998\*.2
- Destination:**
  - 3 Destination POV
  - 4 Dimension Type: UD3
  - 5 Member Filter: U3#NA.Base
  - 6 Dimension Type 2: UD4
  - 7 Member Filter 2: U4#Root.Base
  - 8 Weight Calculation Script: O#Import
  - 9 Destination Calculation Script: |SourceAmount| \* (|Weight| / |TotalWeight|)
  - 10 Translate Destination If Different Currency: True
  - 11 Save Zeros As No Data: True
- Offset:**
  - 12 Source Transfer POV
  - 13 Source Transfer Offset POV
  - 14 Destination Offset POV
- Generate Allocation Data** (Step 15)

Buttons at the bottom right include 'Close' and a large blue 'OK' button.

### 1. Source POV

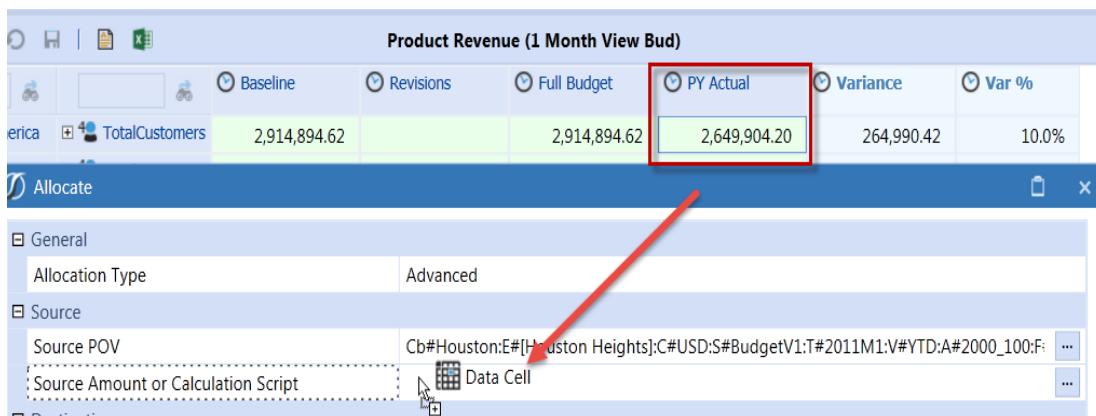
The Source POV defaults to the last cell selected for Allocation. Every Dimension is represented in the POV. In this example, it defaults to the data cell under Revisions because that is where the allocation option was selected.

Cb#Houston:E#[HoustonHeights]:C#USD:S#BudgetV1:T#2011M1:V#YTD:A#2000\_100:F#None:O#Forms:I#None:U1#None:U2#Mach5:U3#NA:U4#TotalCustomers:U5#None:U6#None:U7#None:U8#None

Users can also select a data cell from the grid and drag and drop the cell's POV into this field. The Source POV is the default Source Amount for the allocation.

### 2. Source Amount or Calculation Script

To override the Source POV value, enter a source amount or a calculation script. In this example, the PY Actual value is used for the Source Amount. Click on the cell in the grid and drag and drop the value into the Source Amount property. To increase this amount by 20%, the value is multiplied by .2.



### 3. Destination POV

This is where the allocation is applied. In this example, the Destination POV is blank because it is using the same Members from the Source POV. Users can also drag and drop a data cell's POV.

### 4. Dimension Type/Member Filters

These properties override the Destination POV and allow allocations to occur to several Members at a time. In this example, the UD3 (Regions) and UD4 (Customers) Dimensions are specified and therefore will override the UD3 and UD4 Members in the Destination POV.

5. **Weight Calculation Script**

The Weight Calculation Script determines how the allocation is weighted. Any Members not specified in the script derive from the Destination POV. In this example, the weight for each Member is determined by the imported value in the Baseline column.

6. **Destination Calculation Script**

The default Destination Calculation Script is  $|SourceAmount|^*$   $(|Weight|/|TotalWeight|)$ . Additional calculations may be added to this field to customize how the weight calculation is performed. This example uses the default calculation.

7. **Offset**

The offset properties are optional and not used in this example.

8. **Generate Allocation Data**

After the allocation dialog is complete, click this button to see the allocation data before applying it.

The allocation results dialog provides information on all the allocation destinations, weight information, and displays all the data rows that will be updated upon selecting Save Allocation Data. Check the Show All Dimensions box to see every Dimension intersection for each data row. After the allocation data is saved, the form data will update and store the data to the Cube.

## Collecting Data

---

**Allocation Results**

Source POV  
Cb#Houston:E#[Houston Heights]:P#?:C#USD:S#BudgetV1:T#2011M1:V#YTD:A#2000\_100:F#None:O#Forms:I#None:U1#None:U2#Mach5:U3#NA:U4#TotalCustomers:U5#None:U6#None:U7#None:U8#None

Source POV Amount 0.00  
|SourceAmount| 529,980.84  
|NumDestinations| 483  
|TotalWeight| 2,914,894.62

Show All Dimensions

Amount	Is NoData	Weight	Type	UD3	UD4	
4,980.47	<input type="checkbox"/>	27,392.60	Destination	Michigan	Active Hub	
6,445.32	<input type="checkbox"/>	35,449.24	Destination	Michigan	Double Eagle	
161.13	<input type="checkbox"/>	886.23	Destination	Michigan	East Sports	
336.91	<input type="checkbox"/>	1,853.03	Destination	Michigan	EZ Sporting Goods	
146.48	<input type="checkbox"/>	805.66	Destination	Michigan	Golf Hub	
981.45	<input type="checkbox"/>	5,397.95	Destination	Michigan	Golf Pro	

**Save Allocation Data** **Cancel**

Results:

**Product Revenue (1 Month View Bud)**

		<input type="radio"/> Baseline	<input type="radio"/> Revisions	<input type="radio"/> Full Budget	<input type="radio"/> PY Actual	<input type="radio"/> Variance	<input type="radio"/> Var %
3 <sup>o</sup> Midwest	+ 4 <sup>o</sup> TotalCustomers	397,676.06	72,304.74	469,980.80	361,523.69	108,457.11	30.0%
3 <sup>o</sup> Michigan	- 4 <sup>o</sup> TotalCustomers	99,419.01	18,076.18	117,495.20		117,495.20	
	4 <sup>o</sup> Active Hub	27,392.60	4,980.47	32,373.07		32,373.07	
	4 <sup>o</sup> Double Eagle	35,449.24	6,445.32	41,894.56		41,894.56	
	4 <sup>o</sup> East Sports	886.23	161.13	1,047.36		1,047.36	
	4 <sup>o</sup> EZ Sporting Goods	1,853.03	336.91	2,189.94		2,189.94	
	4 <sup>o</sup> Golf Hub	805.66	146.48	952.15		952.15	
	4 <sup>o</sup> Golf Pro	5,397.95	981.45	6,379.40		6,379.40	

**NOTE:** The Var % column updated itself to 30% from 10% because of the additional 20% added to the allocation. The Full Budget column also updated itself with the new total from the Baseline and Revisions columns.

## Applying Literal Value Parameters to Form Templates

A Delimited List Parameter containing several Cube View or Spreadsheet names can be applied to a Dashboard to use the Dashboard in multiple Form Templates via the Name Value Pairs Form Template Property. Instead of creating multiple Dashboards to assign to multiple Form Templates, users can define the Parameter name thus defining which Cube View or spreadsheet the specific form should use. This approach helps in achieving a common toolbar and look for all data entry forms.

The example below uses Cube Views, however, if the forms are driven from Spreadsheets, a Spreadsheet Dashboard Component can also be used.

1. Design the Cube Views necessary for data entry. After the Cube Views are complete, create a Dashboard Maintenance Unit.
2. Within the Dashboard Maintenance Unit, create a Delimited List Dashboard Parameter specifying all Cube View names in both the Value Items Property.

General (Parameter)	
Name	BudRev
Description	Budget Revenue Input Forms NA Course Mgmt
User Prompt	
Maintenance Unit	BudgetRevenue
Sort Order	0
Data Source	
Parameter Type	Delimited List
Default Value	
Display Items (comma delimited)	Design NA Revenue (1 Month View),Event Coordination NA Revenue (1 Month View),Services NA Revenue (1 Month View)
Value Items (comma delimited)	Design NA Revenue (1 Month View),Event Coordination NA Revenue (1 Month View),Services NA Revenue (1 Month View)

3. Create a Cube View Dashboard Component and enter the Parameter name in Cube View Property enclosed in Pipes and Exclamation Marks.

## Collecting Data

---

Component Properties	
General (Component)	
Name	cv_BudRevForms
Description	
Maintenance Unit	BudgetRevenue
Component Type	Cube View
Cube View	
Bind Parameter	!BudRev!
Show Header	True
Show Toggle Size Button	True

4. Create a Supplied Parameter Dashboard Component to pass the Parameter value from the Dashboard to the Form Template. Specify the Parameter Name in the Bound Parameter property.

Component Properties	
General (Component)	
Name	supp_BudRev
Description	
Maintenance Unit	BudgetRevenue
Component Type	Supplied Parameter
Action	
Bound Parameter	BudRev

5. Create a Dashboard with a Uniform Layout Type and assign the Cube View and Supplied Parameter Components to it.
6. Create a Form Template and set the Form Type to Dashboard and assign the desired Dashboard.
7. Define which Cube View this specific Form should use in the Name Value Pairs property by hardcoding a specific Cube View name from the Delimited List Parameter.

General	
Name	Budget Revenue
Description	
Form Type	Dashboard
Dashboard	BudgetRevenue
Excel File (optional)	
Refresh Spreadsheet When Opened	True
Workflow	
Form Requirement Level	Optional
Form Frequency	All Time Periods
Frequency Member Filter	
Time Member Filter For 'Complete Form'	
Literal Parameter Values	
Name Value Pairs (e.g., Param1=Value1, ...)	BudRev = Design NA Revenue (1 Month View)

When the Form Template is used in the Workflow, the specified Cube View will display for data entry.

## Loading Data through Excel Templates or CSV

Use Excel templates or CSV files to load data into stage using the Import method in Workflow, Forms data, Journals, Cell Detail, Data Attachments or to Custom Tables. This section describes the configuration required for this integration.

## Loading Stage Data

### Import Excel Template

When importing data into the Stage via Excel, a specific template must be used for OneStream to read and load the data. The user must first create Dimension Tokens (e.g. E#, A#, S#, etc.) to organize the data correctly. The Dimension Token specifies the specific type of data in any given column. For example, if the column header is E#, OneStream will read every row in that column as an Entity name when loading into Stage. After the Dimension Tokens are specified, create a Named Range beginning with xfd. There can multiple xfd Named Ranges across multiple tabs within an Excel workbook. The Dimension Tokens must be the Named Range's first row.

The following Dimension Tokens are used within an Import Excel Template. Please note these tokens can be in any order on the Excel template.

Dimension Tokens	Meaning
A#	Account: each row below will list the accounts to be imported
AMT#	Amount: using the AMT.ZS# header will automatically apply zero suppression to this import.  Tip: Apply zero suppression to a Matrix-style Excel import template by using this same .ZS extension on the AMT column.
F#	Flow
IC#	Intercompany
E#	Entity
C#	Consolidation
S#	Scenario
T#	Time Period

Dimension Tokens	Meaning
V#	View
O#	Origin
UD1#-UD8#	Each row must have a value even if a User Defined Member is not used in the application. Create a Static Value of None for any UD Members where this applies. Ex. UD5#[None] UX# or UDX# can be used for all User Defined Dimensions.
LB#	Label: This is used for an Account description related to a line of data. It is imported just for reference purposes and not stored in the Cube.
SI#	Source ID: This is a key for data imported into Stage. This typically includes a reference to the Entity being loaded but depends on the implementation. It is a best practice to have only one Source ID per Named Range and these can be the same or different for every Named Range imported for one Excel workbook.
TV#	Text Value: this is used to store large amounts of textual data.
A1# through A20#	Attribute Dimensions: these 20 Dimensions can each store 100 characters of text.
AV1# through AV12#	Attribute Value Dimensions: these 12 Dimensions can store numeric data.

## Header Abbreviations

### Static Value

Use :[] to fix a specific Member to the entire column creating a Static Value for the specified Source Dimension. For example, F#[None] imports the None Flow Member for every Flow row within the Named Range. This syntax applies to all Dimension Tokens.

## Collecting Data

---

Data Sources allow text values to be loaded as a View Member from the same row as the numeric value. Specify #Annotation, #VarianceExplanation, #AuditComment, #Footnote, or #Assumption as the Static Text Value of the TextValue Source Dimension and a new row will be created for the comment row. For example, use TV#:[#Annotation] to add an additional Annotation row.

### Business Rule

Pass a Business Rule for any specified Source Dimension to set a specific value.

```
AMT#:[]:[BusinessRuleNameThatSetsAValue]
```

### Matrix Member

This repeats for each Member. For example, if there were twelve time periods in the named range the syntax would be as follows:

```
T#:[]:[]:[2012M3]
```

To use Current/Global Scenario and Time, use .C# and .G# which creates a Static Value for the Time and Scenario within the Named Range. T.C# and S.C# returns the current Workflow Time and Scenario. T.G# and S.G# returns the Global Time and Scenario.

## Import Data Extracted via Data Management

Any type of data (Import, Forms, or Journals) extracted to a CSV file through a Data Management Job can be imported into Stage via an Extensibility Business Rule. This simplifies the migration of data between applications.

### Example

```
Dim objXFResult As XFRResult = BRApi.Finance.Data.SetDataCellsUsingCsvFile(si, filePath, delimiter,  
    originFilter, targetOriginMember, loadZeros)
```

When using this BRApi make sure to specify the Origin Filter which determines the type of data desired from the file (Import, Forms or Adjustments), and the Target Origin Member which determines where the data will be stored upon loading the file.

## Loading Form Data

### Form Excel Template

When loading Form data via Excel, a specific template must be created to determine the form properties, the Dimensions to which the data is loaded, the data entry amount, and any data attachment information.

OneStream reads this template using a specific Named Range which is explained later in this section. Ensure the following information is included in the Named Range.

## Collecting Data

---

### Property Tokens

The first four rows of the Named Range in the Excel template must include the following token definitions:

#### Form Import Template

FormTemplateName:	Accounting Standard Adjustments	Form Template Name
Workflow Name:	Houston.Forms	Valid Form Workflow Name
Workflow Scenario:	WFSenario	Workflow Scenario Name
Workflow Time:	WFTime	Workflow Time Name

#### Form Template Name

Enter the Form Template name intended for the Form data load.

#### Workflow Name

Enter the Form Workflow name. For example, if the name of the Workflow Profile is Houston, and the Form input type is named Forms, enter Houston.Forms.

#### Workflow Scenario

Enter the current Workflow Scenario such as Actual, Budget, etc. To dynamically use the current Workflow Scenario, use the |WFSenario| Substitution Variable.

#### Workflow Time

Enter the current Workflow Time Period. To dynamically use the current Workflow Time, use the |WFTime| Substitution Variable.

#### Dimension Tokens

Next, create the Dimension Tokens necessary to load the form data to the correct Dimensions in OneStream. The Dimension tokens need to be the column header for each data row. The standard tokens used determine the Cube, Entity, Parent, Account, Flow, Intercompany, the User Defined Members, and an Amount. Refer to Loading Stage Data for the syntax. The form specific tokens are as follows:

##### HD#

Has Data

Enter Yes or No to specify whether the row has data.

##### AN#

Annotation

##### AS#

Assumption

##### AD#

Audit Comment

## Collecting Data

---

**FN#**

Footnote

**VE#**

Variance Explanation

## Header Abbreviations

### Static Value

Use :[] to fix a specific Member to the entire column creating a Static Value for the specified Source Dimension. For example, F#[None] imports the None Flow Member for every Flow row within the Named Range. This syntax applies to all Dimension Tokens.

### Using Substitution Variables

If a Substitution Variable is used to define the Workflow Scenario or Workflow Time Tokens, link the Scenario and Time Dimension Tokens using the Ampersand (&) Excel Function and referencing the Excel cell.

Example:

	A	B	C	D	E	F
1	<b>Form Import Template</b>					
2						
3	FormTemplateName:	Accounting Standard Adjustments	Form Template Name			
4	Workflow Name:	Houston.Forms	Valid Journal Workflow Name			
5	Workflow Scenario:	WFScenario	Workflow Scenario Name			
6	Workflow Time:	WFTime	Workflow Time Name			
7	Dimension Tokens:	CB#	E#	P#	C# 2	S#: [WFScenario ]
8	Data Rows:	Houston	Houston Heights		Local	
9		Houston	Houston Heights		Local	
10		Houston	Houston Heights		Local	
11						

1. The Workflow Scenario Token, located in cell B5, is using a Substitution Variable to dynamically reference the user's current Scenario.
2. The Scenario Dimension Token needs to reference that Substitution Variable to ensure the correct Scenario is used and the template functions properly.
3. The syntax to reference the cell B5 is ="S#:[" & B5 & "]"  
This references the correct variable and displays it in the proper cell.

## Collecting Data

---

After the Dimension Tokens are configured, enter the data in the corresponding column. The Dimensions can be in any order.

Form Import Template								
FormTemplateName:	Accounting Standard Adjustments	Form Template Name						
Workflow Name:	Houston.Forms	Valid Journal Workflow Name						
Workflow Scenario:	WFSenario	Workflow Scenario Name						
Workflow Time:	WFTime	Workflow Time Name						
Dimension Tokens:	CB#	E#	P#	C#	S#[ WFSenario ]	T#[ WFTime ]	V#[YTD]	A#
Data Rows:	Houston	Houston Heights	Local					53100
	Houston	Houston Heights	Local					53110
	Houston	Houston Heights	Local					53120

F#	O#:Forms	IC#	UD1#	UD2#	UD3#	UD4#	UD5#[IFRS Adj]	UD6#	UD7#	UD8#	AMT#
None		None	None	None	None	None		None	None	None	9,000.00
None		None	None	None	None	None		None	None	None	9,000.00
None		None	None	None	None	None		None	None	None	12,000.00

HD#[Yes]	AN#	AS#	AD#	FN#	VE#

The final step is to create a Named Range beginning with XFF making sure to include the definition of each property, the Dimension tokens, and the data rows. The Named Range must begin with XFF for OneStream to read and load the form data correctly. Multiple XFF Named Ranges can be used across multiple tabs.

1	Form Import Template								
3	FormTemplateName:	Accounting Standard Adjustments	Form Template Name						
4	Workflow Name:	Houston.Forms	Valid Journal Workflow Name						
5	Workflow Scenario:	WFSenario	Workflow Scenario Name						
6	Workflow Time:	WFTime	Workflow Time Name						
7	Dimension Tokens:	CB#	E#	P#	C#	S#[ WFSenario ]	T#[ WFTime ]	V#[YTD]	A#
8	Data Rows:	Houston	Houston Heights	Local					53100
9		Houston	Houston Heights	Local					53110
10		Houston	Houston Heights	Local					53120
11									

## Collecting Data

---

### Form Matrix Excel Template

A form matrix template is used to load multiple amount columns to multiple time periods at once. In this template, the Time Dimension Token is combined with the amount to identify which amount should load to which period.

This template uses the same property tokens as a regular Excel Form template shown above.

### Form Matrix Import Template

FormTemplateName:	Accounting Standard Adjustments	Form Template Name
Workflow Name:	Houston.Forms	Valid Journal Workflow Name
Workflow Scenario:	WFSenario	Workflow Scenario Name
Workflow Time:	WFTime	Workflow Time Name

In the Matrix Form template, Amount and Time must be specified in the same column. A third Dimension can be specified (e.g., Scenario) if desired. The example below is indicating the Amount Column using AMT# and then specifying to which Time Members the Amount detail belongs.

AMT#:T#2011M1	AMT#:T#2011M2	AMT#:T#2011M3
1,000.00	2,000.00	3,000.00
2,000.00	4,000.00	6,000.00
3,000.00	6,000.00	9,000.00

### Form CSV Template

To set up a CSV template for a Form, the Header and Detail values must be specified.

1	A	B	C	D	E
2	!RowType (H=Header)	FormTemplateName	WFProfileName	WFSenarioName	WFTimeName
3	!RowType (D=Detail)	FormTemplateName	Cube	Entity	Parent
	H	Accounting Standard Adjustments	Houston.Forms	Actual	2011M4
	D	Accounting Standard Adjustments	Houston	Houston Heights	
	D	Accounting Standard Adjustments	Houston	Houston Heights	
4	D	Accounting Standard Adjustments	Houston	Houston Heights	
	D	Accounting Standard Adjustments	Houston	South Houston	
	D	Accounting Standard Adjustments	Houston	South Houston	
	D	Accounting Standard Adjustments	Houston	South Houston	

1. Column A Specifies Row Type

In the first two rows of Column A, create two Row Type Parameters specifying the Header and the Detail. In the example above, !RowType (H=Header) and !RowType (D=Detail) are used to tag the corresponding rows with H or D identifying where the Header and Detail information is located in the CSV file.

2. Row One Specifies the Headers

After the Header Parameter is configured, enter the form column headers. The required Form Headers are FormTemplateName, WFProfileName, WFSenarioName, and WFTimeName. For more details on these, refer to Form Excel Template.

3. Row Two Specifies the Details

After the Detail Parameter is configured, enter the form detail headers. The required Form Detail Headers are FormTemplateName, all 18 standard Dimensions, Amount, HasData, Annotation, Assumptions, AuditComment, Footnote, and VarianceExplanation. For more details on these, refer to Form Excel Template.

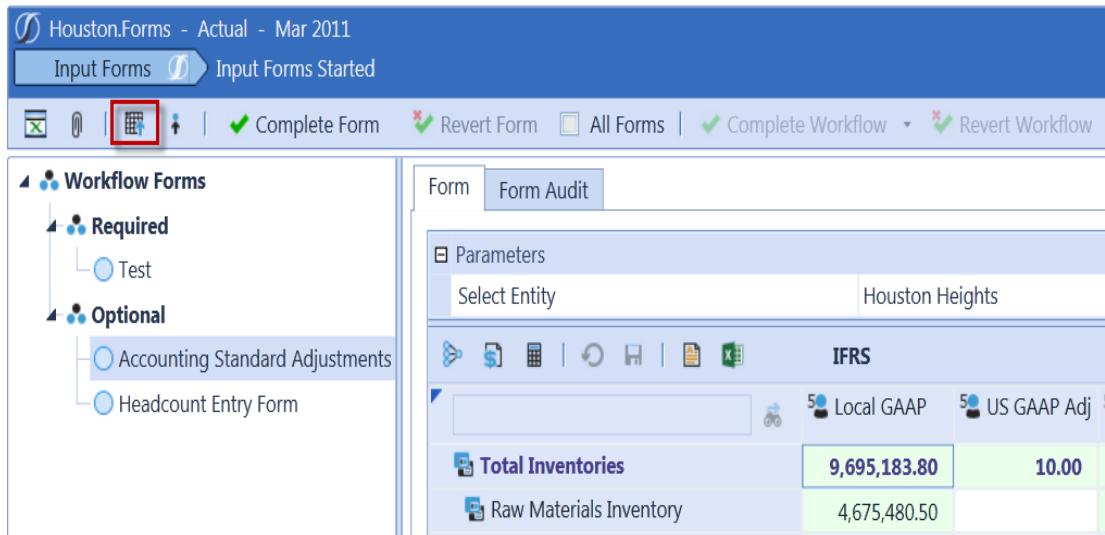
4. Header and Detail Tags

The Form data is driven by how each row in column A is tagged. Any rows tagged with H load as the Headers and any row tagged with D load as the details.

## Loading Form Details via Workflow

After the template is configured, users can load it directly into OneStream during the Workflow

process. While the Form Input type is selected, click the  icon in the Form toolbar. This allows the user to select the desired Excel or CSV template and load it into OneStream. After the file is loaded, the data will appear in the Form grid and it auto-saves upon importing to the Cube.



## Loading Form Details via Business Rules

Users can load Form details from an Excel Template or CSV file by configuring an Extensibility Business Rule and using the ImportAndProcessForms BRApi. Within the BRApi, define the Session Info, file path, and Form actions.

### Load Example

```
'BRApi.Forms.Data.ImportAndProcessForms(si, filePath, save, complete, throwError)
```

## Loading Journal Data

### Journal Excel Template

When loading journal entries via Excel a specific template must be created to determine the journal properties, the Dimensions to which the entries are being made, and the debit and credit amounts.

OneStream reads this template using a specific Named Range which is explained later in this section. Ensure the following information is included in the Named Range.

#### Property Tokens

The first eleven rows of the Named Range in the Excel template must include the following token definitions:

## Collecting Data

---

Journal Import Template	
Template Name:	
Name:	PeriodJournal
Description:	Test Journal Load
Journal Type:	Standard
Balance Type:	Balanced
Is Single Entity:	FALSE
Entity Filter:	E#YourMemberFilter, Blank
Consolidation Member:	USD
Workflow Name:	Houston.Journals
Workflow Scenario:	WFSzenario
Workflow Time:	WFTime
	Workflow Scenario Name

**Template Name**

Enter a template name if applicable. If this is a free form journal, leave this blank.

**Name**

Enter the name of the journal.

**Description**

If desired, enter a description for the journal.

**Journal Type\***

Standard or Auto-reversing. Allocation is not supported for Excel or CSV import.

**Balance Type\***

Balanced, Balanced by Entity, or Unbalanced

**Is Single Entity\***

True or False

**Entity Filter\***

Use a Member Filter to specify the Entities used with this journal.

**Consolidation Member**

Enter the specific currency or Local Member of Consolidation.

**Workflow Name**

Enter the Journal Workflow name. For example, if the name of the Workflow Profile is Houston, and the Adj input type is named Journals, enter Houston.Journals.

**Workflow Scenario**

Enter the Workflow Scenario or make the template dynamic by entering the |WFSzenario| Substitution Variable.

## Collecting Data

---

### Workflow Time

Enter the Workflow Time or make the template dynamic by entering the |WFTime| Substitution Variable. Workflow Time support two fields. The available cell immediately to the right is option representing the CubeTimeName. This field can be used when the Scenario's Workflow Tracking Frequency is Yearly, and the Input Frequency is Monthly. For example, the Workflow Time would be |WFTime| or 2019 and the CubeTimeName would be the period to post, 2019M7.

Workflow Scenario:	WFScenario	Workflow Scenario Name
Workflow Time:	WFTime	

When Tracking Frequency is Yearly and Input Frequency is Monthly:

Workflow Scenario:	WFScenario	Workflow Scenario Name
Workflow Time:	2019M7	

Workflow Scenario:	WFScenario	Workflow Scenario Name
Workflow Time:	WFTime	2019M7

Workflow Scenario:	WFScenario	Workflow Scenario Name
Workflow Time:	WFTime	WFYear M7

\*See Journal Templates "Data Collection" on page 511 for details on these Journal properties.

### Dimension Tokens

Next, create the Dimension Tokens necessary to load the journal to the correct Dimensions in OneStream. The Dimension tokens need to be the column header for each data row. The standard tokens used determine the Cube, Entity, Parent, Account, Flow, Intercompany, the User Defined Members, and a Label if needed. Refer to Loading Stage Data for the syntax. The journal specific tokens are as follows:

#### AMTDR#

This indicates the debited amount.

#### AMTCR#

This indicates the credited amount.

# Header Abbreviations

### Static Value

Use :[] to fix a specific Member to the entire column creating a Static Value for the specified Source Dimension. For example, F#[None] imports the None Flow Member for every Flow row within the Named Range. This syntax applies to all Dimension Tokens.

After the Dimension Tokens are setup, enter the data in the corresponding column.

Template Example:

Journal Import Template																	
Template Name:	Journal Template Name to Associate with Journal																
Name:	TestJV																
Description:	Test Journal Load																
Journal Type:	Standard																
Balance Type:	Balanced, Balanced By Entity, Unbalanced																
Is Single Entity:	Balanced																
Entity Filter:	False																
Consolidation Member:	Entity																
Workflow Name:	Houston.Journals																
Workflow Scenario:	WFSenario																
Workflow Time:	WFTime																
Dimension Tokens:	C#	E#	P#	A#	F#	I#	UD1#	UD2#	UD3#	UD4#	UD5#	UD6#	UD7#	UD8#	AMTDR#	AMTCR#	LB#
Data Rows:	Houston	Houston Heights	\$6000	None	9,000.00	Federal Taxes											
	Houston	Houston Heights	\$6100	None	9,000.00	State Taxes											
	Houston	Houston Heights	21200	None	18,000.00	Accrued Taxes											
															18,000.00	18,000.00	

The final step is to create a XFJ Named Range making sure to include the definition of each property, the Dimension tokens, and the data rows. The Named Range must begin with XFJ for OneStream to read and load it correctly. Multiple XFJ Named Ranges can be used within the template over multiple tabs.

**NOTE:** Loading of Journal Templates or previously exported Journal data only requires a Parent (P#) column value to be populated if the target Consolidation dimension member being updated is OwnerPreAdj or OwnerPostAdj. Otherwise, this entry can be left blank.

Named Range Example:

## Collecting Data

---

Journal Import Template																	
Template Name:	TestIV																
Name:	TestIV																
Description:	Test Journal Load																
Journal Type:	Standard																
Balance Type:	Balanced																
Is Single Entity:	FALSE																
Entity Filter:	EMHouston.Base																
Consolidation Member:	USD																
Workflow Name:	Houston.Journals																
Workflow Scenario:	[WFSenario]																
Workflow Time:	[WFTime]																
Dimension Tokens:	CB#	E#	P#	A#	F#	IC#	UD1#	UD2#	UD3#	UD4#	UD5#	UD6#	UD7#	UD8#	AMTDR#	AMTCR#	LB#
Data Rows:	Houston	Houston Heights	\$6000	None	9,000.00	Federal Taxes											
	Houston	Houston Heights	\$6100	None	9,000.00	State Taxes											
	Houston	Houston Heights	21200	None	18,000.00	Accrued Taxes											
															18,000.00	18,000.00	

**NOTE:** The Named Range only covers each property definition.

## Journal CSV Template

To set up a CSV template for a Journal, the Header and Detail values must be specified.

A	B	C	D	E	F
1	!RowType (H=Header) JournalName	OriginatingTemplateName	JournalDescription	JournalType	JournalBalanceType
2	!RowType (D=Detail) JournalName	CubeName	EntityName	ParentName	AccountName
H	Test Tax Accruals_Houston.Journals_Actual_2011M3	Tax Accruals	Test Journal Load	Standard	Balanced
D	Test Tax Accruals_Houston.Journals_Actual_2011M3	Houston	Houston Heights		56000
D	Test Tax Accruals_Houston.Journals_Actual_2011M3	Houston	Houston Heights		56100
D	Test Tax Accruals_Houston.Journals_Actual_2011M3	Houston	Houston Heights		21200
H	Test Tax Accruals_Houston.Journals_Actual_2011M3_1	Tax Accruals		Standard	Balanced
D	Test Tax Accruals_Houston.Journals_Actual_2011M3_1	Houston	Houston Heights		56000
D	Test Tax Accruals_Houston.Journals_Actual_2011M3_1	Houston	Houston Heights		56100
D	Test Tax Accruals_Houston.Journals_Actual_2011M3_1	Houston	Houston Heights		21200
D	Test Tax Accruals_Houston.Journals_Actual_2011M3_1	Houston	South Houston		56000
D	Test Tax Accruals_Houston.Journals_Actual_2011M3_1	Houston	South Houston		56100
D	Test Tax Accruals_Houston.Journals_Actual_2011M3_1	Houston	South Houston		21200

### 1. Column A Specifies Row Type

In the first two rows of Column A, create two Row Type Parameters specifying the Header and the Detail. In the example above, !RowType (H=Header) and !RowType (D=Detail) are used to tag the corresponding rows with H or D identifying where the Header and Detail information is located in the CSV file.

### 2. Row One Specifies the Headers

After the Header Parameter is configured, enter the Journal Column Headers. The required Journal Headers are JournalName, OriginatingTemplateName, JournalDescription, JournalType, JournalBalanceType, IsSingleEntity, EntityMemberFilter, ConsName, WFProfileName, WFScearioName, WFTimeName and CubeTimeName. For more details on these, refer to the Journal Excel Template.

### 3. Row Two Specifies the Details

After the Detail Parameter is configured, enter the Journal Detail Headers. The required Journal Detail Headers are JournalName, CubeName, EntityName, ParentName, AccountName, FlowName, ICName, all UDNames, DebitAmount, CreditAmount, and LineDescription. For more details on these, refer to Journal Excel Template.

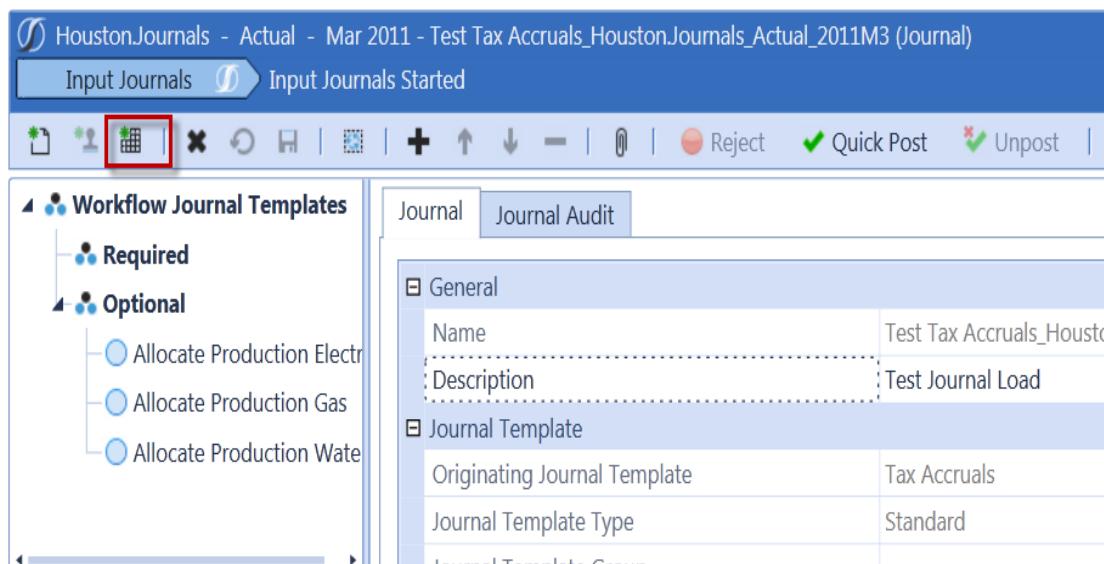
### 4. Header and Detail Tags

The Journal data is driven by how each row in column A is tagged. Any rows tagged with H load as the Headers and any row tagged with D load as the details.

## Loading Journal Details via Workflow

After the template is configured, users can load it directly into OneStream during the Workflow

process. While the Journal Input type is selected, click the  icon in the Journal toolbar. This lets you select the desired Excel or CSV template and load it into OneStream. After the file is loaded, the journal line items will appear in the journal and the user can save it to the Cube.



# Extracting/Loading Journal Details via BRApi

Users can export journal details into a CSV or XSLX file and load journals from a CSV or XSLX file by configuring an Extensibility Business Rule. This allows journal details to be extracted from one application and loaded into another. Users can also extract the journal, make changes, and re-load.

### Extract Journals to CSV

Use BRApi.Journals.Data.ExportJournalsToCSV and define the session, filepath, Workflow Profile, Scenario, Time Filter, and Journal Status.

```
BRApi.Journals.Data.ExportJournalsToCsv(si, filePath, "Houston", "Actual", "T#|WFYear|.Base",  
"Posted")
```

### Extract Journals to XSLX

Export journals associated with defined parameters to an XSLX file using the ExportJournalsToXSLX BRApi. It is sometimes easier to read and modify journal data in an XSLX format as compared to CSV files that are generated by the ExportJournalsToCSV BRApi. Use the parameters listed in the following table.

Parameter	Description
si	Session information
serverFilePath	Name and location to save the file
wfProfileName	Name of the Workflow Profile
wfScenarioName	Name of the Scenario
wfTimeMemberFilter	Filters workflow profile times. For example, if the scenario's Workflow Tracking Frequency is set to Quarterly, and Input Frequency is set to Monthly, the filter searches for journals associated with the Workflow Tracking Frequency.  Time must be prefixed with T#. You can type a comma separated list of times such as T#2022H1, T#2022H2.
journalName	Filters journals that contain the specified text

Parameter	Description
journalStatus	Status of the journal. You can search by status or use All to return all statuses. You cannot type a comma separated list.

The following code example uses the ExportJournalsToXlsx BRApi:

```
1 Imports System
2 Imports System.Collections.Generic
3 Imports System.Data
4 Imports System.Data.Common
5 Imports System.Globalization
6 Imports System.IO
7 Imports System.Linq
8 Imports Microsoft.VisualBasic
9 Imports OneStream.Finance.Database
10 Imports OneStream.Finance.Engine
11 Imports OneStream.Shared.Common
12 Imports OneStream.Shared.Database
13 Imports OneStream.Shared.Engine
14 Imports OneStream.Shared.Wcf
15 Imports OneStream.Stage.Database
16 Imports OneStream.Stage.Engine
17
18 Namespace OneStream.BusinessRule.Extender.ExportJournalsToXSLX
19     Public Class MainClass
20         Public Function Main(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Object, ByVal args As ExtenderArgs) As Object
21             Try
22                 Dim configSettings As AppServerConfigSettings = AppServerConfig.GetSettings(si)
23                 Dim sourcefolder As String = BRApi.Utilities.GetFileShareFolder(si, FileShareFolderTypes.FileShareRoot, Nothing) & "\System\Contents\Export"
24                 Dim journalFilePath As String = sourcefolder & "\" & "Journal_Export.xlsx"
25                 Dim filePath As String = BRApi.Journals.Data.ExportJournalsToXlsx(si, journalFilePath, "Houston", "Actual", "T#2023HS", "", "All")
26
27                 Return Nothing
28             Catch ex As Exception
29                 Throw ErrorHandler.LogError(si, New XFEException(si, ex))
30             End Try
31         End Function
32     End Class
33 End Namespace
```

### Load Example

Use BRApi.Journals.Data.ImportAndProcessJournals and define the session, filepath, and journal tasks to complete upon loading the journal details.

'BRApi.Journals.Data.ImportAndProcessJournals(si, filePath, save, submit, approve, post, unpostAndOverwrite, throwOnError)

## Loading Cell Detail

### Cell Detail Excel Template

When loading Cell Detail via Excel, a specific template must be created to determine the Cell Detail Dimension Tokens and each Cell Detail line data.

## Collecting Data

---

OneStream reads this template using a specific Named Range which is explained later in this section. Ensure the following information is included in the Named Range.

### Dimension Tokens

The first 19 rows of the Named Range in the Excel template must include the following token definitions:

CubeName	Houston
EntityName	Houston Heights
ParentName	
ConsName	USD
ScenarioName	ActualTest
TimeName	2011M3
AccountName	56100
FlowName	None
Origin	Forms
ICName	None
UD1Name	None
UD2Name	None
UD3Name	None
UD4Name	None
UD5Name	IFRS Adj
UD6Name	None
UD7Name	None
UD8Name	None
DetailViewType	Periodic

The Cube and each Dimension Member must be specified. All User Defined Members must be specified. If a specific User Defined Member is not used in the application, enter None.

Next, create the Dimension Tokens necessary to load each Cell Detail line. The specific tokens are as follows:

Dimension Tokens:	AMT#	LIT#	AW#	CL#	LB#
Data Rows:	700.00	Default		1.00	Description or Detail

#### AMT#

Amount

#### LIT#

Line Item Type

## Collecting Data

---

### **AW#**

Aggregation Weight

### **CL#**

Classification Type

### **LB#**

Label allowing users to add a description or additional detail.

After the Dimension Tokens are configured, enter the data in the corresponding column. The Dimensions can be in any order.

CubeName	Houston					
EntityName	Houston Heights					
ParentName						
ConstName	USD					
ScenarioName	ActualTest					
TimeName	2011M3					
AccountName	56100					
FlowName	None					
Origin	Forms					
ICName	None					
UD1Name	None					
UD2Name	None					
UD3Name	None					
UD4Name	None					
UD5Name	IFRS Adj					
UD6Name	None					
UD7Name	None					
UD8Name	None					
DetailViewType	Periodic	Amount	Line Item Type	Aggregation Weight	Classification	Description
Dimension Tokens:		AMT#	LIT#	AW#	CL#	LB#
Data Rows:		700.00	Default	1.00		Description or Detail

The final step is to create a Named Range beginning with XFC making sure to include the definition of Dimension token, and the data rows. The Named Range must begin with XFC for OneStream to read and load the Cell Detail correctly. Multiple XFC Named Ranges can be used across multiple tabs.

## Collecting Data

---

xfcDetailItems		X	✓	fx	Houston	
A	B	C	D	E	F	G
1	Cell Detail Import Template					
2						
3	CubeName	Houston				
4	EntityName	Houston Heights				
5	ParentName					
6	ConsName	USD				
7	ScenarioName	ActualTest				
8	TimeName	2011M3				
9	AccountName	56100				
10	FlowName	None				
11	Origin	Forms				
12	CName	None				
13	UD1Name	None				
14	UD2Name	None				
15	UD3Name	None				
16	UD4Name	None				
17	UD5Name	IFRS Adj				
18	UD6Name	None				
19	UD7Name	None				
20	UD8Name	None				
21	DetailViewType	Periodic	Amount	Line Item Type	Aggregation Weight	Classification
22	Dimension Tokens:		AMT#	LIT#	AW#	CL#
23	Data Rows:		700.00	Default	1.00	Description or Detail
24						
25						
26						

## Cell Detail CSV Template

To set up a CSV template for Cell Detail, the Header and Detail values must be specified.

1	A	B	C	D	E	F	G	H
2	!H (RowType Header)	CubeName	EntityName	ParentName	ConsName	ScenarioName	TimeName	AccountName
3	!D (RowType Detail)	Amount	LineItemType	AggregationWeight	Classification	Description		
	H	Houston	Houston Heights		USD	Actual	2011M3	56100
4	D	700	Default		1 Class1	Desc1		
	H	Houston	South Houston		USD	Actual	2011M3	56100
	D	2000	Default		1 Rebates Received on Supplies	Desc2		

### 1. Column A Specifies Row Type

In the first two rows of Column A, create two Row Type Parameters specifying the Header and the Detail. In the example above, !RowType (H=Header) and !RowType (D=Detail) are used to tag the corresponding rows with H or D identifying where the Header and Detail information is located in the CSV file.

### 2. Row One Specifies the Headers

After the Header Parameter is configured, enter the Cell Detail column headers. The required Cell Detail Headers specify the Cube and all 18 Dimension Members. For more details on these, refer to Cell Detail Excel Template.

### 3. Row Two Specifies the Details

After the Detail Parameter is configured, enter the Cell Detail detail headers. The required Detail Headers are Amount, LineItemType, AggregationWeight, Classification, and Description. For more details on these, refer to Cell Detail Excel Template.

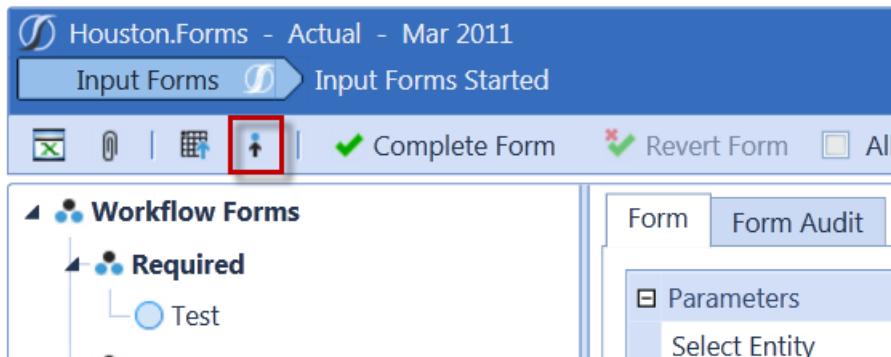
### 4. Header and Detail Tags

The Cell Detail's data is driven by how each row in column A is tagged. Any rows tagged with H load as the Headers and any row tagged with D load as the details.

## Loading Cell Detail via Workflow

After the template is configured, users can load it directly into OneStream during the Workflow

process. While the Form Input type is selected, click the  icon in the Form toolbar. This allows the user to select the desired Excel or CSV template and load it into OneStream. After the file is loaded, it has been successfully stored to the Cube.



## Extracting/Loading Cell Detail via Business Rules

Users can export Cell Details into a CSV file and load Cell Details from a CSV file by configuring an Extensibility Business Rule. This allows Cell Detail to be extracted from one application and loaded into another. Users can also extract the Cell Detail, make changes, and re-load.

### Extract

Use BRApi.Finance.Data.ExportCellDetailtoCSV and define the session, filepath, the Entity Dimension, the Entity Member Filter, the Scenario and the Time Member Filter.

```
BRApi.Finance.Data.ExportCellDetailToCsv(si, filePath, entityDimensionName,  
entityMemberFilter, scenarioName, timeMemberFilter)
```

### Load Example

Use BRApi.Finance.Data.ImportCellDetail and define the session and filepath.

```
BRApi.Finance.Data.ImportCellDetail(si, filePath, throwOnError)
```

## Exporting Data Attachments

### Exporting Data Attachments via Business Rule

You can export data attachments to perform variety of actions such as retrieve the file name, read the contents, store them in a File Explorer folder, or import them into another application. This process is configured in an Extensibility Business Rule using the following API.

Example:

```
Dim objDataAttachmentList As DataAttachmentList = BRApi.Finance.Data.  
GetDataAttachments(si, memberScript, includeFileBytes)
```

### Exporting Data Attachment Text

You can export the Data Attachment Text field to a CSV file using an Extensibility Business Rule. For example, use BRApi.Finance.Data.ExportCellTextToCSV and define the session, filepath, the Entity Dimension, the Entity Member Filter, the Scenario and the Time Member Filter.

```
BRApi.Finance.Data.ExportCellTextToCsv(si, cellTextFilePath,  
entityDimensionName, entityMemberFilter, scenarioName, timeMemberFilter)
```

## Loading Excel Templates to Custom Tables

OneStream MarketPlace Solutions typically have related SQL Server tables. Other custom solutions may also include adding custom SQL Server tables. This allows users to load data to these custom tables using an Excel template. The mechanism for loading these tables could be through the user interface of a MarketPlace Solution or through OneStream Extensibility Rules.

OneStream reads this template using a specific Named Range which is explained later in this section. Ensure the following information is included in the Named Range.

In the first three rows of the Named Range in Column A, specify the following:

### Database Location

Application or System specifies which database contains the custom tables.

### Table Name

Custom tables only; enter the Table name

### Load Method

The load method determines the action and any additional criteria for the action.

The syntax is:

Action:[Where Clause Criteria] (Where Clause Criteria is optional)

## Load Method Definitions

### Merge

If there are no criteria, Merge updates the data if it finds a matching key, otherwise it inserts it

### Merge Where Clause Criteria Example

First, this will clear the values for emp1 and then Merge

Merge:[EmployeeID = 'emp1']

### Merge Where Clause Criteria with Substitution Variable Example

Substitution Variables can be used in the Where Clause Criteria

Merge:[WFProfileName = '|WFProfile|']

### Replace

If there are no criteria, Replace clears everything first. By default, instead of merging, it clears the entire table. This will perform better for high volume because it does not try to match rows from the file to the table. An error will occur if it finds a match.

### Replace Where Clause Criteria Example

This does not try to locate, it only does inserts and appends.

Replace:[EmployeeID = 'emp1']

### Replace Where Clause Criteria with Substitution Variable Example

Replace:[WFProfileName = '|WFProfile|']

Next, define the Field Types and Field Names beginning in Column A Row 4 and spanning as many columns as necessary.

The column definition syntax is:

FieldType#[FieldName]:StaticValue(optional):DefaultValue(optional)

### Field Type

This relates to the column name in the table.

**xfGuid**

Unique identifier [SQL = uniqueidentifier]

**xfText**

Text defined column in the table [SQL = nvarchar, nchar, ntext]

**xfInt**

Short integer (4 byte integer) [SQL = int]

**xfBit**

0,1 (True, False) [SQL = bit]

**xfDec**

Decimal [SQL = Decimal (28,9)]

**xfDbl**

Floating point number (8 byte floating) [SQL = Float]

**xfDateTime**

Date [SQL = datetime]

**Field Name**

This is specific to the SQL table to be loaded.

**StaticValue**

Whatever is specified as the Static Value will override every row for that column regardless if it is blank or not.

**StaticValueExample**

This example will override all rows and enter 50,000 as the Static Value.

xfDec#[Salary]:50,000

**DefaultValue**

This only applies to blank rows.

**NOTE:** If something is specified in the Static Value, it will ignore whatever is in the DefaultValue.

**Default Value Example**

This example will enter a New Guid for all blank rows in the column.

xfGuid#[EmployeeID]::.NewGuid

**Substitution Variable Example**

Substitution Variables can be used in both StaticValue and DefaultValue.

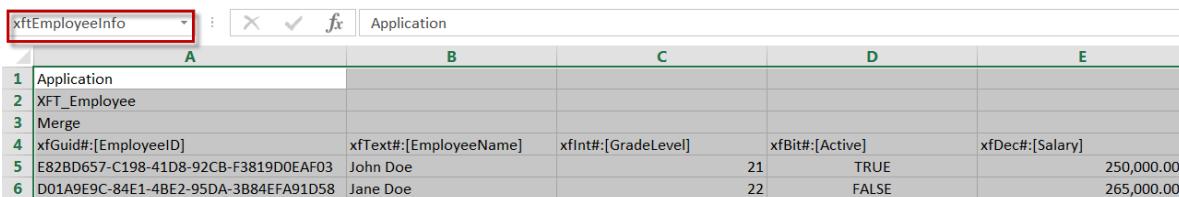
xfText#[EmployeeName]::|Username|

## Collecting Data

---

	A	B	C	D	E
1	Application				
2	XFT_Employee				
3	Merge				
4	xfGuid#[EmployeeID]	xfText#[EmployeeName]	xfInt#[GradeLevel]	xfBit#[Active]	xfDec#[Salary]
5	E82BD657-C198-41D8-92CB-F3819D0EAF03	John Doe		21	TRUE
6	D01A9E9C-84E1-4BE2-95DA-3B84EFA91D58	Jane Doe		22	FALSE
-					

Finally, create a Named Range beginning with XFT making sure to include the entire template.



	A	B	C	D	E
1	Application				
2	XFT_Employee				
3	Merge				
4	xfGuid#[EmployeeID]	xfText#[EmployeeName]	xfInt#[GradeLevel]	xfBit#[Active]	xfDec#[Salary]
5	E82BD657-C198-41D8-92CB-F3819D0EAF03	John Doe		21	TRUE
6	D01A9E9C-84E1-4BE2-95DA-3B84EFA91D58	Jane Doe		22	FALSE
-					

After the template is complete, it is ready to be loaded into the custom table. If this is being used in conjunction with a MarketPlace Solution, refer to the Solution for further instructions on how to load the template to the table. If this is being loaded via an Extensibility Business Rule, refer to the following example.

### Example

```
Dim fieldTokens As New List(Of String)
fieldTokens.Add("xfGuid#[EmployeeID]::.NewGuid")
'fieldTokens.Add("xfGuid#[EmployeeID]")
fieldTokens.Add("xfText#[EmployeeName]")
'fieldTokens.Add("xfText#[EmployeeName]::|Username|")
fieldTokens.Add("xfInt#[GradeLevel]")
fieldTokens.Add("xfBit#[Active]")
fieldTokens.Add("xfDec#[Salary]")
fieldTokens.Add("xfDb1#[VacationDays]")
fieldTokens.Add("xfDateTime#[HireDate]")
BRApi.Utilities.LoadCustomTableUsingDelimitedFile(si, SourceDataOriginTypes.FromFileShare, filePath,
Nothing, ",",
dbLocation, tableName, loadMethod, fieldTokens, True)
```

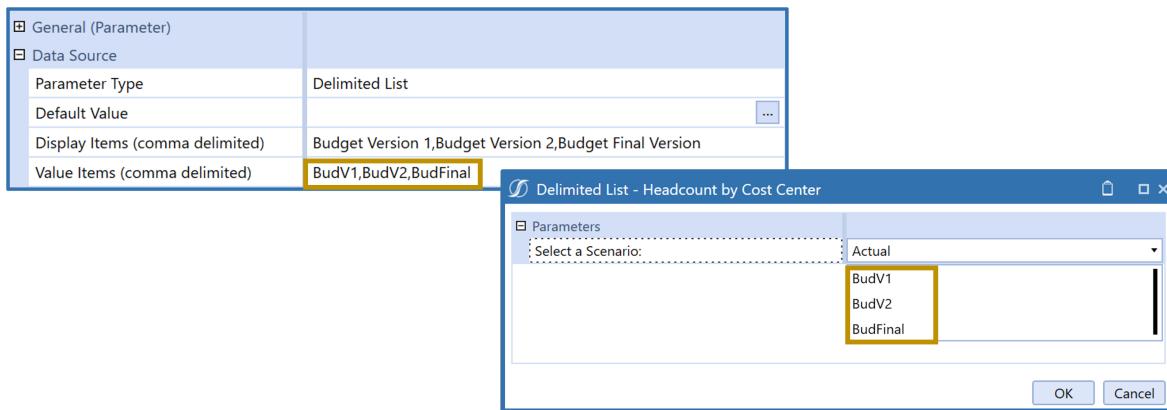
# Filter Runtime Data and Reduce Maintenance with Parameters

Parameters prompt users to filter or customize data at runtime, helping streamline objects such as forms, dashboards, and cube view reports to best suit different user needs. Administrators typically define parameters primarily in maintenance units, and then assign parameters to the appropriate object. See:

- [About Parameters](#)
- [Parameter Types](#)
- [Benefits](#)
- [Ways to Use Parameters](#)
- [Requirements](#)
- [Best Practices](#)
- [Create Parameters](#)

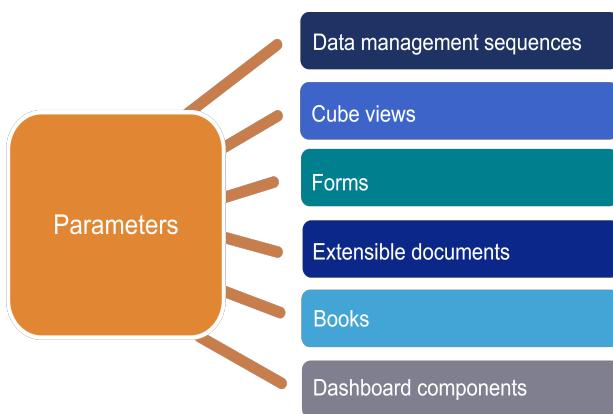
## About Parameters

Depending on the [type](#), parameters prompt you to specify data by entering text or selecting items from a drop-down list or dialog box. For example, a Delimited List parameter could prompt you to select from 1 of 3 budget versions to use in a Headcount by Cost Center report, as shown:



As shown below, you can use parameters with a variety of objects, such as:

- Data management sequences
- Cube views
- Forms
- Extensible documents
- Books
- Dashboard components



For example, use parameters as placeholders or with substitution variables and retrieve functions to enhance extensible documents with selections for time period, entity, data cell values, and more. See [Ways to Use Parameters](#).

## Parameter Types

Parameters are created in dashboard maintenance units. The following parameter types are available, each with specific uses.

- [Literal Value parameters](#)
- [Input Value parameters](#)
- [Delimited List parameters](#)
- [Bound List parameters](#)

- [Member List parameters](#)
- [Member Dialog parameters](#)

To reference an existing parameter, when formatting a cube view header or cell for example, enclose the parameter name in pipes and exclamation points. For example, `!ParameterName!`.

Each parameter type has two sections of properties, General and Data Source parameter properties. Properties identify a parameter, determine a parameter's behavior, and control how the parameter is used.

[General](#) properties are standard across all parameter types. [Data Source](#) properties vary by parameter type, but the first property for all parameter types is the Parameter Type property, where you select the type of parameter from a list.

## Literal Value Parameters

Literal value parameters are the only parameter type that does not prompt for any selections during creation.

Use literal value parameters to standardize formatting for cube view columns and rows and dashboard components. This helps reduce reporting maintenance. .

To create a literal value parameter, first [create a new parameter](#), then [specify the Literal Value parameter settings](#).

## Input Value Parameters

Use Input Value parameters to flexibly enter or change a value used in a cube view or other object. You can enter or modify a value to use in dashboard components and cube view rows and columns.

The Input Value parameter type helps you standardize and more easily maintain report and dashboard formats. For example, you can create an Input Value parameter that, when run in a cube view, prompts a for the name to display as **Report Run By** in a product sales report.



### Product Sales

Report Run By: Gemma Daly

	Budget V1	Budget V2	Final Budget
Total Products	534,686.10	534,686.10	534,686.10
GolfBalls	360,843.60	360,843.60	360,843.60
Tour	109,858.60	109,858.60	109,858.60
Premium	81,400.00	81,400.00	81,400.00
Range	45,570.00	45,570.00	45,570.00
Logo	83,995.00	83,995.00	83,995.00
LogoTour	40,020.00	40,020.00	40,020.00
Clubs	143,019.50	143,019.50	143,019.50
Woods	106,931.00	106,931.00	106,931.00
Drivers	50,400.00	50,400.00	50,400.00
FairwayWoods	56,531.00	56,531.00	56,531.00

To create an Input Value parameter, first "Create a Parameter" on page 198, then [specify the Input Value parameter settings](#).

## Delimited List Parameters

Use Delimited List parameters to create a distinct list of specified values, including an easily recognizable name. For example, you can create a list containing the months of the year to use as value items. Use the display items field for the delimited list parameter to list each month value separated by a comma, then associate each with a value item to create the list.

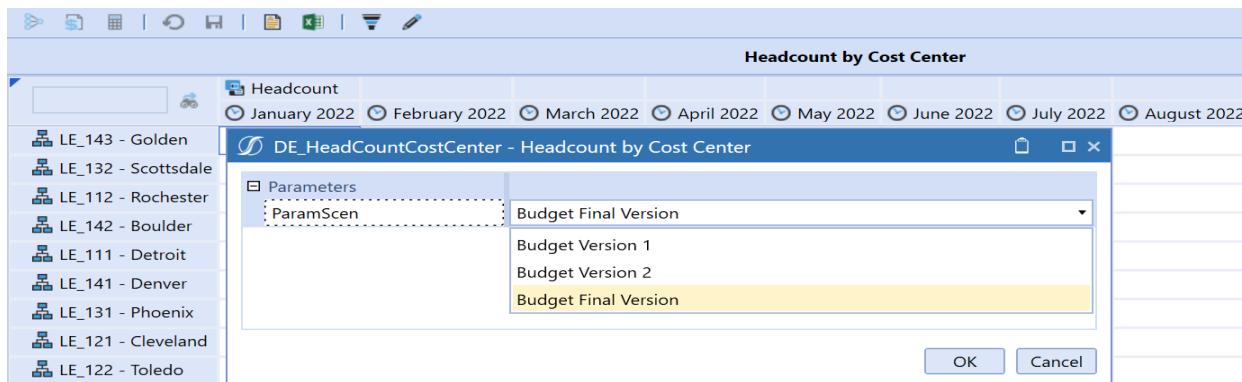
Delimited List parameters let you pick a value from a drop-down list instead of typing a value into a cell. Parameters can be assigned as a list source to a cube view row or column. These parameters are supported in Excel, web browsers, and cube view-driven reports.

You can also specify a parameter name that can be a cube view row or column. Edit cells using a drop-down list containing the parameter's list of items. A number is stored in the data cell as specified in the parameter's definition. If using a Delimited List parameter on a numeric cell, ensure each value in the parameter's name-value pairs is a number.

## Filter Runtime Data and Reduce Maintenance with Parameters

---

Select from a list of values, specified as display or value items, such as months or financial quarters. For example, a Delimited List parameter could prompt a user to select from one of three budget versions for a Headcount by Cost Center report.



These types of parameters also help you prompt for columns when sharing columns in a cube view, so you can specify column names at runtime. For example, you could specify columns for products or years in an income statement.

To create a Delimited List parameter, first "Create a Parameter" on page 198, then [specify the Delimited List parameter properties](#). See a [sample](#).

## Bound List Parameters

A bound list is group of members or other objects that were created using a predefined method query or by entering an SQL expression to retrieve members. Use Default Value to enter an input value.

For example, to list all entities in a dimension, use this method query with a command type of member:

```
{Dimension type}{Dimension name}{Member Filter}{where clause}  
{Entity}{CorpEntities}{E#Root.TreeDescendants}{}
```

The difference between the two query types is that a method query is an SQL query helper that you can use to quickly perform queries.

SQL queries are more customizable and often used in dashboard design to prompt for information such as the workflow profile name. See [Determine the Query Method](#).

To create a Bound List parameter, first "Create a Parameter" on page 198, then [specify the Bound List parameter settings](#).

## Member List Parameters

A Member List parameter is similar to a Delimited List parameter, but you must specify more properties. Also, a Member List parameter does not have the option to display a value item or display item.

Users select a member, such as an account or entity dimension member. For example, you could prompt users to select a base Scenario member when they run a cube view to display budget actuals.

Members that display in the member list are based on the member filters you specify for a dimension.

To create a Member List parameter, first "Create a Parameter" on page 198, then [specify the Member List parameter settings](#).

## Member Dialog Parameters

Similar to Member List parameter, the Member Dialog parameter lets you associate a member with the parameter through a dialog box with search capabilities. This is more appropriate for a dimension such as accounts or entities where you can use the hierarchy to select a base or parent member.

To create a Member Dialog parameter, first "Create a Parameter" on page 198, then [specify the Member Dialog parameter settings](#).

## Benefits

As configurable prompts, parameters enhance reporting and analysis flexibility because they let you dynamically filter the data to retrieve. You can also use some [types](#) of parameters with form templates to streamline data collection. Some parameters also store values that you can share across objects, helping make applications more dynamic and reduce some maintenance tasks. For example:

- Use parameters with substitution variables and retrieve functions to hold values so you can insert time periods, entities, text comments, and data cells in extensible documents.

- Use Literal Value parameters to define and apply standard formatting to cube view rows and columns and to dashboard components, reducing repetitive, error-prone tasks and ensuring a consistent look and feel in your analytics. You could create a Default Header parameter that captures the header format to use in reports as shown below and then assign the parameter to the appropriate cube views.

General (Parameter)	
Name	DefaultHeader
Description	Report header
User Prompt	
Maintenance Unit	GolfStream Selection Parameters
Sort Order	7
Data Source	
Parameter Type	Literal Value
Default Value	Bold = True, FontFamily = Calibri, TextColor = Gray

For information about some of the tasks you can perform with parameters, see [Ways to Use Parameters](#).

## Ways to Use Parameters

You can use parameters extensively across OneStream objects. This topic identifies some of the advantages of using parameters with some objects such as the following with:

- [Cube Views](#)
- [Data Adapters and Dashboard Components](#)
- [Extensible Documents](#)
- [Books](#)

## Cube Views

Assign parameters to a cube view to:

- Specify and apply standard formatting to cube views and dashboards for a consistent look and feel in your reports.

- Identify the details users can access when they navigate linked cube views and dashboards to access more data. See [Linked Cube Views](#).
- Let users refine the data to supply or retrieve in reports by prompting them to select entities, workflow profiles, views, scenarios, and more. To limit a query, refer to the parameter in the point of view, rows, or columns. This filters data in reports based on parameter selections.
- Specify the point of view to display more focused data.
- Nest parameters to use a series of parameter prompts to refine the data to use on dashboards and in reports. See [Nest Parameters](#).

## Data Adapters and Dashboard Components

Use parameters with data adapters to:

- Prompt users to select the data to retrieve and display on dashboards, so they can focus their analysis using custom intersections of some of the cube view data that populates the dashboard. For example, assign parameters to prompt for entity and year selections to a data adapter so users can visualize data by entity and year while accessing the rest of the cube view data.
- Enable dashboard actions such as custom calculations or launching another dashboard.
- Identify static values using Literal Value parameters, for dashboard content and design.

See [Parameter Components in Dashboards](#).

## Extensible Documents

Use parameters as placeholders and with substitution variables and retrieve functions to enhance extensible documents, such as Word and PowerPoint files, with dynamic selections for time period, entity, data cell value, text comments, and more. For example, insert the following parameters to prompt users to specify the years and quarter to use in a report:

- `!CurrentQtr!`
- `!ReportingYear!`
- `!ReportingYearPrior!`

You can use parameters in Microsoft text, Excel, Word, and PowerPoint files. See [Present Data With Extensible Documents](#).

## Books

Use books to combine a variety of reports and files in one object for custom reporting. You can automatically generate input parameters based on book content or manually specify the required parameters. You can also use parameters in loops- sequences of variable-based instructions, to further customize the data to display. For example, a book can loop to display all base entities in a hierarchy, providing the same cube view report for each entity. Or, you could assign a parameter called *ParamSalesRegions* to include all sales regions in book content.

If the source objects used in a book have assigned parameters, use Change parameters to override them instead of modifying the underlying cube view or report. For example, configure a Change parameter to use the workflow time in a book instead of prompting users to specify a Time dimension member.

**NOTE:** Before creating books and using Change parameters, ensure that you understand the source objects, such as cube views, that generate the content to use.

## Requirements

This topic describes who should create parameters and the syntax to use when you assign parameters to application objects.

## Definition and Assignment

Because they are more familiar with OneStream applications, objects and processes, a OneStream Administrator or an application designer should define parameters after identifying:

- The information that a parameter must gather from users at runtime.
- If a sequence of parameters is needed to gather the required data. Specify a staggered sort order to display parameters in the right order to collect increasingly granular levels of detail.

See [Best Practices](#).

## Syntax

Enclose parameter names in pipes and exclamation points when you assign parameters to objects, regardless of parameter type. For example, `!ParamView!` prompts users to select year to date or periodic data when they run a cube view for reporting.

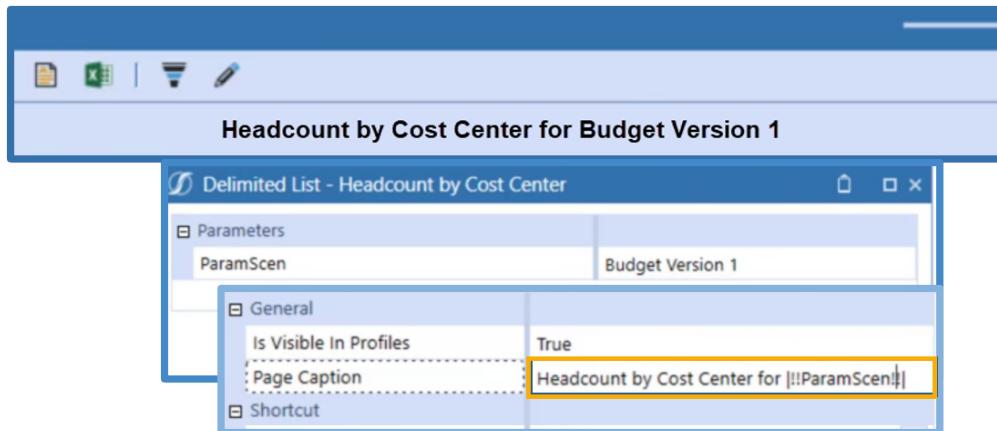
## Filter Runtime Data and Reduce Maintenance with Parameters

---

Use double exclamation points (for example, `!!!ParamEntity!!!`) when assigning Delimited List parameters to present users with the parameter's more intuitive and helpful display item value.

Follow these best practices when assigning Delimited List parameters:

- Assignment to cube view rows and columns - Use single exclamation points to display the member name. If you use double exclamation points, you may receive an error at runtime as the parameter tries to retrieve the Display Item value.
- Assignment to page captions - Use single exclamation points to display the Value Items. Use double exclamation points to show the Display Item. For example, instead of displaying BudV1 - the Value Item, specify `!!!Budget Version 1!!!` to display Budget Version 1 on reports as shown:



## Best Practices

This topic provides tips to help you design and maintain parameters. See:

- [Design](#)
- [Naming](#)

## Design

Since parameters can be used with different objects such as similar types of reports, reduce the associated maintenance tasks by designing parameters with:

- Reuse in mind, making parameters applicable to different objects. If you modify a parameter to address changing reporting needs, keep in mind that your changes apply to all objects to which the parameter is assigned.
- A staggered sort order where at least 20 sort numbers are unassigned to support new parameters possibly needed for future report and dashboard redesign.

For example, two parameters with respective sort orders of five and six display consecutively. However, to display a new parameter in between them, you must change the parameter with sort order of 6 to 7 and then assign sort order of 6 to the new parameter. If you have hundreds of parameters, changing the existing sort order to support new parameters is tedious and error prone. In this example, assign a sort order of 5 to the first parameter and 25 to the second parameter to easily add new parameters in between them.

General (Parameter)	
Name	AnotherSampleParameter
Description	
User Prompt	
Maintenance Unit	GolfStream Selection Parameters
Sort Order	5
Data Source	
General (Parameter)	
Name	SampleParameter
Description	
User Prompt	
Maintenance Unit	GolfStream Selection Parameters
Sort Order	25
Data Source	

## Naming

To quickly identify parameters, prefix parameter names with "Param". For example, ParamProductSegment, ParamCostCenter and ParamColumnColor.

Do not use special characters or dashes (-) in parameter names. Also, specify intuitive names for parameters. If you rename a parameter, you must re-assign it to all, potentially hundreds, of its associated objects.

## About Parameter Definitions

This topic orients you with the properties you specify to create parameters and how to look up objects and values to specify as the default value of a parameter.

## Property Types

You specify two types of properties to create a parameter, as shown below:

- General properties, such as user prompt and sort order, that are common to all parameter types.
- Data source properties that are [parameter type-specific](#). These properties define the value or objects to apply with the parameter or the prompt options that users can select at runtime.

General (Parameter)	
Name	ParamUserName
Description	User name entry
User Prompt	Please enter your name:
Maintenance Unit	GolfStream Selection Parameters
Sort Order	5
Data Source	
Parameter Type	Input Value
Default Value	J. Khan
	...

See [Create a Parameter](#).

## Create a Parameter

Parameters prompt users to filter or customize data at runtime, helping tailor objects such as forms, dashboards, and cube views to best suit different user needs. Administrators define parameters in dashboard maintenance units, then assign them to the appropriate objects.

**TIP:** To further refine data selections, refer one parameter to another. See [Nested Parameters](#).

1. Review the [Requirements](#) and [Best Practices](#).
2. Click **Application > Presentation > Dashboards**

**TIP:** You can also define parameters by clicking **System > Administration > Dashboards**.
3. Click the dashboard maintenance unit in which to create a parameter, then **Parameters**.  
After you save a parameter, you cannot put it in another dashboard maintenance unit.
4. Click  **Create Parameter** in the dashboard toolbar.
5. In **Name** and **Description**, enter a name and description that indicates the information the parameter prompts users to specify, such as *ParamBudget Scenario*. Do not use special characters, dashes (-) or underscores (\_) in names.  
See [Naming](#) in *Parameter Best Practices* for tips on naming and organizing parameters.
6. In **User Prompt**, enter the message that prompts users to specify information. For example, *Select a Product member* or *Enter your name*.  
Skip this step for literal value parameters since they do not prompt for selections.
7. In **Sort Order**, enter the sequence in which to display the parameter:
  - In the list of parameters for the dashboard maintenance unit. For example, to list the parameter sixth, enter six.
  - For use with other parameters. For example, to display parameterB after parameterA, ensure that the sort order value for parameterB is greater than that of parameterA. See "Best Practices" on page 196.
8. From **Parameter Type**, select the kind of parameter to define which determines the information or objects users are prompted to specify. See [Parameter Types](#).
  - **Literal Value:** Instead of prompting users for a selection, use a predefined value or object such as a color or number format.
  - **Input Value:** Users enter or modify a value to customize cube view results that generate reports. For example, prompt users to enter their name when they run a cube view to generate a report.

- **Delimited List:** Users select one value or item from a list, such as one of two budget scenarios.
- **Bound List:** Users select a member from a list of members generated with a method or a SQL query.
- **Member List:** Users select a member from a drop-down list.
- **Member Dialog:** Users select a member by navigating, searching, and filtering the member hierarchy displayed in a dialog box.

9. Define the settings for the parameter type:

- [Specify Literal Value Settings.](#)
- [Specify Input Value Settings.](#)
- [Specify Delimited List Settings.](#)
- [Specify Bound List Settings.](#)
- [Specify Member Dialog Settings.](#)
- [Specify Member List Settings.](#)

10. Click **Save**.

11. Add the parameter to the appropriate object, such as a cube view. See [Assign a Parameter](#).

## Specify Literal Value Settings

To finish defining a Literal Value parameter:

1. In **Default Value**, specify the value or object to apply with the parameter. Users cannot change this at run time. Enter a value (color or currency, for example) or an object name, or click **Edit** to [look it up](#).
2. Click **Save**.

**IMPORTANT:** When you assign the parameter, enclose the name in pipes and exclamation points. For example, `!ParamColumnColor!`.

## Specify Input Value Parameter Settings

To finish defining an Input Value parameter:

1. (Optional) Although users can override it at runtime, use **Default Value** to specify a value or object to initially display the first time the parameter runs. Enter the object name or value, or click **Edit** to [look it up](#).
2. Click **Save**.

**IMPORTANT:** When you assign the parameter, enclose the name in pipes and exclamation points. For example, `!DefaultHeader!`.

## Specify Delimited List Settings

To finish defining a Delimited List parameter:

1. In **Default Value**, specify a value or object to initially display the first time the parameter runs. Enter the member name or click **Edit** to [look it up](#).
2. In **Display Items** (comma-delimited), define a custom, comma-delimited list to display several options in the application from which to select, for example *Budget Version 1*, *Budget Version 2*. Separate names with commas.
3. In **Value Items** (comma-delimited), enter the values or objects, such as member names, that correspond to each Display Item option in the application. Enter options in a comma-delimited list using their names in the application. Separate the values with commas.

For example, if the Scenario dimension member for Budget Version 1 is BudV1, enter *BudV1*. Similarly, enter *BudV2* as the member used if Budget Version 2 is selected. Separate names with commas as shown:

General (Parameter)	
Data Source	
Parameter Type	Delimited List
Default Value	BudV1
Display Items (comma delimited)	Budget version 1, Budget version 2
Value Items (comma delimited)	BudV1, BudV2

The syntax you use when you assign the parameter to an object depends on:

- The object type and component, such as a cube view's page caption or row.
- If you want to present the Value Item or Display Item to users at run time.

See [Syntax](#).

**NOTE:** If a delimited list parameter is used in a cube view page caption, surround it in two exclamation points to reference the display items, not the value. For example, **||!ParameterName!!|**.

4. Click **Save** to save your property edits.

## Specify Bound List Settings

You can define two types of Bound List parameters.

- An SQL-based query to leverage data in the OneStream application or framework database or in an external data source.
- A method-based query to use object-specific variables to refine and customize the member data to return.

To finish defining a method-based parameter:

1. In **Default Value**, enter the actual value used by default in the parameter, if no other value is specified. The default value must be part of the parameter's definition.

You can either type the default value if known, use [object lookup](#) to create the default value, or use a [cube view](#) to select the appropriate formatting and copy the result to the Default Value field.

The Default Value property is the member you would set before defining a value using the Data Source properties specific to the selected parameter type.

2. Use **Result Format String Type** to determine if the default value or a custom value is used.

- **Default:** Use the formatting defined in the **Default Value** property.
- **Custom:** Use a format string that you enter in **Result Custom Format String** as the display text for the parameter.

Selecting **Custom** enables the **Result Custom Format String** property, where you can design the custom string.

3. In **Result Custom Format String**, optionally enter a string to use as the display text. Use `String.Format` syntax in .NET. For example, if a value is Blue and the format string is `The color is {0}.`, the display text is `The color is blue.`
4. In **Command Type** choose one of the following:
  - **Method** to use a specified method type and query to define the command to be used in the parameter.
  - **SQL Query** to determine the database to use and define the SQL query.

Both selections enable additional properties to further define the command.

**For a Method Command Type:** Use these properties to define the database location if using a Method command type.

- **Method Type:** Select the object-specific variables to define how members are evaluated, so the most useful members are returned. See Method Query for more information on using the method type in a parameter definition.
- **Method Query:** Customize variables to define which members to return and display in prompts. You can either type the default value if known, or use [object lookup](#) to create the default value. Use this syntax:

```
{Dimension type}{Dimension name}{Member Filter}{where clause}
```

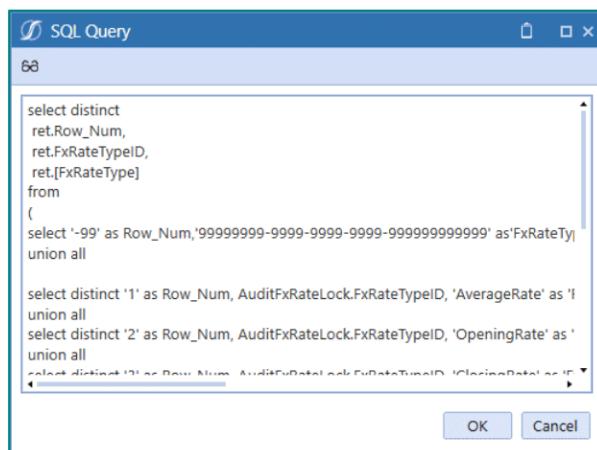
**For a SQL Command Type:** Use these properties to define the database location if using a SQL command type.

- **Database Location:** Determines the location of the database that includes the SQL command type. Select one of the following:
  - **Application:** Use the current OneStream Application database containing stage and financial cube data.
  - **Framework** Use the connected OneStream Framework database containing security and log data.
  - **External** Indicates that a database outside of OneStream is to be used.

If selecting **External** for the database, use the enabled **External Database Connection** property to select from the list of external databases. External database connections are defined in OneStream using the Server Configuration Tool. Select the external database connection you want to use from the list.

See *Configuring Application Servers* in the *Installation Guide* for information on configured application servers.

- **SQL Query:** This is the SQL statement to run for the parameter. Enter a query that evaluates objects (such as members) to return a particular set. For example:



**NOTE:** If you do not know the method query syntax, leave this field blank and click . This identifies the syntax location, runs the parameter, and provides a sample result set.

5. In **Results Table Name**, enter the name of the table generated when the data adapter runs. The default name is **table**.
6. In **Display Member**, enter one of the following:
  - **Name** to display the member by name.
  - **Description** to display the member's description.
  - **Name and Description** to display both the member name and the member description.

7. In **Value Member**, enter the name of the members, as used in the application, to correspond to each member displayed.
8. Click **Save** to save your property edits.

## Sample Method-Based Parameter

The following image shows the definition of a parameter that retrieves Time members for an Application Analysis report.

General (Parameter)	
Name	Members_Time_AllYears_Base_RTPA
Description	Generates list of Time members
User Prompt	Select a Time member
Maintenance Unit	
Sort Order	189
Data Source	
Parameter Type	Bound List
Default Value	WFTime
Result Format String Type	Default
Result Custom Format String	
Command Type	Method
Method Type	Members
Method Query	{Time}{{Time}{T*Root.Children.Base}{}}
Results Table Name	Members
Display Member	Name
Value Member	Name

## Sample SQL-Based Parameter

The following image shows the definition of a parameter that generates a list of FX Rate Types for an FX Rates Audit report.

General (Parameter)	
Name	FXRateTypes_RTPA
Description	Generates FX Rate types
User Prompt	
Maintenance Unit	
Sort Order	105
Data Source	
Parameter Type	Bound List
Default Value	(All) <span style="border: 1px solid #ccc; padding: 2px;">...</span>
Result Format String Type	Default
Result Custom Format String	
Command Type	SQL
Database Location	Application
SQL Query	union all <span style="border: 1px solid #ccc; padding: 2px;">...</span>
Results Table Name	FXRateTypes
Display Member	FXRateType
Value Member	FXRateTypeID

## Specify Member List Settings

To finish defining a Member List parameter:

1. (Optional) In **Default Value**, specify a member to display at the top of the list when the parameter runs and prompts users to select a member. Enter the member name or click **Edit** to [look it up](#).
2. Right-click in the **Default Value** dialog box, select **Paste** and then click **OK**.
3. Use **Display Member** to define how to display the returned members.
  - Type **Name** to display the member by name.
  - Type **Description** to display the member's description.
  - Type **Name and Description** to use both.
4. In **Cube**, either type the default value if known, or click **Edit** to use [object lookup](#) to select the cube containing the required dimensions.
5. From **Dimension Type**, select the dimension category (such as Time or Account) that contains the appropriate dimension.
6. In **Dimension**, enter the name of the dimension to use or click **Edit** to select the dimension in the hierarchy.

7. In **Member Filter**, either type the member filter if known or click **Edit** to use the [Member Filter Builder](#) to specify the members to list.
8. Click **Save** to save your changes.

## Specify Member Dialog Settings

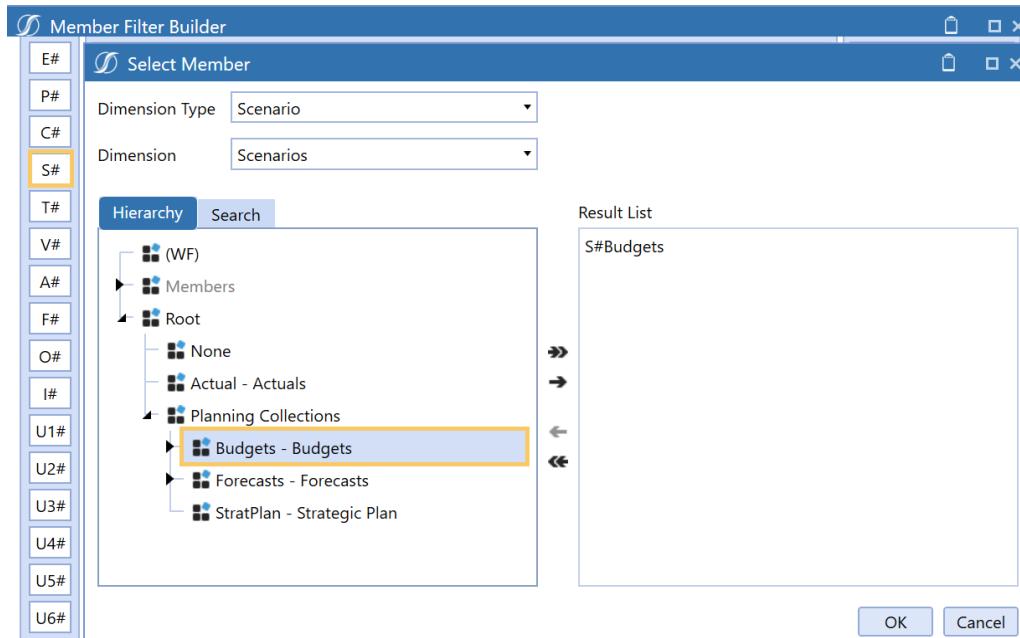
To finish defining a Member Dialog parameter:

1. (Optional) In **Default Value**, specify a dimension in the dimensional hierarchy to display when the parameter first runs. Enter the dimension name or click **Edit** to [look up](#) the dimension.
2. In **Cube**, click **Edit** to select the cube that uses the required dimension. To quickly find a cube, type the name in **Filter**.
3. From **Dimension Type**, select the dimension category such as Consolidation or Scenario that contains the appropriate dimension.
4. In **Dimension**, enter the name of the dimension to use or click **Edit** to select the dimension from the hierarchy.
5. In **Member Filter**, specify the member that determines the hierarchy to display in parameter prompts. Enter a member name or click **Edit** to use the [Member Filter Builder](#) to refine the members to display. For example:

## Filter Runtime Data and Reduce Maintenance with Parameters

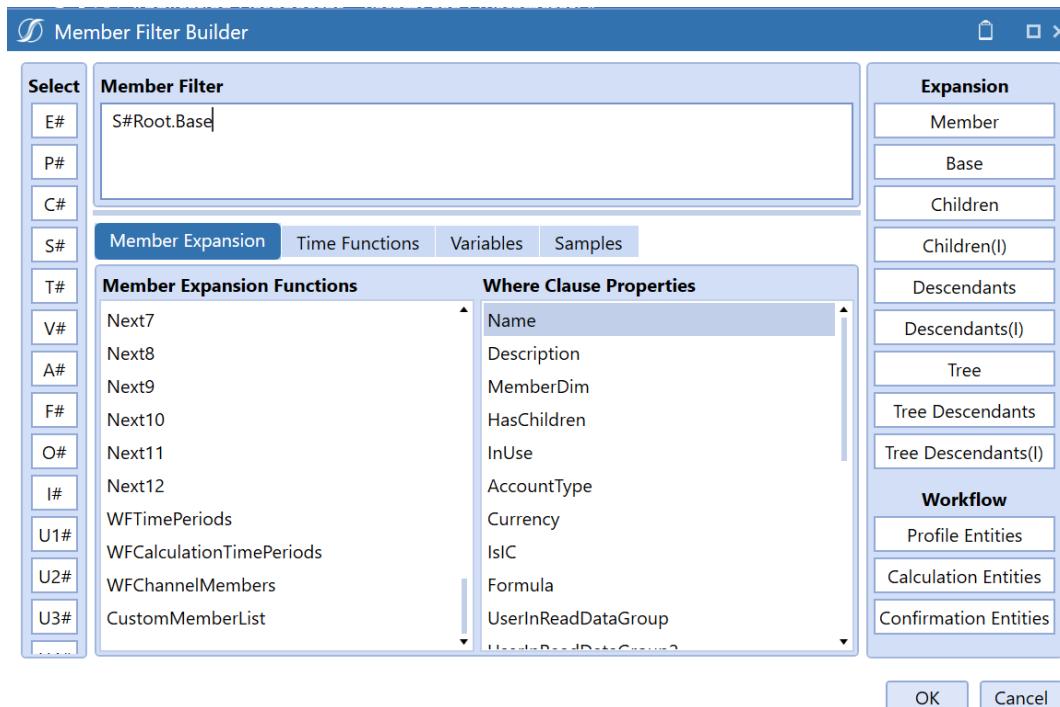
---

- To display only Budget members in a Scenario dimension with multiple member groups such as Actual and Forecast, click **S#** and select **Budgets** as shown:



## Filter Runtime Data and Reduce Maintenance with Parameters

- To display the entire hierarchy, select **Root** and **Base**.



**NOTE:** Dashboard-based parameters are similar to those used in forms. If a form references a parameter that cannot be found in the form template, a dashboard parameter of the same name is used.

- Select members and use the arrows to add them to **Results**.
- Click **OK** twice.
- Click **Save**.

**IMPORTANT:** When you assign the parameter, enclose the name in pipes and exclamation points. For example, **!BudgetScenario!**.

## Sample Delimited List Parameter in a Cube View

In this example, a Delimited List parameter is created and assigned to a sample GolfStream cube view so users can indicate if reports are not started, under review, or completed for each entity.

## Filter Runtime Data and Reduce Maintenance with Parameters

---

1. Click **Application > Presentation > Dashboard** and then select the dashboard maintenance unit where you want to store the parameter.
2. Click **Create New Parameter**.
3. Define the parameter by specifying the settings shown below. Enter an intuitive name to reflect the parameter's purpose of prompting users to select from a list of report status settings.

General (Parameter)	
Name	ParamReportStatus
Description	Status selection parameter for report analysis
User Prompt	Report status:
Maintenance Unit	GolfStream Selection Parameters
Sort Order	82
Data Source	
Parameter Type	Delimited List
Default Value	<input type="button" value="..."/>
Display Items (comma delimited)	Not Started, Under Review, Completed
Value Items (comma delimited)	0NotStarted, 1Review, 2Complete

4. In **Display Items**, enter the status options that users can select when they run the cube view.
5. In **Value Items**, enter the corresponding status member names defined in the application. For example, *0NotStarted* is the Value Item for the Display Name *Not Started*.
6. Click **Save**.
7. Click **Application > Presentation > Cube Views** and select the cube view to use with the parameter.

## Filter Runtime Data and Reduce Maintenance with Parameters

---

8. Click **Rows and Columns** to see that column members display time based on workflow and a non-financial account called Report Status. The row members identify all base entities for Clubs.

Column	ReportStatus
Time	T#WF
Account	A#[Report Status]

9. Click a row header and then **Data**.
10. In **Cell Type**, select **Combo Box**.
11. In **List Parameter**, enter **ParamReportStatus** and then save the cube view.

When users run the cube view, they can specify a report status for each entity, as shown:

The screenshot shows a Data Explorer window titled "Report Status". On the left, there is a tree view of entities: Montreal, Quebec City, Augusta, Carlsbad, Houston Heights, South Houston, and Frankfurt. To the right of the tree view, there is a table with two columns: "Report Status" and "Status". The "Report Status" column contains dropdown menus for each entity. The "Status" column lists the current status for each entity: Not Started, Under Review, Completed, or No Data. The "Houston Heights" entry has a dropdown menu open, showing "No Data", "Not Started", "Under Review", and "Completed".

Report Status	Status
Montreal	Not Started
Quebec City	Under Review
Augusta	Completed
Carlsbad	Under Review
Houston Heights	No Data
South Houston	No Data
Frankfurt	Not Started Under Review Completed

## Manage Parameters

This topic describes the tasks administrators can perform to maintain parameters. See:

- [Assign a parameter](#)
- [Modify a parameter](#)
- [Copy a parameter](#)

- [Delete a parameter](#)
- [Rename a parameter](#)
- [Identify assignments](#)
- [Nest parameters](#)

## Assign a Parameter

After defining a parameter, assign it to the object where it should be used to filter runtime data or apply a stored value.

1. Click **Application** and then the appropriate object.
2. Edit the object to assign the parameter. Click **Object Lookup** to find the parameter, and enclose the parameter name in pipes and exclamation points. For example, `!|ParamDate!|`. See [Syntax](#).
3. Click **Save**.

See these sample use cases:

- [Assign to a Report Subheader](#)
- [Assign to a Cube View Column Header](#)

## Assign to a Report Subheader

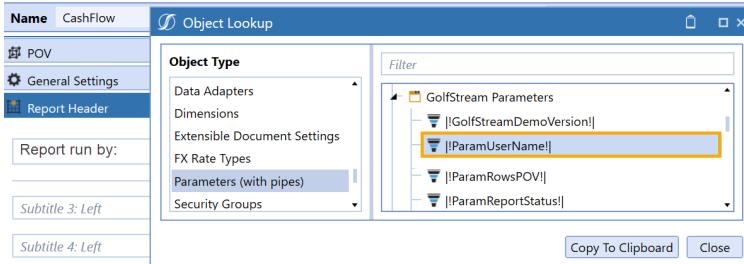
Perform these task to assign an Input Value parameter to prompt users to enter their name in a report header:

1. Click **Application > Presentation > Cube Views**.
2. Click **Cube View Group**, the appropriate cube view, **Designer**, and then **Report Header**.
3. In **Subtitle** enter **Report run by**:
4. Click **Object Lookup**, and then **Parameters (with pipes)**.

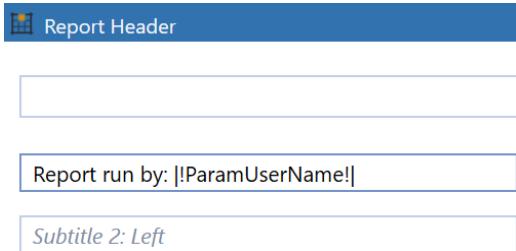
## Filter Runtime Data and Reduce Maintenance with Parameters

---

5. Select the parameter to assign, and then click **Copy to Clipboard**.



6. Paste the parameter after **Report run by:**, as shown:



7. Click **Save** and then **Open Data Explorer** to run the cube view.

8. Enter your name and click **Show Report**. The report generates, displaying your name as shown:

**GolfStream™**  
Get Back to the Green  
Report run by: Clare Byers

### Product Sales

	Budget V1	Budget V2	Final Budget
Total Product	534,686.10	534,686.10	534,686.10
GolfBalls	360,843.60	360,843.60	360,843.60
Tour	109,858.60	109,858.60	109,858.60
Premium	81,400.00	81,400.00	81,400.00
Range	45,570.00	45,570.00	45,570.00
Logo	83,995.00	83,995.00	83,995.00
LogoTour	40,020.00	40,020.00	40,020.00
Clubs	143,019.50	143,019.50	143,019.50
Woods	106,931.00	106,931.00	106,931.00
Drivers	50,400.00	50,400.00	50,400.00
FairwayWoods	56,531.00	56,531.00	56,531.00
Irons	0.00	0.00	0.00
Hybrids	0.00	0.00	0.00
Wedges	0.00	0.00	0.00
Putters	36,088.50	36,088.50	36,088.50
Conventional	0.00	0.00	0.00
Long	36,088.50	36,088.50	36,088.50

## Assign to a Cube View Column Header

In this example, a Delimited List parameter is assigned to a sample GolfStream cube view so users can indicate if reports are not started, under review, or completed for each entity.

1. Click **Application > Presentation > Cube Views**.
2. Click the appropriate cube view.

## Filter Runtime Data and Reduce Maintenance with Parameters

---

3. Click **Rows and Columns** to see that column members display time based on workflow and a non-financial account called Report Status. The row members identify all base entities for Clubs.

Column	ReportStatus
Time	T#WF
Account	A#[Report Status]

4. Click a row header, and then **Data**.
5. In **Cell Type**, select **Combo Box**.
6. In **List Parameter**, enter **ParamReportStatus** and save the cube view.

When users run the cube view, they can specify a report status for each entity, as shown:

The screenshot shows a Data Explorer window titled "Data Explorer - Report Status". The main pane is titled "Report Status" and contains a list of entities with their corresponding report statuses. The entities listed are Montreal, Quebec City, Augusta, Carlsbad, Houston Heights, South Houston, and Frankfurt. The report statuses are Not Started, Under Review, Completed, and No Data. The "Houston Heights" entry has a dropdown menu open, showing "No Data" as the selected option. The "South Houston" entry also has a dropdown menu open, showing "No Data" as the selected option.

Entity	Report Status
Montreal	Not Started
Quebec City	Under Review
Augusta	Completed
Carlsbad	Under Review
Houston Heights	No Data
South Houston	No Data
Frankfurt	Not Started Under Review Completed

## Modify a Parameter

**TIP:** Before editing a parameter, identify where it is used to understand the affects of modifying it. Select the parameter and click **Show Objects That Reference The Selected Item**.

1. Click **Application > Presentation > Dashboards**.
2. Click the dashboard maintenance unit that uses the parameter, and then **Parameters**.

3. Click the parameter to modify settings described in [Create a Parameter](#).
4. Click **Save**.

## Copy a Parameter

Copy a parameter to quickly create a new parameter that must function similarly.

1. Click **Application > Presentation > Dashboards**.
2. Click **Dashboard Maintenance Unit**, the dashboard maintenance unit in which the parameter is used, and then **Parameters**.
3. Click the parameter upon which to base a new parameter, then click **Copy**.
4. Click **Paste** and then **Rename** to specify a name for the new parameter.
5. Modify the new parameter as needed and assign it to an appropriate object. See:
  - [Create a Parameter](#)
  - [Assign a Parameter](#)

## Delete or Remove a Parameter

You can delete a parameter after removing it from its assigned objects.

1. Click **Application > Presentation > Dashboards**.
2. Click **Dashboard Maintenance Unit**, then the dashboard maintenance unit in which the parameter is used.

## Filter Runtime Data and Reduce Maintenance with Parameters

---

3. Click the parameter, then **Show Objects That Reference The Selected Item**. This identifies the objects from which you must remove the parameter. For example:

The screenshot shows the 'Object Usage Viewer' window with the title 'ParamRowsPOV (Dashboard Parameter)'. The table has three columns: Type, Name, and Description. The 'Type' column includes categories like 'Dashboard Parameter', 'Cube View', 'Cube View Group', 'Cube View Profile', 'Dashboard DataAdapter', 'Dashboard Component', and 'Dashboard Component'. The 'Name' column lists specific objects such as 'ParamRowsPOV', 'Quick Analysis Wizard', 'Financial Reports (General)', etc. The 'Description' column provides details like 'DelimitedList' and 'Quick Analysis Wizard - |!!ParamRowsPOV!!| (Drillable)'. A 'Close' button is visible in the bottom right corner.

Type	Name	Description
Dashboard Parameter	ParamRowsPOV	DelimitedList
Cube View	Quick Analysis Wizard	
Cube View Group	Financial Reports (General)	
Cube View Profile	Financial Reports (General)	
Dashboard DataAdapter	Quick Analysis Wizard	
Dashboard Component	Quick Analysis Wizard	Quick Analysis Wizard -  !!ParamRowsPOV!!  (Drillable)
Dashboard Component	Quick Analysis Wizard Rpt	Quick Analysis Wizard (Report)

4. Click **Application** to find and edit the objects that use the parameter. For example, to remove a parameter from report header in a cube view:
  - a. Click **Cube View Groups**, the cube view, and then **Designer**.
  - b. Click **Report Header**, delete the parameter from the subtitle, and then click **Save**.
5. To delete the parameter:
  - a. Repeat steps 2 - 4 to ensure you remove the parameter from all objects.
  - b. Click the parameter and then **Delete**.

## Rename a Parameter

**CAUTION:** If you rename a parameter, you must re-assign it to all objects because the new name is not automatically applied. See [Identify parameter assignments](#). Do not use special characters and dashes (-) in names.

## Filter Runtime Data and Reduce Maintenance with Parameters

---

1. Click **Application > Presentation > Dashboards**.
2. Click the dashboard maintenance unit that uses the parameter, and then **Parameters**.
3. Click the parameter, then **Show Objects That Reference The Selected Item** to identify the objects using the parameter. You must assign the renamed parameter to these objects. For example:

The screenshot shows the 'Object Usage Viewer' interface with a title bar 'Object Usage Viewer'. Below it is a table titled 'ParamRowsPOV (Dashboard Parameter)'. The table has three columns: 'Type', 'Name', and 'Description'. The 'Type' column includes categories like 'Dashboard Parameter', 'Cube View', 'Cube View Group', 'Cube View Profile', 'Dashboard DataAdapter', 'Dashboard Component', and 'Dashboard Component'. The 'Name' column lists specific objects such as 'ParamRowsPOV', 'Quick Analysis Wizard', 'Financial Reports (General)', 'Financial Reports (General)', 'Quick Analysis Wizard', 'Quick Analysis Wizard', and 'Quick Analysis Wizard Rpt'. The 'Description' column provides details like 'DelimitedList' and 'Quick Analysis Wizard - !!!ParamRowsPOV!!! (Drillable)'.

ParamRowsPOV (Dashboard Parameter)		
Type	Name	Description
Dashboard Parameter	ParamRowsPOV	DelimitedList
Cube View	Quick Analysis Wizard	
Cube View Group	Financial Reports (General)	
Cube View Profile	Financial Reports (General)	
Dashboard DataAdapter	Quick Analysis Wizard	
Dashboard Component	Quick Analysis Wizard	Quick Analysis Wizard - !!!ParamRowsPOV!!! (Drillable)
Dashboard Component	Quick Analysis Wizard Rpt	Quick Analysis Wizard (Report)

4. Click **Close**.
5. Click the parameter, then **Rename** and enter a different name.
6. Click **Save**.

## Identify Parameter Assignments

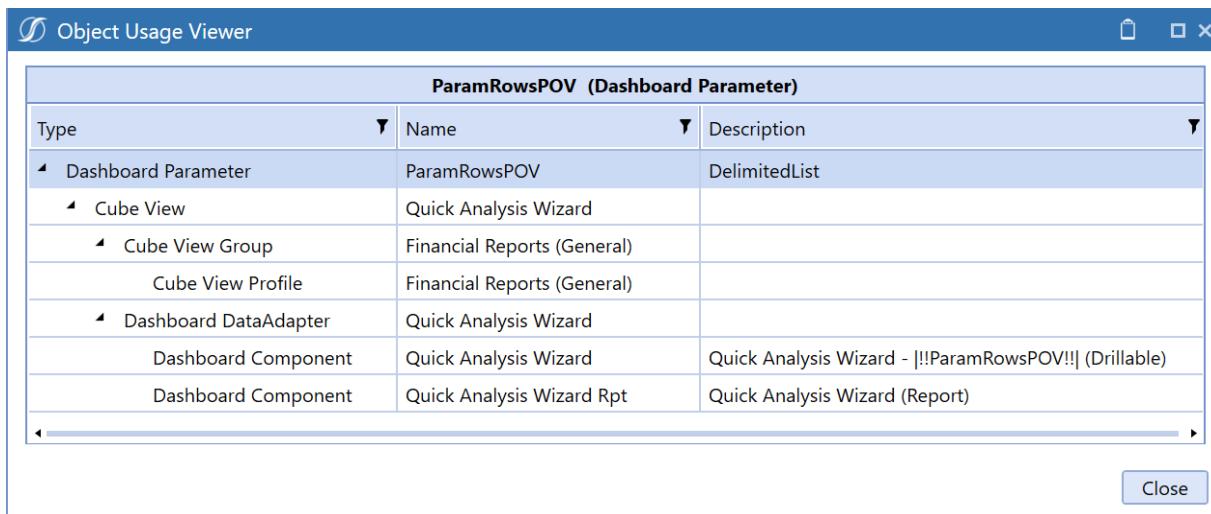
Perform these steps to access a list of objects to which a parameter is assigned:

1. Click **Application > Presentation > Dashboards**.
2. Click the dashboard maintenance unit associated with the parameter, and then **Parameters**.

## Filter Runtime Data and Reduce Maintenance with Parameters

---

3. Click the parameter and then **Show Objects That Reference The Selected Item**. This lists all the objects that use the parameter.



## Nest Parameters

A nested parameter refers to another parameter, so you can further refine data selections for more granular data entry, reporting, and analysis. The following procedure based on the GolfStream application, describes how to create a nested parameter for a cube view to display profit by product. The parameters defined below let users filter data by product segment, then by a specific product in a selected segment.

1. Click **Application > Presentation > Dashboards**.
2. Click the dashboard maintenance unit in which to store the parameters.
3. Click **Parameters**, and then **Create Parameter**.

## Filter Runtime Data and Reduce Maintenance with Parameters

---

4. Define the first parameter as shown below to prompt users to select a product segment.

General (Parameter)	
Name	ParamProductSegments
Description	
User Prompt	Select Product Segment
Maintenance Unit	GolfStream Selection Parameters
Sort Order	310
Data Source	
Parameter Type	Member List
Default Value	Clubs
Display Member	
Cube	GolfStream
Dimension Type	UD2 (SalesRegion)
Dimension	CorpProducts
Member Filter	UD2#Top.Children

The Member Filter returns the child members of Top, in the UD2 dimension, in a drop-down list so users can select a product segment. For example, if Top has 2 children, TotalProducts and TotalServices, users can select only those members.

5. Define the second parameter as shown below, referencing the first parameter in Member Filter.

General (Parameter)	
Name	ParamBaseProducts
Description	
User Prompt	Select Product
Maintenance Unit	GolfStream Parameters
Sort Order	31
Data Source	
Parameter Type	Member List
Default Value	
Display Member	
Cube	GolfStream
Dimension Type	UD2
Dimension	CorpProducts
Member Filter	UD2#{ParamProductSegments}.Base

The nested parameter will list the selected product segment's base members, so users can select a base-level member in TotalProducts orTotalServices.

6. Save both parameters, and then click **Application > Presentation > Cube Views**.
7. Select the cube view where you want to use the parameters.

## Filter Runtime Data and Reduce Maintenance with Parameters

---

8. Select the column or row that requires a parameter and enter the parameter name using pipes and exclamation points. See [Syntax](#). The sample cube view below displays products in columns, so a column is selected.

A screenshot of a cube view interface. At the top, there's a header bar with 'Column' and 'Products'. Below it, there are two dropdown menus: 'UD2' which is currently set to 'U2#|!ParamBaseProducts|', and '(Not Used)' which has 'Level 2: Member Filter' selected. There are also two edit icons next to the dropdowns.

9. Select a dimension and click **Member Filter Builder** to ensure the ParamBaseProducts parameter references the ParamProductSegments parameter first, then returns a product list based on a selected segment.
10. Click **Open Data Explorer** to see how the nested parameters work. ParamProductSegments runs first, prompting you to select a segment.

A screenshot of the Open Data Explorer. On the left, there's a tree view with 'Parameters' expanded, showing 'Select Product Segment'. To the right, a dropdown menu is open with a list of product segments: Clubs, Balls, Apparel, Accessories, Electronics, Parts, Subassemblies, Services, and None. The 'Clubs' option is highlighted.

Then ParamBaseProducts runs, prompting you to select a product.

A screenshot of the Open Data Explorer. On the left, there's a tree view with 'Parameters' expanded, showing 'Select Product'. To the right, a dropdown menu is open with a list of products: Mach5, Mach10, Elite, Launcher, Hybrid LT, Hybrid XF, Hybrid SL, Iron LT, and Iron XF. The 'Mach5' option is highlighted.

# **Presenting Data With Extensible Documents**

The Extensible Document Framework blends OneStream content with Microsoft Office and text files. In turn, these documents become known as Extensible Documents. You can create Extensible Documents by selecting the type of document desired, which can be a Microsoft Word Document, Excel spreadsheet, PowerPoint presentation, or a text File. Extensible Documents allow you to display any information you want from OneStream, and because it is integrated with these different products, the data stays current and dynamic.

This section describes how to work with extensible documents.

## **Extensible Document Framework**

The Extensible Document Framework blends OneStream content with Microsoft Office and text files. In turn, these documents become known as extensible documents.

You can create extensible documents by first selecting the type of document desired, which can be a Microsoft Word document, Excel spreadsheet, PowerPoint presentation, or text file. After selecting the document, use OneStream's custom parameters, substitution variables, and retrieve functions to get specific information such as time periods, entity names, data cell values, text comments, and other information. You can also insert images into a Word document, PowerPoint presentation, or Excel spreadsheet for reports, charts, Cube Views, Excel spreadsheets, or PDF files.

After you configure an extensible document, save it as [NameofDocument].xfdoc.ext, where the xfdoc portion of the file name tells OneStream that this is an extensible document (for example, CostSpreadsheet.xfdoc.xlsx). Load the document into OneStream's file share to launch it from OneStream or include it in dashboards or report books.

This feature's value is that special integration tools are not necessary because once the document is launched, it updates itself with the correct parameter value, image, or retrieve function values. Extensible documents allow you to display any information you want from OneStream and because it is integrated with these different products, and the data stays current and dynamic.

## About Creating Extensible Documents

### Extensible Document Creation Process

To create an extensible document, begin with the Microsoft document or text file needed to create the framework. Next, decide the type of data required and insert parameters, images, or retrieve functions needed to pull data from OneStream and dynamically update the extensible document. Once the document is complete, save it as follows (all examples below are using the file name `GolfStream Report`):

#### Saving a Word Document

`GolfStream Report.xfdoc.docx`

#### Saving an Excel Spreadsheet

`GolfStream Report.xfdoc.xlsx`

#### Saving a PowerPoint Presentation

`GolfStream Report.xfdoc.pptx`

#### Saving a Text File

`GolfStream Report.xfdoc.txt`

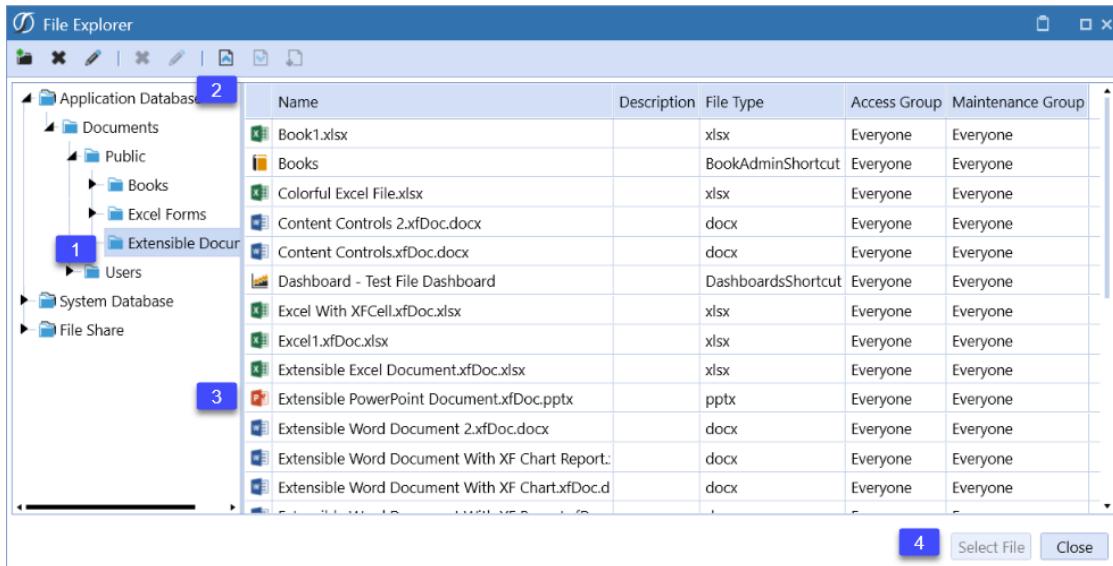
**NOTE:** Extensible documents only work with Microsoft Office version 2007 and later.

Once the document is saved in the format mentioned above, upload the document into OneStream's File Explorer.

Click  in OneStream to launch the **File Explorer** dialog box.

## Presenting Data With Extensible Documents

---



Select the folder in which to save the extensible document.

1. Upload the document into OneStream.
2. Click on the desired file name.
3. Click **Select File** to launch the document and see the updated values.

## Using Parameters and Substitution Variables in Extensible Documents

When creating extensible documents, use parameters as placeholders for OneStream member names, data, or user comments. Place a parameter in different areas of a document where specific information is needed. For example, if the `!MyEntity!` parameter was used in the document's framework, once it was launched from OneStream, the parameter would be recognized and replaced with the desired entity name.

Some parameters may prompt the user to select a specific entity, account, time period, or other data to view the document correctly. Parameters can be used in any Word document, PowerPoint presentation, Excel spreadsheet, or text file. Any parameter that exists in OneStream can be used in an extensible document. See [Parameters](#) for more details.

See one of the Creating an Extensible Document sections in this section for examples on how to incorporate parameters into an extensible document.

Substitution variables can also be used in Word, Excel, PowerPoint, or a text file. These variables call out details such as an Application Name |AppName|, User Name |UserName|, or refer to a specific POV which creates versatility when reusing the same document. See [Substitution Variables](#) for details.

See "Creating a Document in Microsoft PowerPoint" on page 237 for an example on how to incorporate substitution variables into an extensible document.

## Using Images in Extensible Documents

You can display a variety of reports, charts, Cube Views, Excel spreadsheets, and PDFs in an extensible document. Any one of these items can be used in a Word document, PowerPoint presentation, or Excel spreadsheet. This works by inserting any image into the document, configuring the image with the correct reporting information under the **Format Picture** right-click option. When launched from OneStream, the image is replaced by the desired report.

For more details on an image's configuration, refer to the extensible document settings in the **Object Lookup** dialog box in OneStream, or see [Extensible Document Settings](#).

See [Creating an Extensible Document in Microsoft Word](#) for an example on how to insert a report into an extensible document.

## Using Retrieve Functions in Extensible Document

You can use retrieve functions such as XFGetCell in Excel. These functions run on the server when launched from OneStream. Log into the Excel Add-In and click **Refresh Data** to display the updated Excel spreadsheet.

You can also create Excel charts based on the retrieve values. These charts display the correct data when you launch the spreadsheet from OneStream. Use the XFGetCellVolatile retrieve function to refresh the Excel chart and display updated data. Excel requires a volatile function for proper refreshing when using charts that reference calculated cells.

See [Creating an Extensible Document in Microsoft Excel](#) for an example on how to use retrieve functions with extensible documents.

There is also another type of retrieve function called XFCell which retrieves data from a single cell in OneStream. This is intended for text documents such as Word or PowerPoint. For example, XFCell(A#20500:E#Clubs) returns a value for the account and entity intersection.

Parameters and substitution variables may also be used in an XFCell formula. For example, XFCell(A#20500:E#| !MyEntityParameter! | :T#|Global| ) returns a value for the specified account, the entity selected at run-time, and your application's global time period.

You can include additional settings in an XFCell function used within an extensible document to format resulting data (for example, XFCell(A#20500:E#Clubs, Culture=User, NumberFormat=N3, DisplayNoDataAsZero=True, Scale=3, FlipSign=True, ShowPercentSign=False).

**NOTE:** Any dimensions not specified in the formula come the current POV.

For more examples and details on XFCell's syntax, refer to the extensible document Settings in the **Object Lookup** dialog box in OneStream, or see [Extensible Document Settings](#) under Object Lookup in "Presenting Data With Books, Cube Views and Dashboards" on page 547.

See [Creating an Extensible Document in Microsoft PowerPoint](#) for an example on how to use XFCell with extensible documents.

## Using Rich Text Content Controls

Rich text content controls are used in Microsoft Word only and allow users to add a Cube View, a dashboard report, a Word document (.docx or .xfDoc.docx), a rich text file (.rtf), or a text file (.txt) to any extensible Word document. When the extensible document is launched, the content control is replaced with formatted text that can be edited and reformatted as desired. See [Creating an Extensible Document in Microsoft Word](#) for an example on how to insert text using rich text content controls.

**NOTE:** In order to embed any type of file (Word document, rich text file or text file), it must be saved in the OneStream File Explorer. This allows the extensible document to access the file's content at run-time.

## Creating a Document in Microsoft Word

The example below was taken from a company's quarterly report made in Microsoft Word. It shows how to complete the following:

- Insert a Cube View into a Word document.
- Use parameters to display information from OneStream.
- Insert a portion of a Word document using rich text content controls.

**NOTE:** When inserting content into a Word document, the embedded content takes on the page settings of the main document. To change the page settings for the inserted content, add a section break before the inserted content. After the section break, specify the desired page settings for the embedded content.

## Insert a Cube View

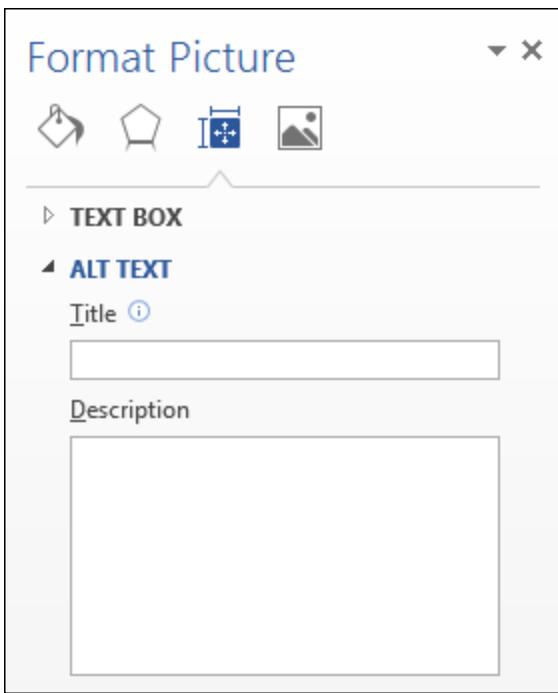
**NOTE:** The same format for inserting an image is used in Microsoft PowerPoint and Excel. Also, as of a Microsoft upgrade Office version 1804 and higher, Microsoft has changed how the **Alt Text Title** property is stored for embedded images. This impacts the creation of extensible documents in OneStream. Existing extensible documents continue to run as expected as this only impacts newly created extensible documents if the user is on version 1804 (or higher) of Microsoft Office. Please see additional section below on this topic.

The following table sets forth the contribution to net sales attributable to the principal product groups for the periods indicated:



2 For a discussion regarding the changes in net sales for each product group from **!CurrentQtr!** **!ReportingYear!** to **!CurrentQtr!** **!ReportingYearPrior!** and from **!CurrentQtr!** **!ReportingYearPrior!** to 2009, see below, “Management’s Discussion and Analysis of Financial Condition and Results of Operations—Results of Operations” contained in Item 7.

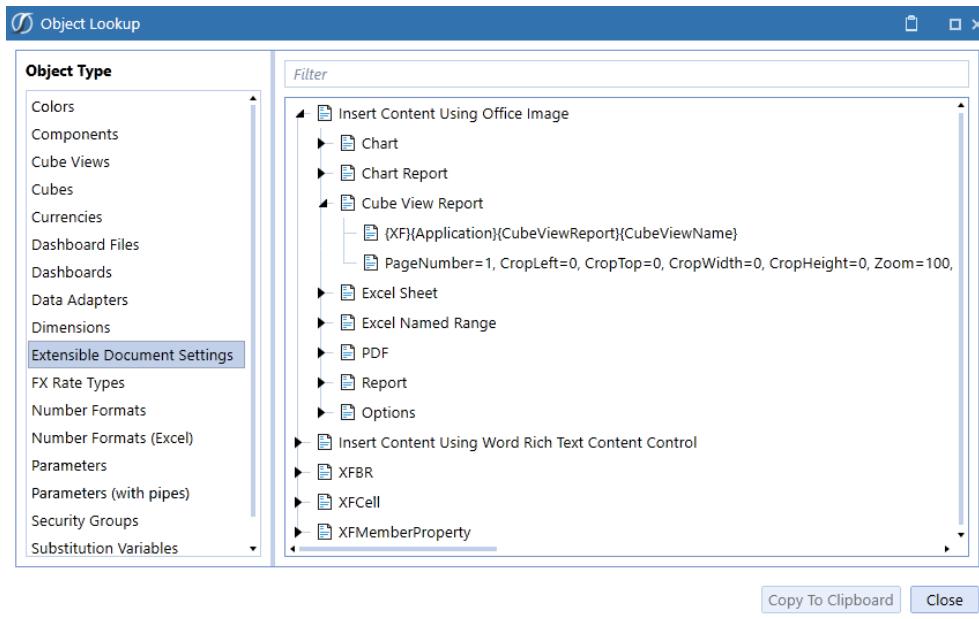
1. The OneStream logo is an image serving as a placeholder for a Cube View report. Any image can be used as a placeholder. Once the image is inserted into the document, right-click on the image and select **Format Picture**. Click **Layout & Properties** and expand **ALT TEXT**.



2. Navigate to the **Object Lookup** dialog box in OneStream. This dialog provides all the syntax needed to insert any type of report into an extensible document. This icon can be found on the following screens under the **Application** tab:
  - Form Templates
  - Books
  - Cube Views
  - Dashboards
  - Data Management

## Presenting Data With Extensible Documents

---



- Once in the dialog box, expand the desired item type (**Cube View Report** in this case) and click **Copy to Clipboard** to copy the first string. Go back into the Word document and paste it into the Title field.

The string `{XF}{Application}{CubeViewReport}{CubeViewName}` displays in the field.

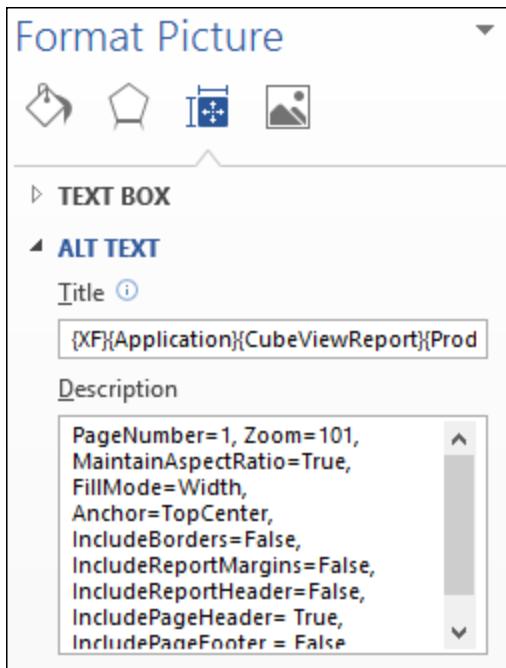
- Update the `{CubeViewName}` with the name of the Cube View required for this document. In this example, the following syntax was entered for the Cube View: `{XF}{Application}{CubeViewReport}{Product Sales}`.

**NOTE:** See [Extensible Document Settings](#) for more examples on these different item types.

- Navigate back to the **Object Lookup** dialog box and copy the second string under the desired item type (Cube View Report in this case). This is the standard formatting for the item type.
- Go back into the Word Document and paste the string into the Description field:

`pageNumber=1, zoom=101, maintainAspectRatio=True, fillMode=width, anchor=topCenter, includeBorders=False, includeReportMargins=False, includeReportHeader=False, includePageHeader=True, includePageFooter=False, includeReportFooter=False, includePageFooter=False`

**NOTE:** Any of these properties can be changed. See [Extensible Document Settings](#) for more formatting details and options.



## Insert a Cube View Using Microsoft Office Update version 1804 and higher

**NOTE:** The same format for inserting an image is used in Microsoft PowerPoint and Excel. OneStream version 4.5.0 and higher is required to process these extensible documents.

The following table sets forth the contribution to net sales attributable to the principal product groups for the periods indicated:



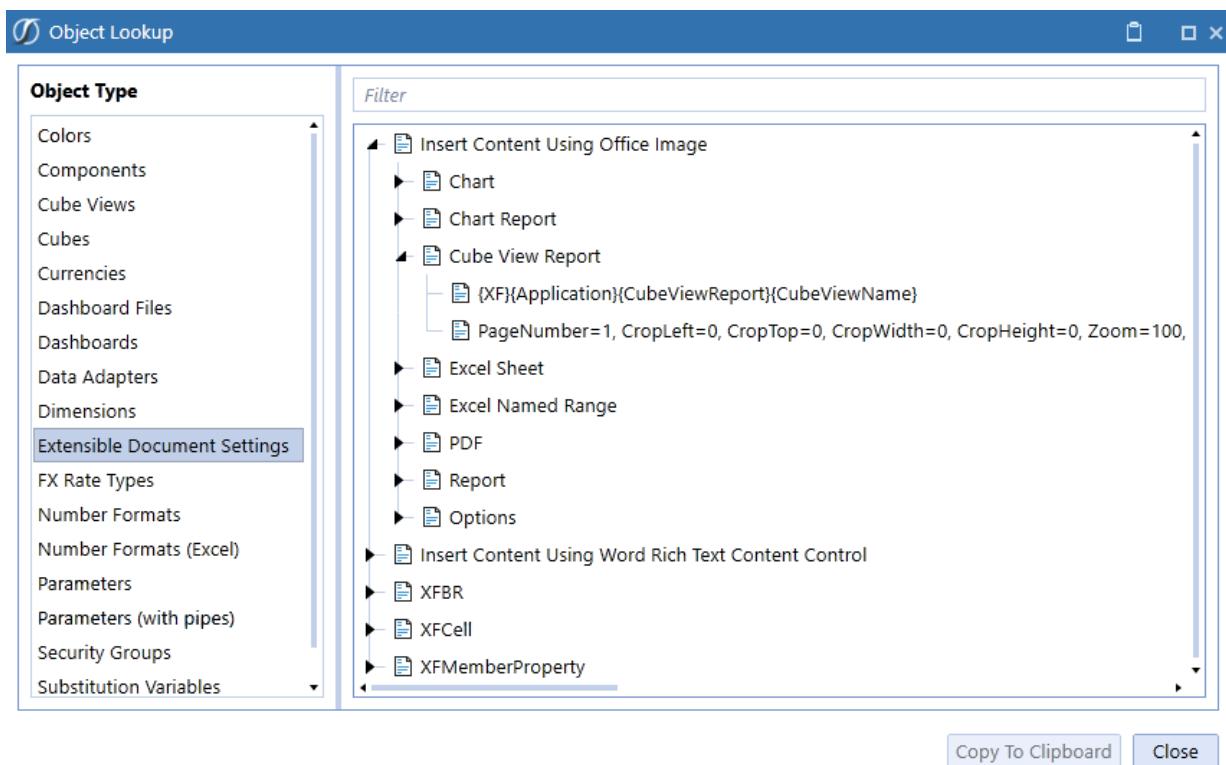
**2** For a discussion regarding the changes in net sales for each product group from **!CurrentQtr!** **!ReportingYear!** to **!CurrentQtr!** **!ReportingYearPrior!** and from **!CurrentQtr!** **!ReportingYearPrior!** to 2009, see below, “Management’s Discussion and Analysis of Financial Condition and Results of Operations—Results of Operations” contained in Item 7.

## Presenting Data With Extensible Documents

---

The OneStream Logo is an image serving as a placeholder for a Cube View Report. Any image can be used as a placeholder.

1. After the image is inserted into the document, right-click on the image and select **Edit Alt Text**.
2. Navigate to the **Object Lookup** dialog box located in the OneStream application. This dialog box provides all the syntax needed to insert any type of report into an extensible document. This icon can be found on the following screens under the Application tab.



3. Once in the dialog, expand the desired item type (Cube View Report in this case) and click **Copy to Keyboard** to copy the first string. Then go back into the Word Document and paste it into the Description Field.

The string `{XF}{Application}{CubeViewReport}{CubeViewName}` populates the field.

4. Update the `{CubeViewName}` with the name of the Cube View required for this document.

In this example, the string {XF}{Application}{CubeViewReport}{Product Sales} is entered for the Cube View

**NOTE:** See [Extensible Document Settings](#) for more examples on these different item types.

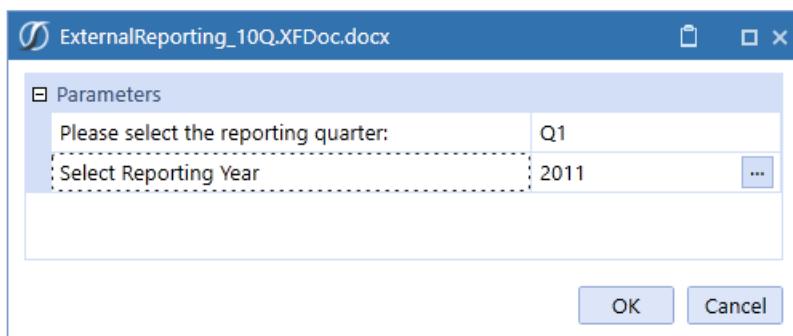
5. Navigate back to the Object Lookup and copy the second string under the desired item type (Cube View Report in this case). This is the standard formatting for the item type. Go back into the Word document and paste the string into the Description field.

The string PageNumber=1, Zoom=101, MaintainAspectRatio=True, FillMode=Width, Anchor=TopCenter, IncludeBorders=False, IncludeReportMargins=False, IncludeReportHeader=False, IncludePageHeader=True, IncludePageFooter=False, IncludeReportFooter=False, IncludePageFooter=False populates the field.

**NOTE:** Any of these properties can be changed. See [Extensible Document Settings](#) for more formatting details and options.

## Using Parameters

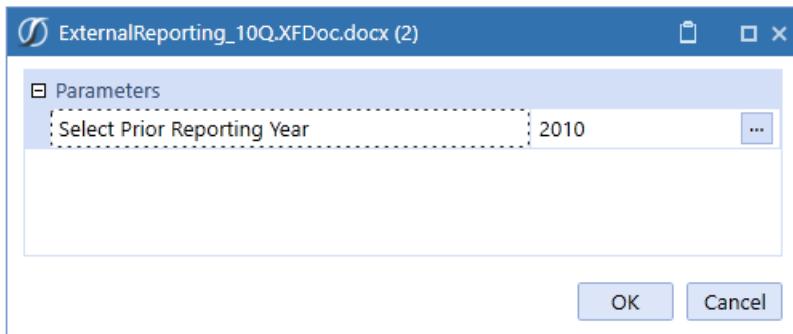
In this Word document example, all the information is typed normally, but parameters were inserted in places to obtain information from OneStream. Three parameters named `!CurrentQtr!`, `!ReportingYear!` and `!ReportingYearPrior!` were created to display the appropriate years and quarters for this document. When the extensible document is processed at run-time, it prompts to select the reporting quarter and reporting year.



Next, it prompts the user to select the prior reporting year:

## Presenting Data With Extensible Documents

---



After selecting parameters, the document displays the desired data.(Highlighted in yellow below)

The following table sets forth the contribution to net sales attributable to the principal product groups for the periods indicated:

1



Product Sales Book

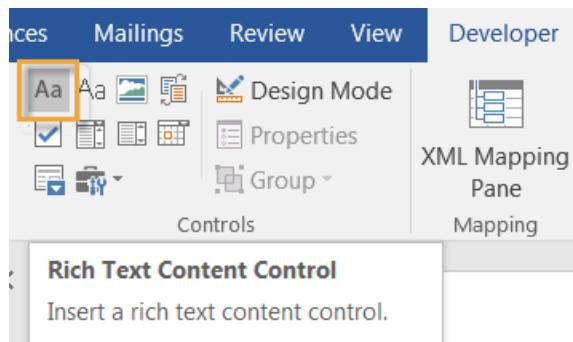
	Q1 2011	Q4 2010	Q3 2010	Q2 2010
Clubs	203,531	406,376	344,783	262,222
Accessories	53,447	0	0	0
Parts	1,068	3,310	2,808	2,135
Subassemblies	278	936	794	604
Services	23,494	63,536	53,903	40,991
Returns	(10,373)	(17)	(17)	(17)
<b>Product Sales</b>	<b>271,445</b>	<b>474,141</b>	<b>402,271</b>	<b>305,935</b>

- 2 For a discussion regarding the changes in net sales for each product group from Q1 2011 to Q1 2010 and from Q1 2010 to 2009, see below, “Management’s Discussion and Analysis of Financial Condition and Results of Operations—Results of Operations” contained in Item 7.

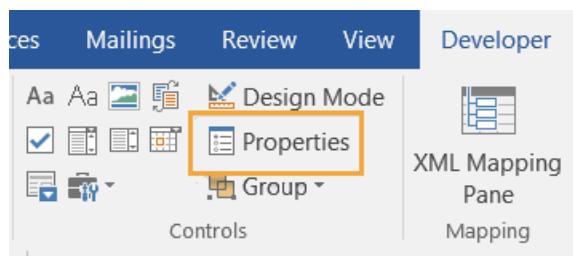
## Insert a Word Document through Rich Text Controls

1. In the Word extensible document, navigate to the section where the separate Word document needs to be inserted.
2. Click the Developer tab in the Word ribbon and click the Rich Text Content Control icon

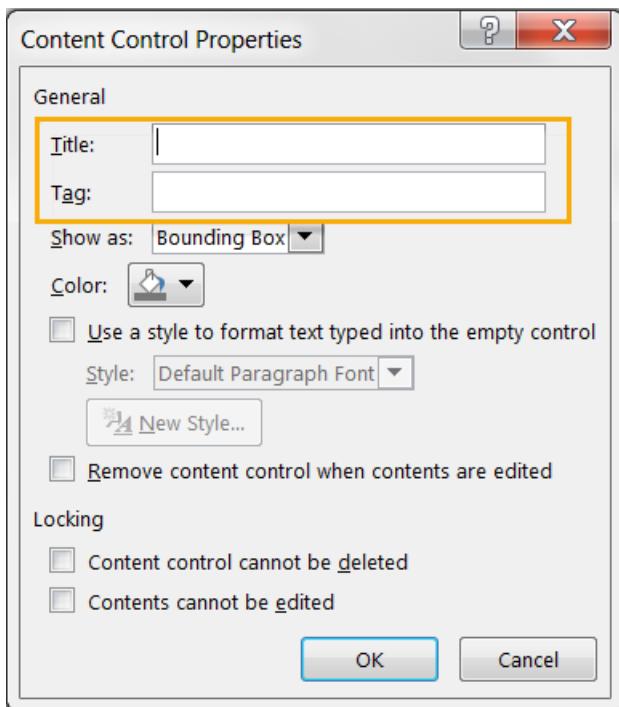
**TIP:** If the Developer tab is not visible, right-click on the ribbon, select **Customize the Ribbon**, and select the Developer check box.



3. Select the Content Control and click **Properties** on the Developer tab.



This opens the **Content Control Properties** dialog box. The Title and Tag fields define the extensible document content.

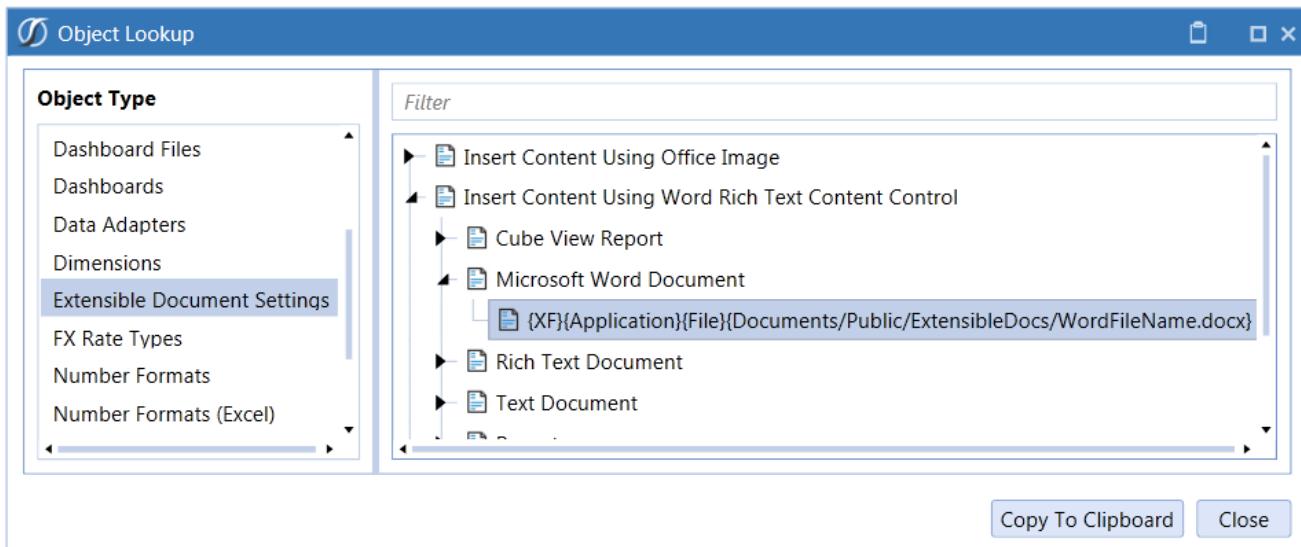


The Title field is where the content item is specified. The Tag field is where additional formatting settings are specified and only applies when embedding a Cube View or dashboard report.

4. Navigate to the Object Lookup  and select **Extensible Document Settings**. Expand **Insert Content Using Word Rich Text Control Content** and then expand **Microsoft Word Document**.

## Presenting Data With Extensible Documents

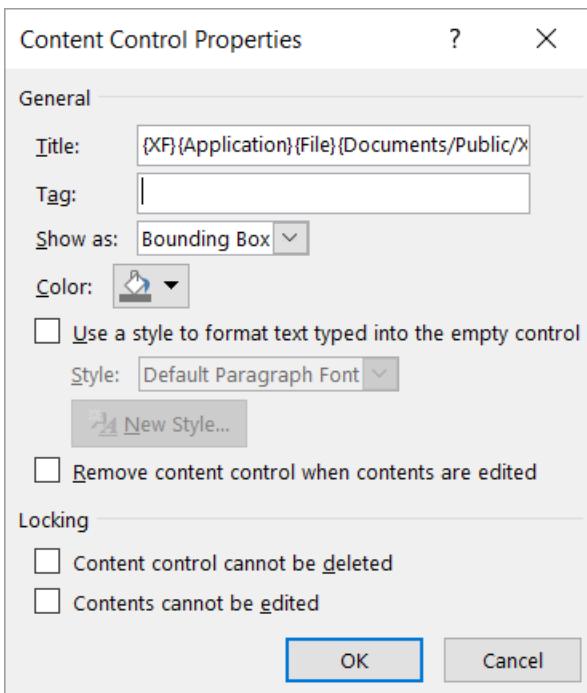
---



5. Select the string and click **Copy to Clipboard**, then navigate back to the Extensible Word Document and paste the string into the Title field. The string `{XF}{Application}{File}{Documents/Public/ExtensibleDocs/WordFileName.docx}` populates the field.
6. Edit the string's file path to specify the exact location of the Word document being embedded into the document. In this example, the final string is `{XF}{Application}{File}{Documents/Public/XF_Docs/Documents/Form10_Q.docx}`.

## Presenting Data With Extensible Documents

---



7. Ensure all changes are saved, navigate back into OneStream, and open the File Explorer.
8. Upload the extensible document into the File Explorer, select the file and launch it. The Word Document is now embedded in the extensible Word document and can be formatted as desired.

**UNITED STATES  
SECURITIES AND EXCHANGE COMMISSION  
Washington, D.C. 20549**

**FORM 10-Q**

**ANNUAL REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE SECURITIES EXCHANGE ACT OF 1934**

For the fiscal quarter ended March 31, 2011

TRANSITION REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE SECURITIES EXCHANGE ACT OF 1934

## Creating a Document in Microsoft PowerPoint

The example below is a slide from a management presentation. It shows how to complete the following:

- Use parameters
- Use substitution variables
- Use the XFCe11 retrieve function

**NOTE:** The same format for XFCe11 is used in Microsoft Word and text files.

## GolfStream Report

1

□ `|!GetCellEntity!|` results in thousands:

□ Total Operating Income:

3

□ `XFCe11(A#62000:E#[|!GetCellEntity!|]:T#|GlobalTime|,`

□ `NumberFormat=N0, Scale=3)`

1. A parameter named `|!GetCellEntity!|` is created to select a specific entity on which to base the data. When the extensible document is processed at run-time, it prompts to select the desired entity.
2. XFCe11 retrieves data from a single cell in OneStream. The example above retrieves data from Account 62000 for the entity selected at run-time, for the application's global time period.

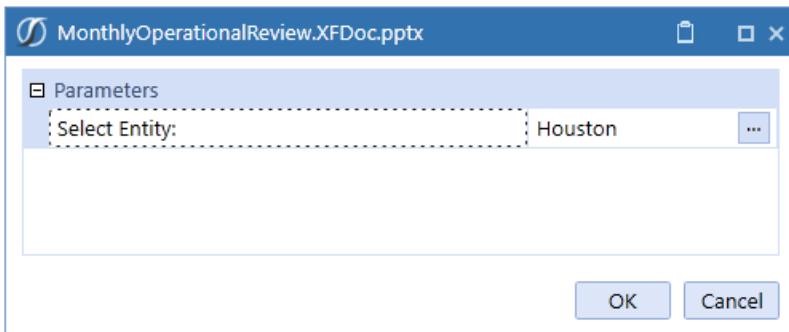
**NOTE:** See [Extensible Document Settings](#) for more details on XFCe11 syntax.

You can also copy and paste XFCe11 syntax from a Cube View's **Cell POV**

**Information** dialog box into a text file. See Cell POV Information in "Using OnePlace Cube Views" on page 962 for more details on this feature.

3. A substitution variable calls the OneStream application's global time, which updates with the Global Time Period is currently set in the application.
4. Additional format settings can be included to control options such as number formatting or scaling.

When the document is processed, it prompts to select an entity.



It then displays the updated data for the selected entity.

□ Houston results in thousands:  
□ Total Operating Income: 29,624

In this example, Houston has now replaced the parameter, and 29,624 is the XFCell function result. Data refreshes and updates when you select a different entity.

□ South Houston results in thousands:  
□ Total Operating Income: 15,256

## Creating a Document in Microsoft Excel

The example below is a summary balance sheet built in Excel. It shows how to:

- Use a parameter
- Use the XFGetCellVolatile retrieve function.
- Use the Excel IF function.
- Incorporate an Excel chart derived from the functions in an Excel spreadsheet.

Balance Sheet in Thousands			
	1	2	3
	2011Q1	2010Q1	Variance
Total Current Assets	#REFRESH	#REFRESH	#VALUE!
Net PP&E	#REFRESH	#REFRESH	#VALUE!
Total Other Assets	#REFRESH	#REFRESH	#VALUE!
Total Assets	#REFRESH	#REFRESH	#VALUE!
Total Liabilities	#REFRESH	#REFRESH	#VALUE!
Total Stockholders Equity	#REFRESH	#REFRESH	#VALUE!
Total Liabilities & Equity	#REFRESH	#REFRESH	#VALUE!

In this example:

1. A parameter named `!GetCellEntity!` was created in order to allow the user to select a specific entity on which to base the data for this spreadsheet. When the extensible document processes at run-time, it prompts to select an entity.
2. The `XFGGetCellVolatile` function was used to retrieve specific data from OneStream and update the Excel chart once the data is refreshed. Excel requires a volatile function for proper refreshing when using charts that reference calculated cells. This `XFGGetCellVolatile` formula derives from the `!GetCellEntity!` parameter and display updated data when selecting an entity at run-time.

**NOTE:** See Retrieve Functions in "Navigating the Excel Add-In" on page 1034 for more details.

3. The IF Excel function was also used which derives from the `XFGGetCellVolatile` function. This data updates when the spreadsheet launches from OneStream and the data refreshes. An example of the IF formula is as follows:

```
=IF((F6=0),"", (D6-F6)/F6)
```

	2011Q1	2010Q1	Variance
Total Current Assets	#REFRESH	#REFRESH	#VALUE!
Net PP&E	#REFRESH	#REFRESH	#VALUE!
Total Other Assets	#REFRESH	#REFRESH	#VALUE!
<b>Total Assets</b>	<b>#REFRESH</b>	<b>#REFRESH</b>	<b>#VALUE!</b>
Total Liabilities	#REFRESH	#REFRESH	#VALUE!
Total Stockholders Equity	#REFRESH	#REFRESH	#VALUE!
<b>Total Liabilities &amp; Equity</b>	<b>#REFRESH</b>	<b>#REFRESH</b>	<b>#VALUE!</b>

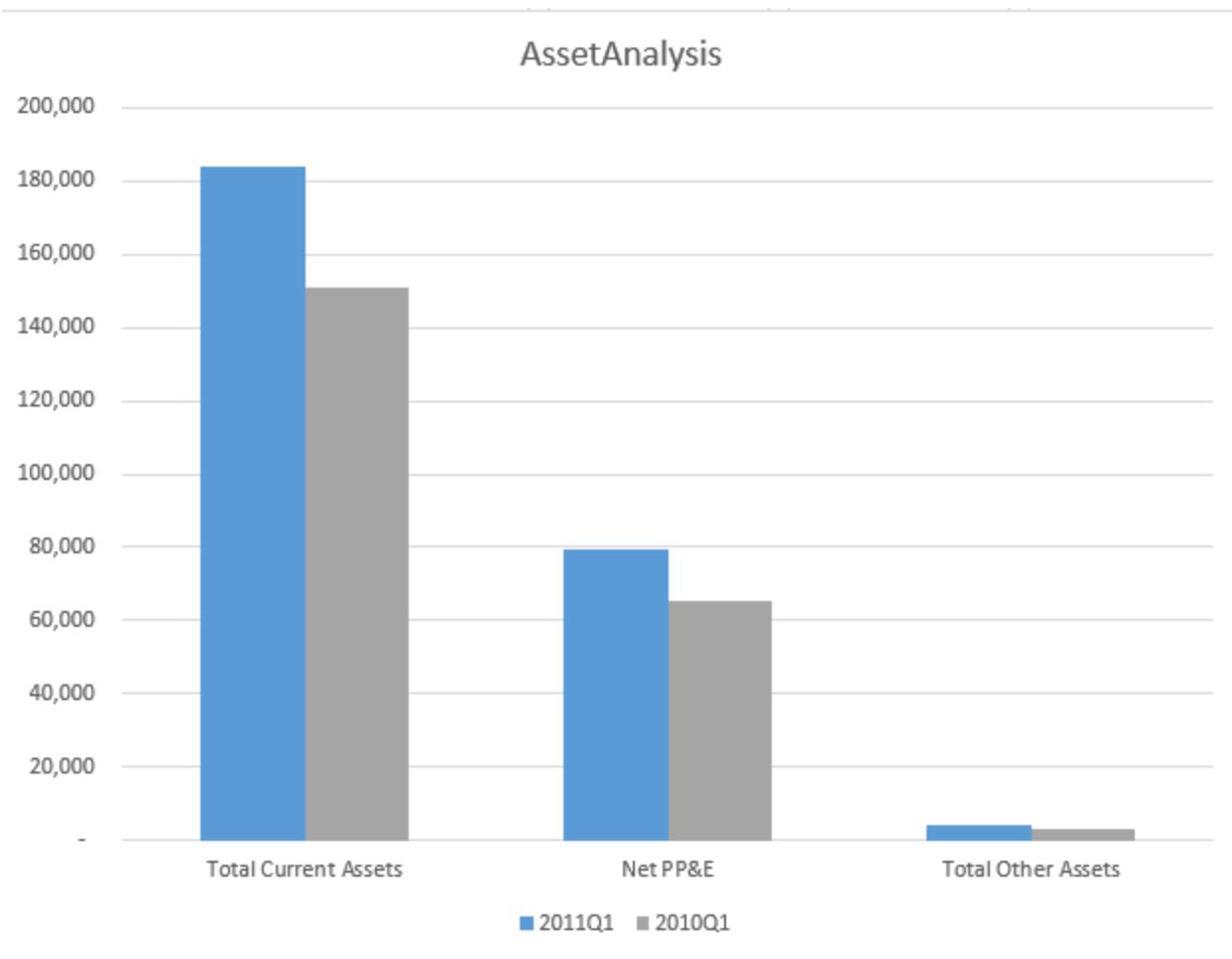


The chart displays four data points at the value of zero, representing the current state of the assets being analyzed.

4. An Excel chart inserted into this spreadsheet is driven by the data. The chart display the correct values whenever it is refreahed.

When the document processes, it prompts to select an entity and then runs the Excel spreadsheet. Log into the Excel Add-In and click **Refresh Data** to see updated values.

Balance Sheet in Thousands			
Houston			
	2011Q1	2010Q1	Variance
Total Current Assets	184,172	150,887	22.1%
Net PP&E	79,573	65,250	22.0%
Total Other Assets	3,867	3,171	22.0%
<b>Total Assets</b>	<b>267,613</b>	<b>219,308</b>	<b>22.0%</b>
Total Liabilities	196,919	207,433	(5.1%)
Total Stockholders Equity	70,694	11,876	495.3%
<b>Total Liabilities &amp; Equity</b>	<b>267,613</b>	<b>219,308</b>	<b>22.0%</b>



## Using Extensible Documents in OneStream

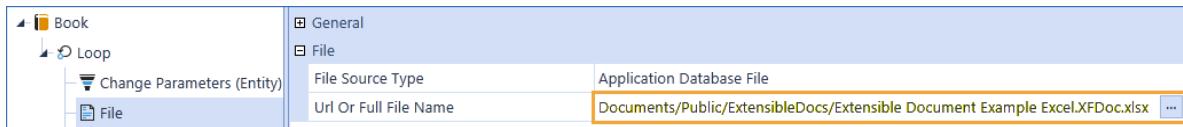
You can use extensible documents throughout OneStream. Each extensible document file is stored in the file share and can be kept private for specific users or placed into public folders for other users to see and use.

### Extensible Documents in Report Books

Add extensible documents to report books by selecting a File Book item and assigning the extensible document file.

## Presenting Data With Extensible Documents

---



The following shows the assignment result:

	2011Q1	2010Q1	Variance
Total Current Assets - Vol	184,172	150,887	22.1%
Net PP&E	79,573	65,250	22.0%
Total Other Assets	3,867	3,171	22.0%
<b>Total Assets</b>	<b>267,613</b>	<b>219,308</b>	<b>22.0%</b>

You must have access to the extensible document to run this book or the file displays blank pages. See "Presenting Data Using Report Books" on page 547 in Presentation for more details on this feature.

## Extensible Documents in Dashboards

View extensible documents in dashboards by assigning them to a File Viewer dashboard component that opens and processes the extensible document at run-time. You can also view extensible documents when assigned to a book that is assigned to a Book Viewer dashboard component. See Dashboards in "Presenting Data With Books, Cube Views and Dashboards" on page 547 for more details on dashboard components.

**NOTE:** If a book contains an extensible Excel document using the XFGetCell function, you do not need to login to the Add-In to see updated values.

## Extensible Documents in Data Management

Run extensible documents using a data management sequence by selecting the Export File step type and assigning the specific extensible document file to it. The extensible document exports to the OneStream file share when this step processes. See Data Management in "Application Tools" on page 727 for more details on this feature.

## **Extensible Document Limitations**

Extensible documents are powerful, but there are some limitations when using different file types.

Extensible documents cannot reference themselves.

### **Text File Limitations**

- Does not support images.

### **Microsoft Word Limitations**

- Word charts cannot be converted to PDF.

### **Microsoft PowerPoint Limitations**

- PowerPoint cannot be converted to PDF.

### **Microsoft Excel Limitations**

- The center across columns function is not supported.
- Excel Spark lines is not supported.
- Word wrap is not supported.
- 3D charts are not supported.
- Modifications to colors of individual chart items in a series does not carry through to a PDF.
- Fitting columns to a page is not supported.
- Repeating columns and rows is not supported.
- Cell comments do not display on a PDF after processing.
- Background images do not display after processing in Excel or PDF.
- Shapes do not display after processing in Excel or PDF.
- Smart Art does not display after processing in Excel or PDF.
- Macros are not supported.

The following Excel functions are not supported in extensible documents.

- ACCRINT
- ACCRINTM
- AGGREGATE
- AMORDEGRC
- AMORLINC
- BAHTTEXT
- BETA.DIST
- BETA.INV
- BINOM.DIST
- BINOM.DIST.RANGE
- BINOM.INV
- BINOMDIST
- CHISQ.DIST
- CHISQ.DIST.RT
- CHISQ.INV
- CHISQ.INV.RT
- CHISQ.TEST
- CONFIDENCE.T
- COUPDAYBS
- COUPDAYS
- COUPDAYSNC
- COUPNCD
- COUPNUM

- COUPPCD
- COVARIANCE.P
- COVARIANCE.S
- CRITBINOM
- CUBEKIPIMEMBER
- CUBEMEMBER
- CUBEMEMBERPROPERTY
- CUBERANKEDMEMBER
- CUBESET
- CUBESETCOUNT
- CUBEVALUE
- DDB
- DEVSQ
- DISC
- DURATION
- EXPON.DIST
- EXPONDIST
- F.DIST
- F.DIST.RT
- F.INV
- F.INV.RT
- F.TEST
- FACTDOUBLE

- FILTERXML
- FISHER
- FISHERINV
- FORECAST
- FVSCHEDULE
- GAMMA
- GAMMA.DIST
- GAMMA.INV
- GAUSS
- GCD
- GETPIVOTDATA
- GROWTH
- HARMEAN
- HYPGEOM.DIST
- HYPGEOMDIST
- INTRATE
- ISPMT
- KURT
- LCM
- LIMEST
- LOGEST
- MDURATION
- MULTINOMINAL

- NEGBINOM.DIST
- NEGBINOMDIST
- ODDFPRICE
- OFFYIELD
- ODDLPRICE
- ODDLYIELD
- PERCENTILE.EXC
- PERCENTILE.ING
- PERCENTILERANK.EXC
- PERCENTILERANK.INC
- PERMUT
- PERMUTATIONA
- PHI
- POISSON
- POISSON.DIST
- PRICE
- PRICEDISC
- PRICEMAT
- PROB
- QUARTILE.EXC
- QUARTILE.INC
- RANK.AVG
- RANK.EQ

- RECEIVED
- ROMAN
- RSQ
- RTD
- SERIESSUM
- SKEW.P
- SLN
- STANDARDIZE
- STDEVPA
- STEYK
- SYN
- T.DIST
- T.DIST.2T
- T.DIST.RT
- T.INV
- T.INV.2T
- T.TEST
- TBILLEQ
- TBILLPRICE
- TBILLYIELD
- TREND
- TRIMMEAN
- UNICHAR

- VARA
- VARPA
- VDB
- WEBSERVICE
- WEIBULL
- WEIBULL.DIST
- XIRR
- YIELD
- YIELDDISC
- YIELDMAT
- Z.TEST
- ZTEST

# Implementing Security

This section describes the four-pronged approach to managing security, which consists of:

- Workflow security
- Entity security
- Account security
- Security roles

Security can be implemented on accounts or dimensions allowing you to control who can review specific dimension members. Security is determined through users and groups, with users given specific roles to determine what data can be accessed or edited.

## Application Security

A four-prong approach to application security consists of:

- Workflow security
- Entity security
- Account security
- Security roles

Once you identify entities and assign them to workflow profiles, data loaders and certifiers can be determined for each entity. Data loaders load data into the system, therefore they need read/write access to entities. Data certifiers review and sign off on the loaded data, so they need read access to entities. Security can also be done on the account or any other dimension to control who can review specific dimension members.

Security is determined through users and groups. Users are given specific roles to determine what data is accessed or edited. For example, if a user is given the ModifyData role in an application, he/she will have write-access to any data in it. Users are also put into security groups. Groups can support native groups, exclusion groups, or groups of groups. For example, a user can be put into an entity's read/write data group for read/write access to the entity's data.

Every object has access and maintenance security rights, with the exception of Task Scheduler. Access allows the security group to view the object, while maintenance allows the groups to edit the definition of the object. This system applies to most objects, such as cube views, dashboards, transformation rules, and workflow profiles.

# Security Best Practices

## Object Security

There needs to be different levels of access for object types and groups of objects, such as Cube Views, Dashboards or Business Rules. Application and System Roles can be granted to User Groups which create subject area administrators and by giving certain rights to a group, such as ManageCubeViews, pseudo administrators are created for these actions. This provides the most power for a specific object type.

A Maintenance Group is the middle level of power for an object at the group level. For example, a Maintenance Group assigned to a specific Entity Transformation Rule Group allows the assigned users to create, edit, and delete rules within that Transformation Rule Group.

An Access Group is the lowest level of power for an object at the group level. This means the object can be used, but its definition cannot be edited.

## Confirmation Rules

Confirmation Rule Groups are assigned to Confirmation Rule Profiles which are then assigned to Workflow Profiles. The run time access to these Confirmation Rules depends on to which Workflow Profile they have been assigned. If a user has Workflow Execution Access, he/she will be able to execute them.

The best way to control Confirmation Rules is to set Access to Everyone and Maintenance to Administrators for both Confirmation Rule Groups and Profiles.

## Certification Questions

Certification Question Groups are assigned to Certification Question Profiles which are then assigned to Workflow Profiles. The run time access to these Certification Questions depends on to which Workflow Profile they have been assigned. If users have Workflow Execution Access, they will be able to execute them.

The best way to control Certification Questions is to set Access to Everyone and Maintenance to Administrators for both Certification Question Groups and Profiles.

## Data Sources

Data Sources are assigned to Workflow Profiles. The run time access to these Data Sources depends on to which Workflow Profile they have been assigned. If a user has Workflow Execution Access, he/she will be able to execute them.

The best way to control Data Sources is to have the ManageDataSources Application role, and no security settings at the object level.

## Transformation Rules

Transformation Rule Groups are assigned to Transformation Rule Profiles which are then assigned to Workflow Profiles. In this case, an appropriate user group needs to be assigned to Access and Maintenance because users will be able to right-click on an Import Workflow Profile and view/edit their Transformation Rules. The user groups should include the users assigned to execute the Workflow Profiles to which the Transformation Rule Profile has been assigned.

The best way to control Transformation Rules is to set Access to Everyone and Maintenance to Administrators for most core, shared, or corporate Transformation Rule Groups. For some specific Transformation Rule Groups, such as an Account Transformation Rule Group that applies to a specific location, assign the appropriate user groups to Access and Maintenance. Block access to the Maintenance screen for anyone except administrators because this could potentially allow users more access than they need.

## Form and Journal Templates

Form or Journal Groups are assigned to Form/Journal Profiles which are then assigned to Workflow Profiles. The run time access to these Forms or Journals depends on to which Workflow Profile they have been assigned. If a user has Workflow Execution Access, he/she will be able to execute them.

The best way to control Form and Journal Templates is to set Access to Everyone and Maintenance to Administrators for both Form/Journal Groups and Profiles.

## Cube Views

The best way to control Cube View Groups is to set Access to Everyone and Maintenance to Administrators and anyone else building a Cube View. To keep the assignment of Cube View Groups to multiple Cube View Profiles flexible, the Cube View Groups need to remain smaller in size. For Cube View Profiles, set Access to anyone who will need to see these Cube Views in OnePlace, Excel, or assign them to Workflow Profiles, Forms, or Dashboards. Set Maintenance to anyone who needs to change the assignment of the Cube View Groups to Cube View Profiles.

OneStream recommends setting the Can Modify Data, Can Calculate, Can Translate, and Can Consolidate properties to False. This can be pre-set for all new Cube Views by creating an example or Cube View Template which can be copied to create new ones. Some examples of when this will not be needed is if the Cube Views are going to be read by administrators only, the Cube Views will be used as a data entry form or are only going to be visible in a formatted report or chart.

## System and Application Dashboards

When assigning Dashboard Groups to Profiles, the Visibility is extremely important. For example, if a user has access to a Dashboard Profile in OnePlace, but not to a certain Dashboard Group in that Profile, the user will not be able see the Dashboards in that group. Also, if a user has access to the Dashboard Groups, but not the Profiles, he/she will not be able to see the Dashboards in OnePlace. If a Cube View is assigned to an Application Dashboard, and the user only has access to the Cube View, he/she will not see the Dashboard. Lastly, if a Dashboard is pointing to an Entity, Scenario, or Cube Data to which the user does not have access, he/she will see one of the following: NoAccess in Data Explorer for the cells the user cannot see, a blank cell in the Data Explorer Report, or No Data Series if he/she is viewing a chart.

The best way to control Dashboard Groups is to set Access to Everyone and Maintenance to Administrators and anyone else building a Dashboard. In order to keep the assignment of Dashboard Groups to multiple Dashboard Profiles flexible, the Dashboard Groups need to remain smaller in size. When assigning Maintenance for Dashboard Profiles, give access to anyone who needs to see the Dashboard in OnePlace, assign it to a Workflow Profile, or change the assignments of Dashboard Groups to Dashboard Profiles.

Use multiple Dashboard Maintenance Units in order to keep them a reasonable size making it easier to manage multiple objects and access. Dashboard Parameters can also be used across all Dashboards and do not need to be copied across all Maintenance Units. Security has no bearing on the use of Parameters.

# Workflow Security

Security groups for Workflow Execution, which is the ability to process a Workflow for a specific Workflow Profile, Certification Signoff and the separate ability to Process, Approve and Post Journals, exist for all Workflow Profiles. In certain cases, the user simply needs the Access and Workflow Execution Group Access to run Workflow. For example, the user does not need Access or Maintenance Group access to Data Sources or Transformation Rules in order to run through the Import Workflow.

In some cases, having access to certain objects is necessary along with Workflow Execution Group Membership. The Manage App Role has to do with creating, reading, updating, and deleting Journal and Form Templates (metadata) themselves, not just instances of these objects at run time. It is expected that 90% of Workflow users will not have any of the Application Roles, but their access will be controlled by the Access Group for those Journal/Form Template Groups and Profiles. Workflow users also need Workflow Execution Group access in order to perform import, forms, and journal actions. The user does not have to be in the Manage Application role to create a Journal or enter data in the Form. Workflow security governs access to the forms. If the user is in the ManageJournalTemplates Application Role group, he/she can create any Journal needed for the Workflow Profiles to which they have proper execution access.

Users need to have at least Access Group privileges to the Cube Root Workflow Profile node to edit Workflow Profiles with having the ManageWorkflowProfiles role. Otherwise they will not be able to see any Workflow Profiles under the Cube Root Workflow Profile.

The order to follow when assigning access to Workflow Profiles and data is to first assign Read and Read/Write Groups to the Entities involved. Next, create an Access Group, Data Group, and Approver Group for each Workflow Profile and include the appropriate Entity groups.

## Import

First, determine whether the users can load data to the Workflow for the assigned Entities and then determine whether they load both GL (BS and PL) and Supplemental data, or one or the other. Next, decide if the users for the assigned Entities can certify the loaded data as part of the Workflow.

## Forms

First, determine whether the users can manually input data into a form and certify it as part of the Workflow for the assigned Entities.

# Adjustments

First, determine whether the users can manually input data into a journal and certify it as part of the Workflow for the assigned Entities.

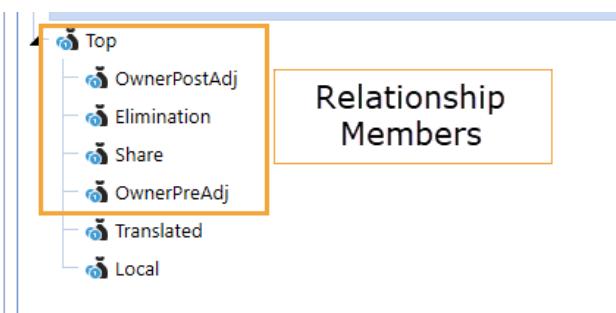
## Entity Security

Entity Security controls the overall read/write access to Entity data and controls whether Cube Security should be used. When creating Entity security groups for the Read Data Group and the Read/Write Data Group, the groups should be named in a logical convention such as XXXX\_View or XXXX\_Mod. The Entity Read/Write Data Group should be designed first because it is needed for data loading in Workflows. The Workflow Execution Security Group should be assigned to all the Entities' Read/Write Security Group for the Workflow to gain loading access to the Entities.

When setting up View Security Groups for Entities, first consider how users need to view their data whether it is by segment or region. Determine whether it makes more sense to have one Entity View Group per Entity, or to create one Entity View Group per segment and apply one Entity View Security Group to many Entities' Read Data Group. All the Entities' View Groups below the Parent must be assigned to the Parent Level Entity View Group in order to gain access to data at the Parent Level Entity and View Entities below it. Try to minimize the amount of View Entity Security Groups where possible.

## Relationship Security

You can change the security model to allow who controls viewing or modifying the relationship members in the consolidation dimension.



For the Use Parent for Relationship Consolidation dimension Members functionality:

## Implementing Security

---

- If set to False, the user's entity rights control their rights to all members of the consolidation dimension. This is the default security model.
- If set to True, the user's rights to the relationship members of the consolidation dimension are determined by their rights to the current entity's immediate parent.

Users have either read or read/write access to view or modify their entity so they can see their entire entity from all of the relationship members.

You can allow the relationship security portion of the consolidation dimension to be controlled by the access to the parent entity in the immediate hierarchy.

**NOTE:** This is strictly a parent and child relationship.

Set this feature to True when:

- A user's rights to the non-relationship consolidation dimension members (local and translated) are determined by the user's rights to the entity itself.
- A user's rights to the relationship consolidation dimension members is determined by the user's rights to the immediate parent entity.

## Change the Relationship Security

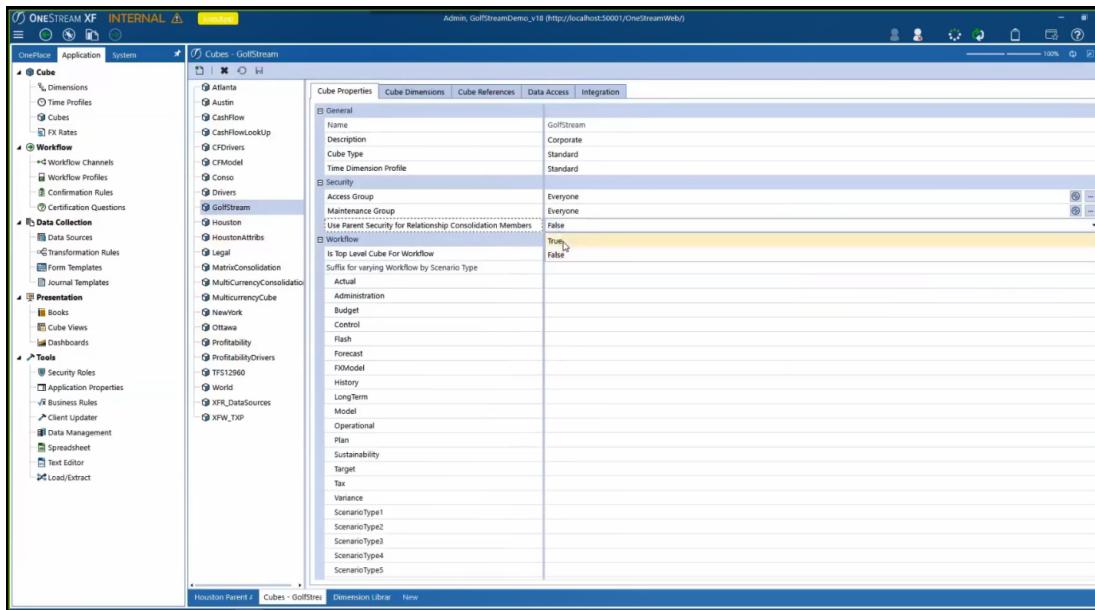
The screenshot shows the OneStream XF Data Explorer interface. The left sidebar contains navigation links like OnePlace, Application, Workflow, and Cube-Views. The main area displays a table titled "NodeSecurity\_HoustonCube". The table has columns: Local, Translated, OwnerPreAdj, Share, Elimination, OwnerPostAdj, and Top. The data in the table represents security settings for various entities based on their parentage. For example, rows show rights for "GolfStream - Parent= NA Clubs - Entity= US Clubs", "GolfStream - Parent= US Clubs - Entity= Calgary", and "GolfStream - Parent= US Clubs - Entity= Augusta". The "Share" column often contains values like 5,000.00 or 11,000.00, while other columns like "OwnerPreAdj" and "OwnerPostAdj" are mostly "No Access".

	Local	Translated	OwnerPreAdj	Share	Elimination	OwnerPostAdj	Top
Cube= GolfStream - Parent= NA Clubs - Entity= US Clubs	5,000.00	5,000.00		5,000.00			5,000.00
Cube= GolfStream - Parent= US Clubs - Entity= Calgary	No Access	No Access	No Access	No Access	No Access	No Access	No Access
Cube= GolfStream - Parent= US Clubs - Entity= Augusta	No Access	No Access	No Access	No Access	No Access	No Access	No Access
Cube= GolfStream - Parent= US Clubs - Entity= Carlsbad	No Access	No Access	No Access	No Access	No Access	No Access	No Access
Cube= GolfStream - Parent= US Clubs - Entity= Houston	11,000.00	11,000.00		11,000.00	-6,000.00		5,000.00
Cube= GolfStream - Parent= Houston - Entity= Houston Heights	14,000.00	14,000.00		14,000.00	-3,000.00		11,000.00
Cube= GolfStream - Parent= Houston - Entity= South Houston	No Access	No Access	No Access	No Access	No Access	No Access	No Access
Cube= Houston - Parent= Texas - Entity= Houston	11,000.00	11,000.00		11,000.00			11,000.00
Cube= Houston - Parent= Houston - Entity= Houston Heights	14,000.00	14,000.00		14,000.00	-3,000.00		11,000.00
Cube= Houston - Parent= Houston - Entity= South Houston	No Access	No Access	No Access	No Access	No Access	No Access	No Access
Cube= GolfStream - Parent= US - Entity= Texas	No Access	No Access	No Access	No Access	No Access	No Access	No Access
Cube= GolfStream - Parent= Texas - Entity= Austin	No Access	No Access	No Access	No Access	No Access	No Access	No Access
Cube= GolfStream - Parent= Texas - Entity= Dallas	No Access	No Access	No Access	No Access	No Access	No Access	No Access
Cube= GolfStream - Parent= Texas - Entity= Houston	11,000.00	11,000.00		11,000.00			11,000.00
Cube= GolfStream - Parent= Houston - Entity= Houston Heights	14,000.00	14,000.00		14,000.00	-3,000.00		11,000.00
Cube= GolfStream - Parent= Houston - Entity= South Houston	No Access	No Access	No Access	No Access	No Access	No Access	No Access

## Implementing Security

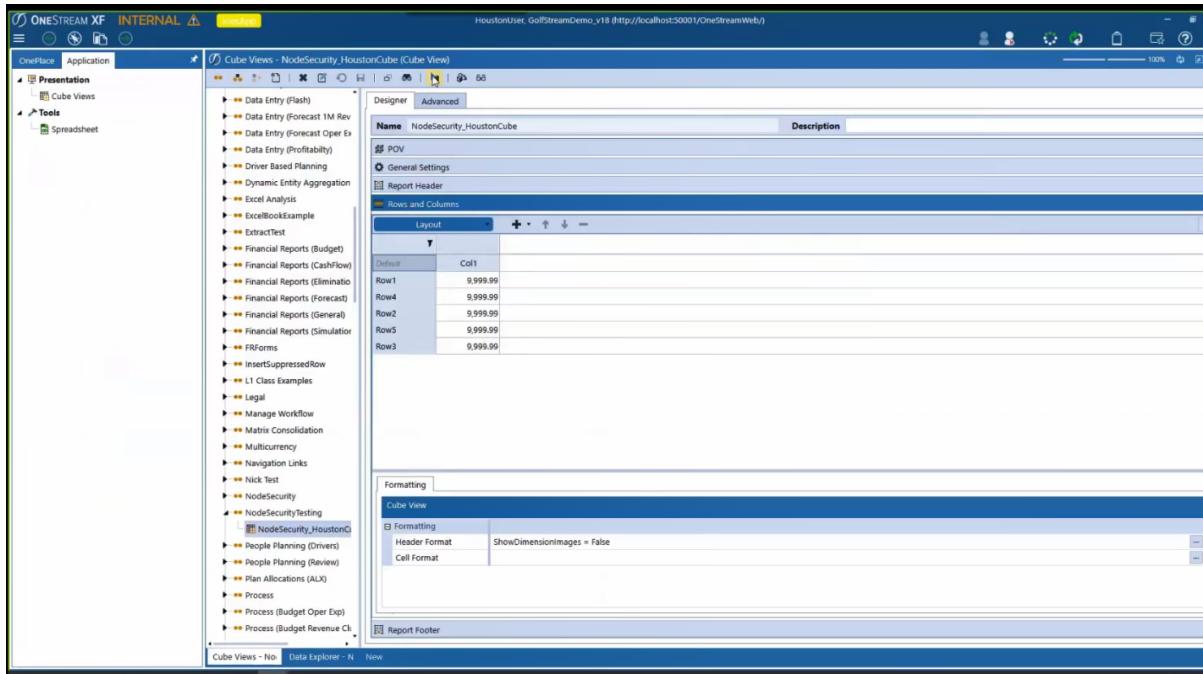
---

1. From the Application tab, under Cube, click **Cubes**.
2. Select a Cube.
3. In the Cube Properties tab, in **Use Parent Security for Relationship Consolidation Dimension Members**, select **True**.



4. Click **Save** to save the cube properties. You have now changed the security model for the read or write relationship for the consolidation dimension members to be based on the security rights to the immediate parent entity.
5. Click **Refresh** to refresh the application.
6. In Cube Views, go to the Designer tab and click **Open Data Explorer**.

## Implementing Security



All of the security is changed.

	Local	Translated	OwnerPreAdj	Share	Elimination	OwnerPostAdj	Top
Cube= GolfStream - Parent= NA Clubs - Entity= US Clubs	5,000.00	5,000.00	No Access	No Access	No Access	No Access	No Access
Cube= GolfStream - Parent= US Clubs - Entity= Calgary			No Access	No Access			
Cube= GolfStream - Parent= US Clubs - Entity= Augusta			No Access	No Access			
Cube= GolfStream - Parent= US Clubs - Entity= Carlsbad			No Access	No Access			
Cube= GolfStream - Parent= US Clubs - Entity= Houston	11,000.00	11,000.00		11,000.00	-6,000.00		5,000.00
Cube= GolfStream - Parent= Houston - Entity= Houston Heights	14,000.00	14,000.00		14,000.00	-3,000.00		11,000.00
Cube= GolfStream - Parent= Houston - Entity= South Houston			No Access	No Access			
Cube= Houston - Parent= Texas - Entity= Houston	11,000.00	11,000.00	No Access	No Access	No Access	No Access	No Access
Cube= Houston - Parent= Houston - Entity= Houston Heights	14,000.00	14,000.00		14,000.00	-3,000.00		11,000.00
Cube= Houston - Parent= Houston - Entity= South Houston			No Access	No Access			
Cube= GolfStream - Parent= US - Entity= Texas			No Access	No Access			
Cube= GolfStream - Parent= Texas - Entity= Austin	No Access	No Access	No Access	No Access	No Access	No Access	No Access
Cube= GolfStream - Parent= Texas - Entity= Dallas	No Access	No Access	No Access	No Access	No Access	No Access	No Access
Cube= GolfStream - Parent= Texas - Entity= Houston	11,000.00	11,000.00	No Access	No Access	No Access	No Access	No Access
Cube= GolfStream - Parent= Houston - Entity= Houston Heights	14,000.00	14,000.00		14,000.00	-3,000.00		11,000.00
Cube= GolfStream - Parent= Houston - Entity= South Houston	No Access	No Access					

Notice the following changes to security:

- In the first screenshot, all the consolidation dimension members are available across Houston.
- When you switch on the new security, in the previous screenshot, the user has rights to USClubs then looking at the Houston member there, all the consolidation dimension members are available.
- Looking at the last group, the user has no rights to Texas, based on the NoAccess to Local and Translated. In Houston you can see the relationship members changed to NoAccess.
- The reason why USClubs turns to NoAccess is because the user apparently has no rights to the parent NAClubs, which is not displayed on the report.

### Data Security

Data security controls the overall read/write access to analytic models. There are several steps to see if you have access to data. First, it checks if you have the ModifyData role in the application, otherwise you do not have write access to data. Next, ensures you have either the access group or maintenance group for a cube, or you do not have access to any of the cube's data. It will then check the scenario and Entity's Read Data Group and Read Write Data Group to ensure access. Finally checks the Data Cell Access Security at the cube level. If you have access to the data and use Excel or any other method to import data, it ensures the entity being written to belongs to a workflow profile with an active forms channel that is not locked or certified. If any of these steps result in no access, the process stops.

- There are several ways to guarantee data is secure. Throughout the dimensions, different security groups are available, and an administrator can decide what users belong to each group.
- The scenario dimension has both Read Data Group and Read and Write Data Group.
- The entity dimension uses both security groups from the scenario and adds display member group and Use Cube Data Access.
- The display member group only refers to the member display access level, not the data access level. Use Cube Data Access is used if slice filters are being applied at the cube level to apply additional layers of security for specified security groups.
- The account, flow, and all user-defined dimensions also use display member group.

At the cube level, more security is put in place through groups, member filters, and complex layered security. A cube slice filters a data entry form to the right member set, such as choosing the cost center in which a user can enter data. For slice filters to take effect, the user must first be granted access to the cube, scenario, and entity. Slice security cannot increase access to data that was not administered first through Users and Groups. An administrator can also lock down more than one dimension by using cube data access and can control user visibility by only giving access to certain accounts. Finally, data cell conditional input and data management access security can be used. These are not security settings but can still control how a dimension can be used for input and how a cube is modified.

### **Application and System Security**

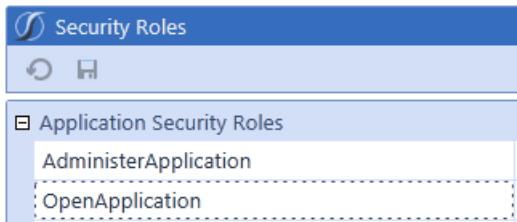
dimensions, cubes, business rules, data management, and File Explorer each have an access and maintenance group. OneStream recommends limiting access to these groups and assigning the screens to administrators. It is also recommended to give the ManageData role to administrators for data management and to give ManageTaskScheduler and TaskScheduler roles in Task Scheduler. The ManageFileShare role should also just be given to administrators. Anyone granted access to a folder in File Explorer has access to every file and folder through OnePlace. For FX rates, limit access by giving administrators the ManageFXRates role and selecting a set of power users.

## **Security Configurations**

### **Restrict Users to an Application**

Setting up access to an application can be done within the application security roles. When an application is first created, the OpenApplication defaults to Everyone. If specific people have access to log into OneStream, but only need access to specific applications, then security groups can be assigned to the OpenApplication role. Once the security group is created, users can be assigned to it. This can be done by performing the following steps:

1. Go to **System > Administration > Security**.
2. Create a new security group.
3. Assign all users who should have access to the application to the new security group.
4. Refresh the application in order for this new security group to appear in all drop down menus.
5. Go to **Application > Tools > Security Roles > OpenApplication**.



6. Click the drop-down and select the new security group.
7. Click **Save**.

## Restrict Data Input by Origin

There may be times where data should be loaded through the Import origin but should not be loaded using the Forms origin. This can be handled in the data cell conditional input. For example, users may be able to load trial balance data through the Import origin, but other users submit statistical data through the Forms origin. The data cell conditional input ensures the statistical data does not overwrite the Trial Balance data in Actual. Perform the following steps to do this:

1. Go to **Application > Cube > Cubes > Data Access**.
2. Go to **Data Cell Conditional Input**.
3. Click to create a new line.
4. Click , or double click on the cell to make changes to the member filter. Add the dimension intersection to restrict data loading. In this case it restricts users from loading to the Trial Balance account through the Forms origin.
5. In the In Filter field, choose a behavior and choose the Read Only Access level.

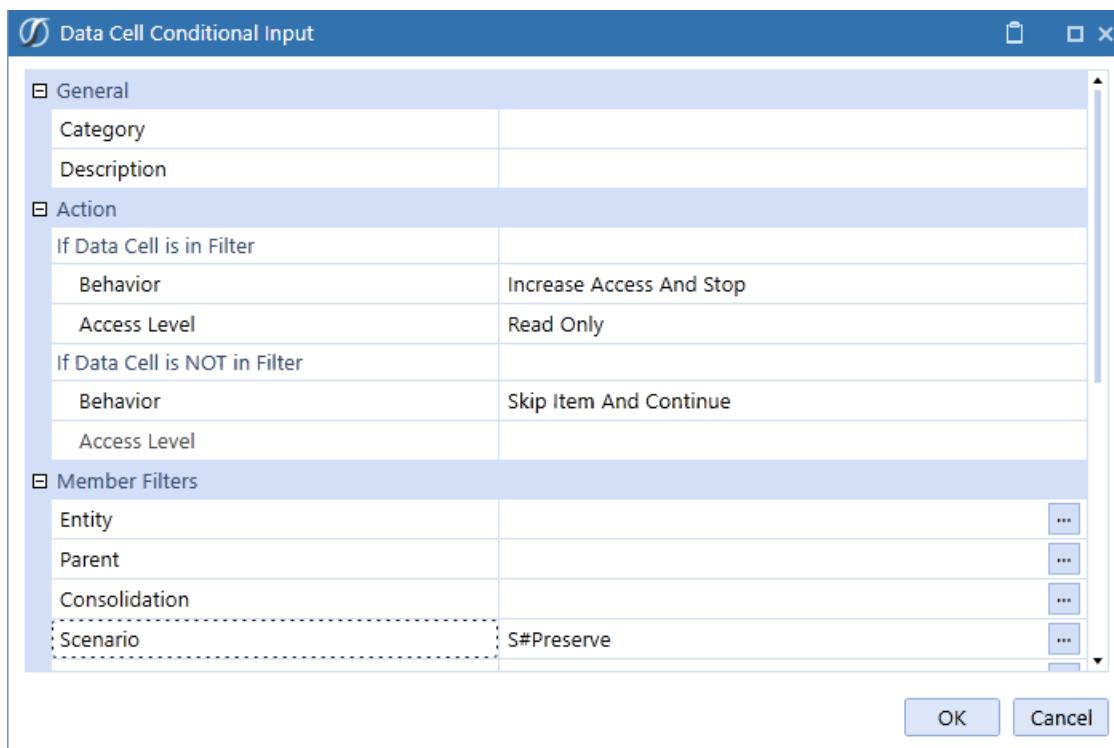
Data Cell Conditional Input			
		Category	Description
		Member Filter	In Filter
		A#TRIAL_BALANCE.children, O#FORMS	Apply Access And Continue, Read Only

# Omitting Data Cell Conditional Input by Scenario

Data cell conditional input restricts access to certain intersections or slices of the cube. This behavior might not be desired for all scenarios, time periods or other elements, so this is a simple way to omit these rules for specific elements. The following method is useful when there are many data cell conditional input rules not defined by a scenario at the time of creation. After these rules are created, however, a scenario becomes a factor for historical data.

For example, there may be many read-only intersections in Actual to which users should not load data. These rules are setup in the data cell conditional input section within Application|Cube|Cube|Data Access. However, for historical purposes, there may be data in these intersections that was there prior to the filters being applied. The scenario to which the data is being copied needs to allow access because the historical data in these intersections may need to be copied for analysis.

Create a new data cell conditional input rule for the entire scenario, set the behavior to Increase Access And Stop, set the Access Level to Read Only.



Position this new rule at the top of the Data Cell Conditional Input for the scenario.

Data Cell Conditional Input			
	Category	Description	Member Filter
		S#Preserve	In Filter Increase Access And Stop, Read Only
		A#TEST, U1#PROJECT, U2#COSTCENTER	Apply Access And Continue, Read Only
		A#TEST1, U1#PROJECT, U2#COSTCENTER	Apply Access And Continue, Read Only

The behavior option Increases Access And Stop is being used because if the current cell matches the filter, access is being increased and all subsequent data access rules are being ignored below. In this case, the Preserve Scenario has access to everything, and the subsequent Data Cell Conditional access rules are ignored or not applied for Preserve.

# System Security

System Security applies to the framework. There are some key assumptions around how the different roles or security groups work. First, it is important to understand the hierarchy of certain System Security Roles and Security Groups:

### System Security Role

The System Role, such as ManageSystemDashboards, means the user has a higher privilege and does not need to be in any Maintenance Group or Access Group to see, edit, or delete all objects of that type. Having the ManageSystemDashboards System Security Role means the user can create, edit and delete any System Dashboard, System Dashboard Group, or Profile.

### Maintenance Group

The Maintenance Group means users cannot only see an object, but can create new objects in Groups, edit, and delete them. Users do not need to be in the Access Group for an object if they are in the Maintenance Group. The Maintenance Group can also control the contents of Profiles.

### Access Group

The Access Group means users can see the object and read its contents.

# Managing a OneStream Environment

Management of all changes to the system are recommended to follow best practice procedures. Whether the changes derive from a OneStream software upgrade, or through regular application maintenance, all changes are recommended to be first deployed and tested in a development environment. There are additional benefits of making a recent copy of the production Application database, renaming it and using this as a base for these changes. Search for “Rename Application” in the Installation and Configuration Guide.

Prior to being deployed to a production environment, it is recommended to extract changes from the development environment and assess this deployment of changes in a separate test environment.

Deploying changes to a production environment should avoid times during high load and high application activity. Changes to these types of application artifacts especially should not be performed against a production environment experiencing heavy activity:

- Business Rules, whether they contain Global functions or not
- Confirmation Rules
- Metadata, especially when using member formulas

Applying changes like this while the production system is under a high level of activity may have a negative impact on servers and have the potential to cause running processes to produce an error.

Standard environments are recommended to schedule production changes during slow periods or non-work hours. Large environments should consider using the Pause functionality within the Environment tab to allow activity to wind down. These large environment managers should also consider the Marketplace solution, Process Blocker, which allows a pause of critical processes to perform maintenance on the system, without having to shut down the entire application. Process Blocker allows current tasks to be completed, while any new requests are queued, allowing the changes to be applied safely and effectively. Once these changes are in place, it is recommended to significantly limit the ability for users to make such changes during high volume.

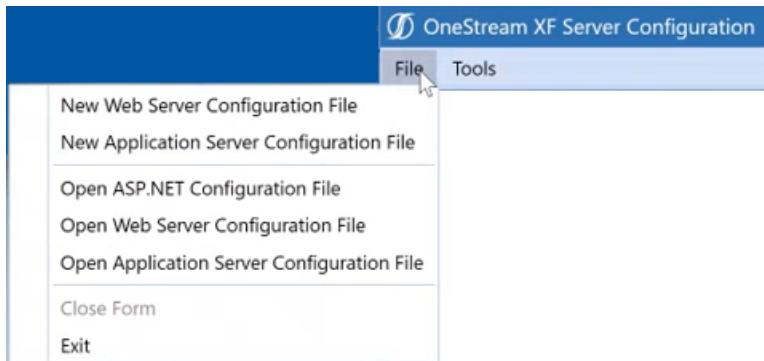
It is key that servers get a chance to recycle for good system memory health. IIS also has an Idle Time-Out setting for our OneStreamAppAppPool. This setting should be set to 0 since OneStream has other settings to recycle IIS. For active, global environments with Data Management Sequences regularly being executed, a recycle of IIS is recommended every 24 hours for these OneStream App Servers. Please discuss this situation with OneStream Support to find what is recommended, as each situation may vary.

## Disable Inactive Users

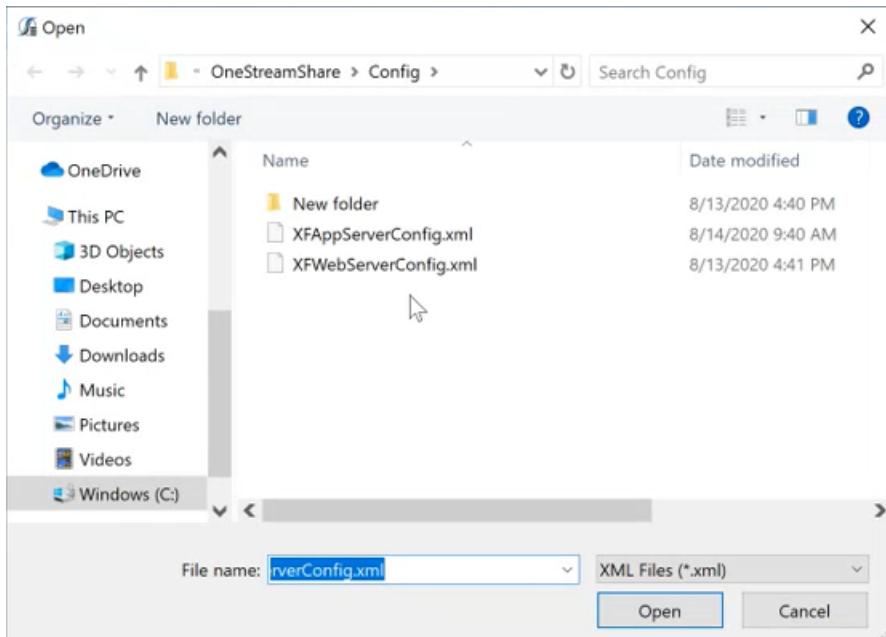
Disable inactive users gives you the ability to create an authorization policy to disable users after a specific amount of logon inactivity, keeping only active users in the system.

# Set the Inactivity Threshold

1. Click File > Open Application Server Configuration File.



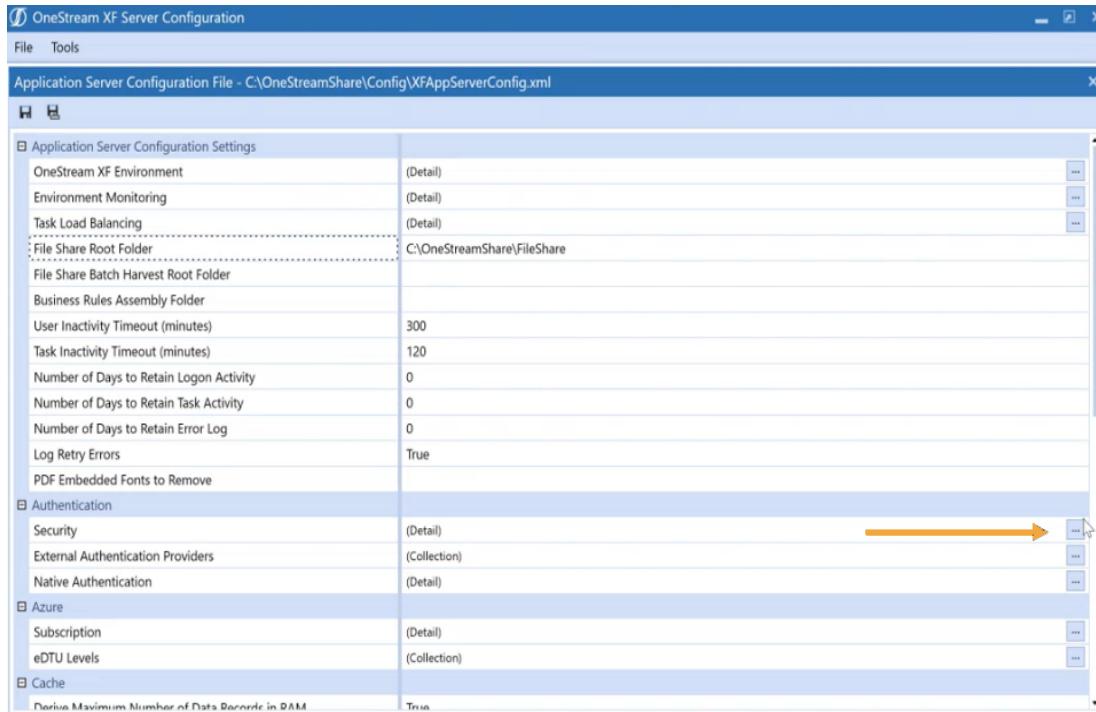
2. Open the configuration file.



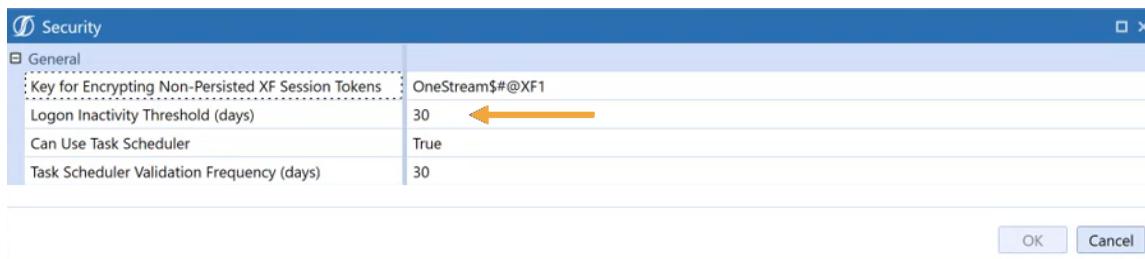
## Implementing Security

---

### 3. Click **Security**.



### 4. Set the Logon Inactivity Threshold (days) to the number of days of inactivity before the user can no longer access the system.

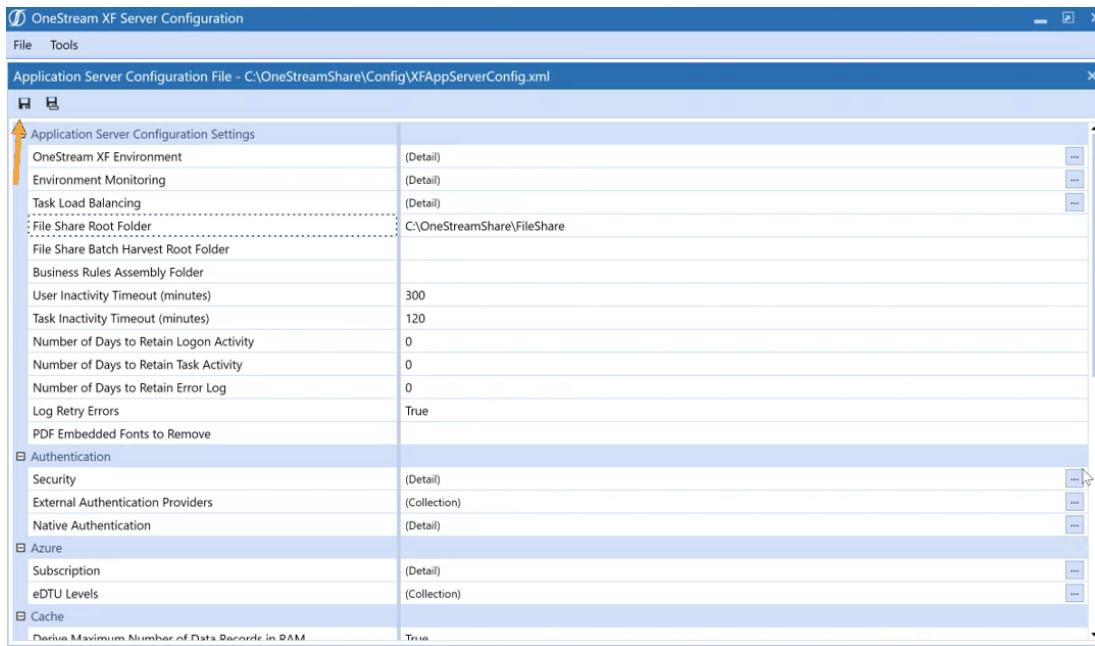


### 5. Click **OK**.

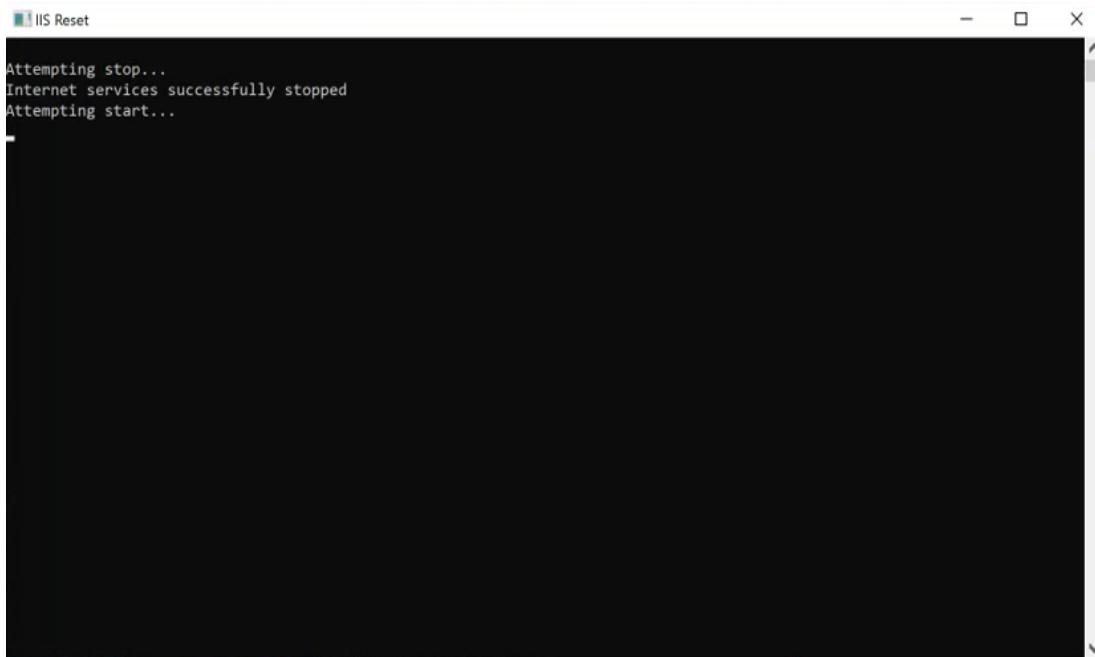
## Implementing Security

---

### 6. Click Save.



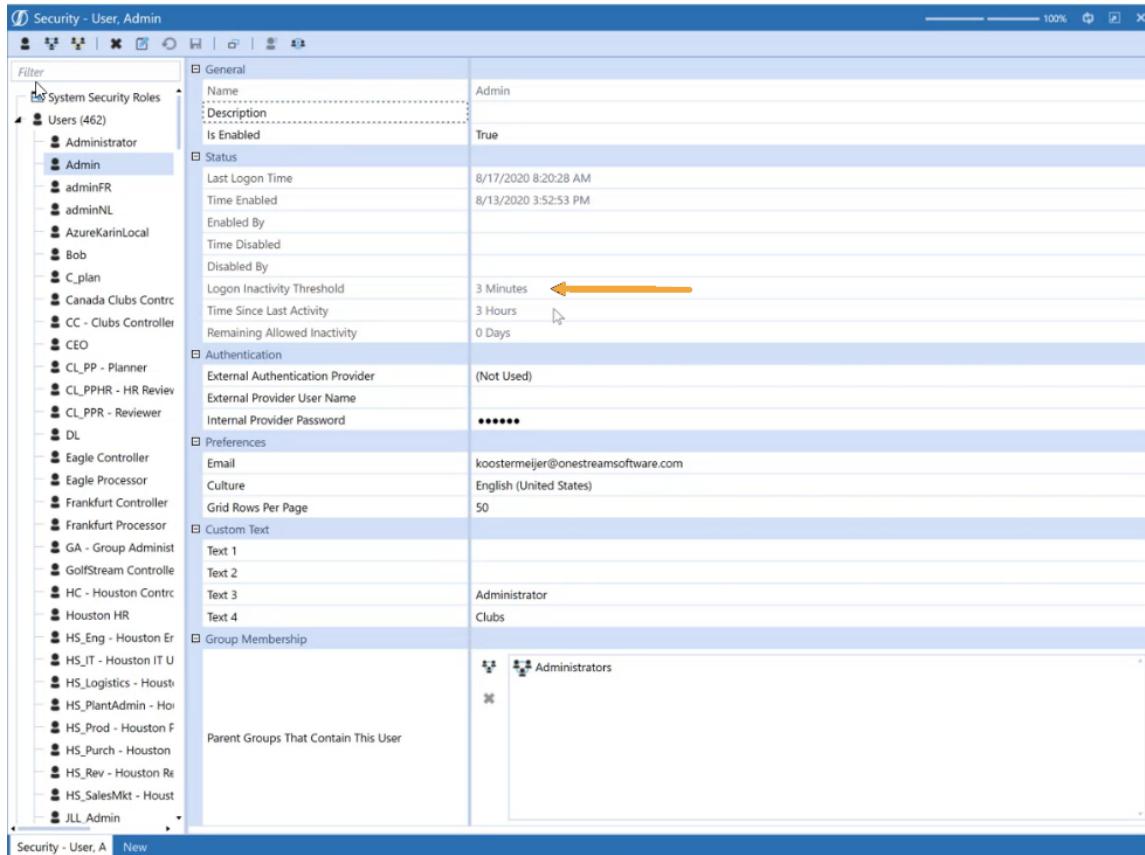
### 7. Reset IIS to recognize the changes.



# Review Settings

The Administrator is the only role that the inactivity threshold does not apply.

1. Click **System > Security**.
2. Click **Users** to review the threshold setting.



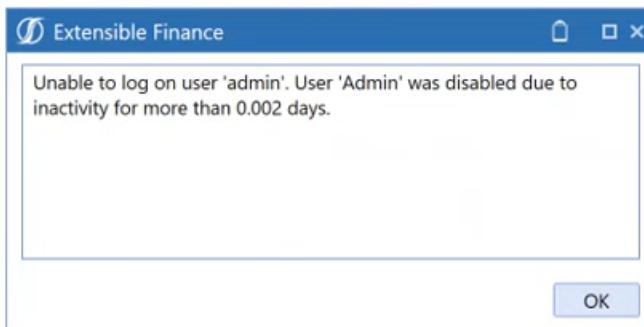
3. If the Remaining Allowed Inactivity is **0 Days**, that means the user no longer has access.

## Implementing Security

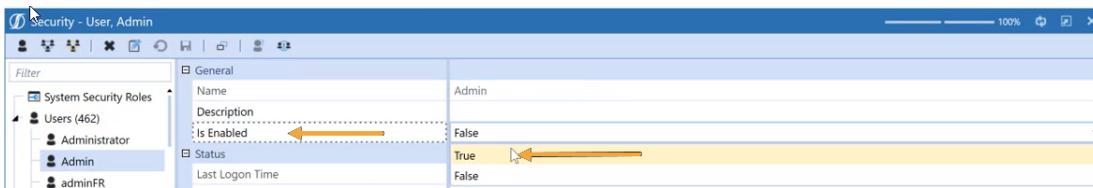
---

The screenshot shows the 'Security - User, Admin' window. On the left is a tree view under 'System Security Roles' with 'Users (462)' expanded, showing various users like 'Administrator', 'Admin', 'adminFR', etc. The main pane displays the 'General' tab for the user 'Admin'. The 'Status' section includes fields for 'Last Logon Time' (8/17/2020 8:20:28 AM), 'Time Enabled' (8/13/2020 3:52:53 PM), and 'Enabled By' (AzureKarinLocal). The 'Inactivity' section shows 'Logon Inactivity Threshold' (3 Minutes), 'Time Since Last Activity' (3 Hours), and 'Remaining Allowed Inactivity' (0 Days, highlighted with an orange arrow). The 'Authentication' section lists 'External Authentication Provider' (Not Used) and 'Internal Provider Password' (\*\*\*\*\*). The 'Custom Text' section contains 'Text 1' through 'Text 4' and 'Group Membership' which lists 'Administrators'. At the bottom, there's a note about 'Parent Groups That Contain This User'.

That user gets a message telling them they have been disabled if they log on.



4. To enable their access, click **System > Security**.



5. Select the user.
6. Click the **Is Enabled** drop-down.
7. Select **True**.
8. Click **Save**.

The Remaining Allowed Inactivity field updates allowing the user to log back in within the time frame specified.

## Business Rules

1. Click Application > Business Rules > Extensibility Rules.
2. The three new business rules are GetStatusByID, GetStatusByName, and LogoninactivityThresholdForAdmin.



## System Table

1. Go to System > Database > System Database> Tables.

2. There is a table name UserLogonStatus



## Business Rules

A Business Rule is a VB.Net Class meaning each Business Rule is an independent object encapsulating VB.Net code. A Business Rule can be a one-line call to write a log message, or it can be a full code library containing other custom VB.Net Classes, Methods and Properties.

This section provides a detailed explanation of the following:

- Platform Engines
- Business Rule structure and fundamentals
- Business Rule Classifications
- Specific Business Rule Types
- Business Rule organization
- OneStream Business Rule framework
- Best practices for Business Rule architecture
- Business Rule organization and referencing

**IMPORTANT:** Whether it be OneStream, Partner, or Customer, we strongly advise against any sensitive or confidential information be included in business rules.

See the API Overview Guide and API Details and Database Documentation for detailed Business Rule engine background, an API guide, and information on each database related.. .

## Platform Engines

The platform is comprised of multiple processing engines. These engines have distinct responsibilities with respect to system processing and consequently they expose different API interfaces to the Business Rules they call. This section provides a brief overview of each engine in the platform and describes the engine's core responsibilities.

## Workflow Engine

The Workflow Engine is thought of as the controlling engine or the puppeteer. The main responsibility of this engine is to control and track the status of the business processes defined in the Workflow hierarchies. This engine is primarily accessed through the BRApi and can be called from other engines in order to check Workflow status during process execution. The Workflow Engine provides a very rich event model allowing each Workflow process to be evaluated and reinforced with customer specific business logic if required (see Appendix 2: Event Listing).

## Stage Engine

The Stage Engine performs the task of sourcing and transforming external data into valid analytic data points. The main responsibility of this engine is to read source data (files or systems) and parse the information into a tabular format. This allows the data to be transformed or mapped to valid Members defined by the Finance Engine. The Stage Engine is an in-memory, multi-threaded engine that provides the opportunity to interact with source data as it is being parsed and transformed. In addition to parsing and transforming data, the Stage Engine also has a sophisticated calculation that enables data to be derived and evaluated based on incoming source data. The Stage Engine provides quality services to source data by validating, mapping, and executing Derivative Check Rules.

## Finance Engine

The Finance Engine is an in-memory financial analytic engine. The main responsibility of this engine is to enrich and aggregate base data cells into consolidated multi-Dimensional information. The Finance Engine provides the opportunity to define sophisticated financial calculations through centralized Business Rules as well as member specific Business Rules (Member Formulas). It works concurrently with the Stage Engine to validate incoming intersections and works with the Data Quality Engine to execute Confirmation Rules which are used to validate analytic data values.

## Data Quality Engine

The Data Quality Engine is responsible for controlling data confirmation and certification processes. This Confirmation Engine is used to define and control the sequence of data value checks required to assert the information submitted from a source system is correct. The Certification Engine is responsible for managing user certifications and determining the Workflow dependents' completion status. This engine is primarily accessed through the BRApi and may be called from other engines in order to check data quality status during process execution.

## Data Management Engine

The Data Management Engine provides task automation services to the platform. This engine executes batches of commands that are organized into sequences which contain steps. Steps represent entry points or mechanisms to execute features of other engines. For example, the Clear Data Step uses the services of the Finance Engine. In addition, the Data Management Engine can execute a Business Rule Step which executes a custom Business Rule as part of a Data Management Sequence. This is an incredibly powerful capability because it provides the ability to string together any combination of predefined processing steps with custom Business Rule steps.

## Presentation Engine

The Presentation Engine provides extensive data visualization services to the Platform. The Presentation Engine is made up of the following component engines: Cube View Engine, Dashboard Engine, Parameter Engine, Book Engine and Extensible Document Engine. The Presentation Engine is responsible for managing and delivering content to the end user as well as providing a development environment for custom user interface elements. This engine enables the MarketPlace application development capabilities and continues to evolve with each product release. Like the Data Management Engine, the Presentation Engine interacts with and can call the services of all other engines in the product.

## Scaling Engine

This feature will be made available in a future release.

The Scaling Engine provides services that will determine whether the customer wants to Scale their Server Set or Database Elastic Pool on the Platform. This is only available to Cloud (Azure) and does not pertain to On-Premise solutions. For example, customer must be utilizing Azure Scale Set, and/or SQL Server Elastic Pool functionality. This provides the ability to create or delete a VM and/or increase/decrease database resources based on the logic that is designated in the System Extender Business Rules to meet the customer needs.

## BRApi

The BRApi is common across all Business Rules, engines and APIs being run, so it is not an engine itself. A BRApi function runs outside of the other engines and can orchestrate certain functions from within other engines. In other words, a BRApi function can be run from one engine (e.g. Parser) to tell other engines (e.g. Finance) to execute their own APIs (e.g. API.Data.GetDataCellUsingMemberScript). For another example, while the API.Data.GetDataCell function is available from within the Finance engine, a similar BRApi called GetDataCellUsingMemberScript can be run from any engine if given the appropriate arguments. A common use is BRApi.ErrorLog.LogMessage from any engine.

## Business Rule Classifications

OneStream provides classifications for business logic organization. At the core, all business logic is delivered and executed as compiled VB.Net or C# code. This means no matter what type of business logic is used, there is a consistency in the syntax and compilation process. The reason for different classifications has to do with when and how the business logic is invoked and how the business rule is scoped.

There are two broad business rule classifications: shared business rules and item specific business rules. Each engine in the system may support one or both business rule classifications. Whenever a processing sequence is executed in the platform, the particular engine(s) involved evaluates how and what business logic is associated with the process. This may include shared business rules (named and event handlers) as well as item specific business rules (member formulas, logical expressions, and confirmation rules).

**NOTE:** Shared business rules can be written in either VB.NET or C#, item specific business rules can be written in VB.NET only.

## Finance Engine Example

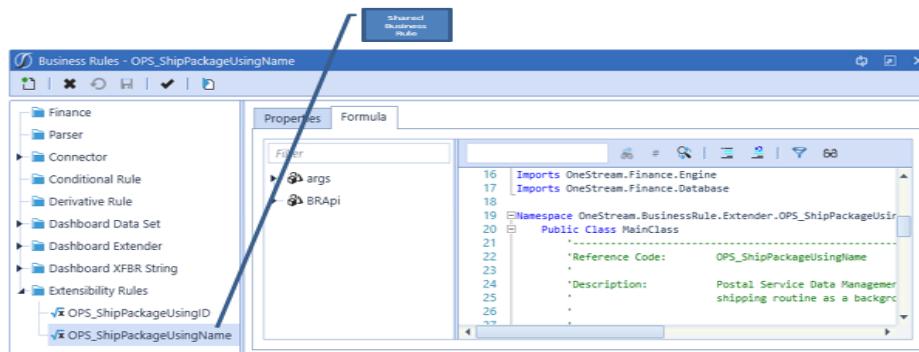
During a consolidation process, a Named Business Rule is associated with the Cube being processed. The Cube contains Member Formulas associated with some of its Dimensions. In this case, the Finance Engine compiles both the Named Business Rule and each individual Member Formula in preparation for the calculation sequence.

## Stage Engine Example

A similar example applies to the Stage Engine. During a parse and transform Workflow process, a Named Business Rule is associated with the Data Source or Transformation Rules. In addition, individual Data Source Dimensions or Transformation Rules have associated Logical Expressions that are also fired. In this case, the Stage Engine compiles both Named Business Rules and each individual Logical Expression in preparation for execution during the parse and transform execution sequence.

## Shared Business Rules

Shared Business Rules are reusable because the rule is written and stored centrally in the Business Rule Library. This means the same rule can be called or referenced by multiple platform components. For example, the Business Rule highlighted in the image below is a general Extensibility Rule. This rule can be executed from the Business Rule Editor, called by a Data Management Job or called by another Business Rule. Shared Business Rules are the code files seen in the tree when the OneStream Syntax Editor is open, they are organized by type, and named by the user who created the rule.



### Event Handler Business Rules

Event Handler Business Rules are a predefined set of Shared Business Rules and are always defined as an Extensibility Rule Type. Event Handler Rules are invoked during a processing sequence by their related platform engine in order to supplement the process. Determine/filter how/if the execution behaves for specific Workflows or the Cube POV. When an Event Handler Business Rule is called, the calling engine supplies information about the executed process providing context about the process and information about the specific sub-event executed.

### Predefined Event Handler Business Rules

The list below details the specific predefined Event Handlers available in the platform. For details on the individual sub-events that fire for each Event Handler Business Rule.

- Data Management Event Handler
- Data Quality Event Handler
- Forms Event Handler
- Journal Event Handler
- Save Data Event Handler
- Transformation Event Handler
- Workflow Event Handler
- Wcf Event Handler

## Item Specific Business Rules

Item Specific Business Rules are complete rules like Shared Business Rules, however they are authored and stored with the specific platform item with which the rule is associated. There are different reasons for using Item Specific Business Rules vs Shared Business Rules.

For example, when creating a one-off rule without any reusable value to other components in the system, write an Item Specific Business Rule directly on the platform component because it requires a very specific piece of business logic. Another example, which is more common when creating calculation logic for an analytic model, is to write a Member Formula that directly associates a calculation with a Dimension Member. This creates system maintenance clarity and maintainability.

Item Specific Rules, in particular Member Formulas, can have a positive performance impact because they allow calculations to be broken down into formula passes and processed in a parallel (multi-threaded) fashion. The same formulas can be written in a Shared Finance Business Rule, but the calculations will always execute in the serial manner defined in the rule.

### Item Specific vs Shared Code Structure

As mentioned above, an Item Specific Business Rule and a Shared Business Rule are identical in code structure. When writing an Item Specific Business Rule, the code editor presents some hidden sections in the code window:

- Formula Header
- Formula Footer

- Helper Function Header
- Helper Function Footer

These hidden sections (i.e. Regions) keep the formula / expression as readable as possible. In a Shared Business Rule, these sections are visible which make the rule more verbose. The idea behind the Item Specific Business Rule is to create discrete code blocks that are easy to manage and have limited interdependencies. If one knows how to write a Shared Business Rule, then she/he also knows how to write an Item Specific Business Rule and vice versa.

Item Specific Rules are categorized into three types: Member Formulas, Complex Expressions, and Confirmation Rules. These relate to the platform engine with which they are associated.

### **Member Formulas**

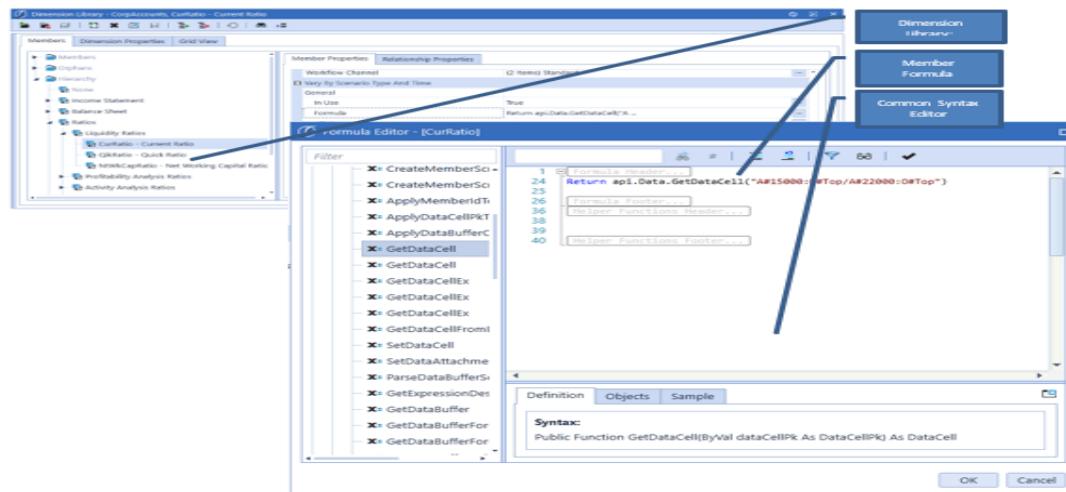
A Member Formula is assigned to a Dimension Member and executes within the Finance Engine during a Cube processing sequence. See Formula Guide in "About the Financial Model" on page 2 for more information on processing sequences. Member Formulas provide the same level of syntax and logic capability that exist when writing a Finance Shared Business Rule, however custom consolidation, elimination, and translation logic cannot be written. Member Formulas are a great choice for writing logic limited to calculations based on a single Member and calculations that do not span Dimensions. If Member Formulas are written with these constraints in mind, then the Dimension Member and its formula can be reused in different Cubes without having dependencies on other Dimensions. This does not mean that a Member Formula cannot look at other Dimensions. Referencing Dimension Members outside of the specific Dimension where the formula exists will limit the reusability of the Dimension, or require all referenced Dimensions be used together in any new Cube.

Member Formulas are written directly on a Dimension Member within the Dimension Library. Navigate to the specific Member's Formula property and click the ellipsis in order to store a Member Formula.

The example below is a simple working capital Member Formula.

## Implementing Security

---



### Complex Expressions

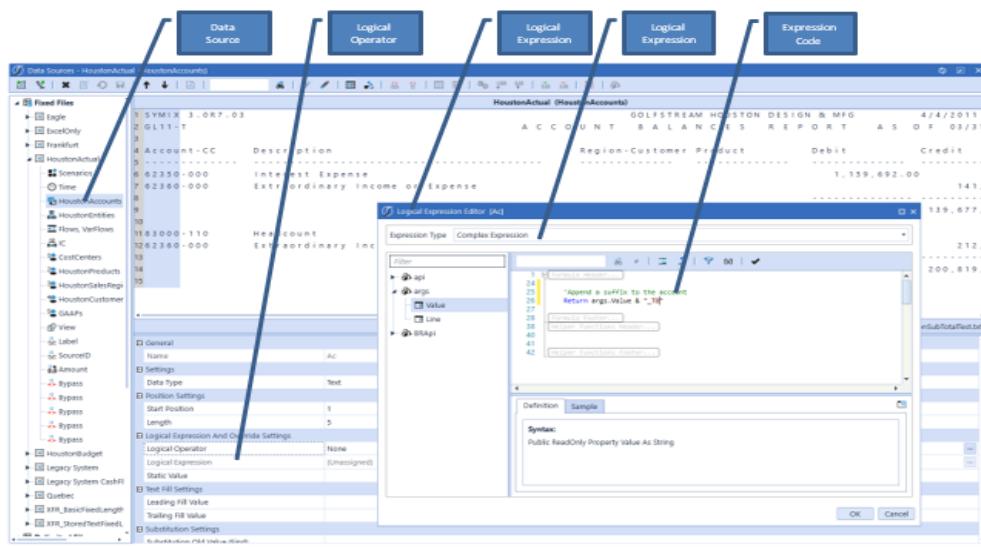
A Complex Expression is a Business Rule assigned to Data Source Dimensions, Derivative Rules, and Transformation Rules and execute within the Stage Engine during a transformation processing sequence. Complex Expressions provide the same level of syntax and logic capability that exist when writing a Stage Shared Business Rule. The primary reason for using a Complex Expression rather than a Stage Shared Business Rule is the logic being written has no reusability. Complex Expressions isolate the logic by associating it directly with a specific item.

### Using Complex Expressions in a Data Source

Apply Complex Expressions to a Data Source Dimension by selecting the Dimension requiring custom logic and setting the Logical Operator. The Logical Operator property opens the Logical Expression Editor dialog and allows the user to either select a Shared Parser Business Rule or write a Complex Expression. Both Shared Parser Business Rules and Parser Complex Expressions result in the exact same compiled Business Rule code. The exception is a Complex Expression is only executed for the Dimension to which it is applied, and a Shared Parser Rule is shared and can be called by many Dimensions.

## Implementing Security

---

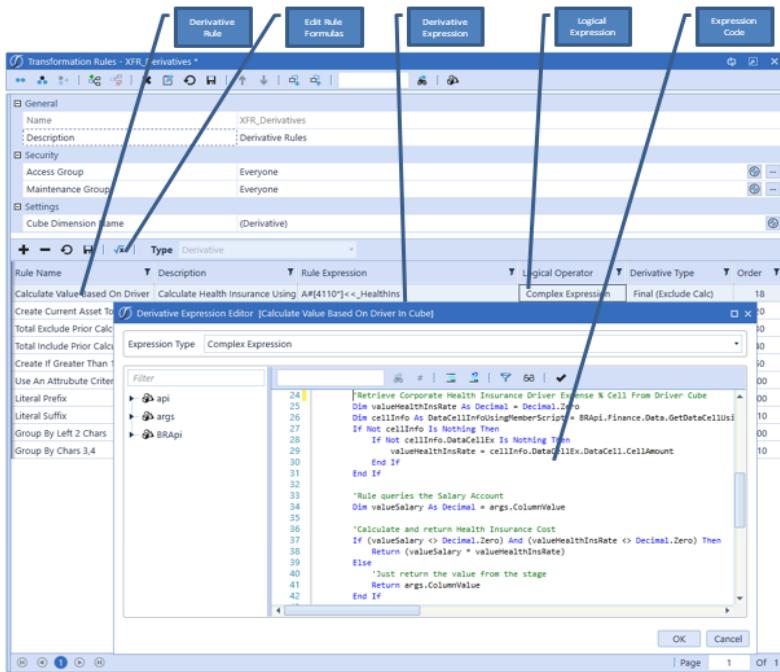


### Using Complex Expressions in a Derivative Rule

Apply Complex Expressions to a Derivative Rule by selecting the individual Derivative Rule

requiring custom logic and setting the Logical Operator. Clicking the Edit Rule Formulas toolbar button opens the Logical Expression Editor dialog and allows the user to either select a Shared Derivative Business Rule, write a Complex Expression, or use a Pre-Built Expression. Both Shared Derivative Business Rules and Derivative Complex Expressions result in the exact same compiled Business Rule code. The exception is a Complex Expression is only executed for the rule to which it is applied, and a Shared Derivative Rule is shared and can be called by many rules.

## Implementing Security



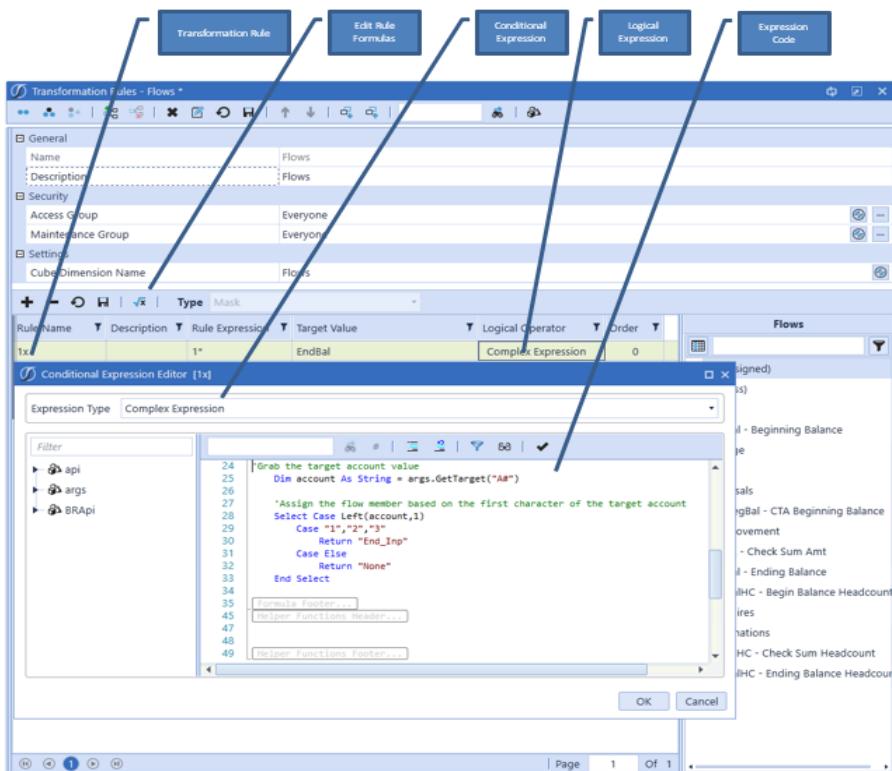
### Using Complex Expressions in a Conditional Transformation Rule

Apply Complex Expressions to a Transformation Rule by selecting the individual Transformation Rule requiring conditional logic and setting the Logical Operator. Clicking the Edit Rule Formulas

toolbar button opens the Logical Expression Editor dialog and allows the user to either select a Shared Conditional Business Rule or write a Complex Expression. Both Shared Conditional Business Rules and Conditional Complex Expressions result in the exact same compiled Business Rule code. The exception is a Complex Expression is only executed for the rule to which it is applied, and a Shared Conditional Rule is shared and can be called by many rules.

**NOTE:** Shared Conditional Business Rules and Complex Expressions cannot be applied to One-To-One Transformation Rule Types. One-To-One Transformation Rules are executed during the parsing process and therefore are completely processed prior to the conditional mapping process.

## Implementing Security



## Confirmation Rules

Confirmation Rules are called by the Data Quality Engine and Finance Engine. Apply Complex Expressions to Confirmation Rules by selecting the individual Confirmation Rule and clicking the

Edit Rule Formulas  toolbar button. This button opens the Rule Editor dialog and allows the user to write a Complex Expression containing the Confirmation Rule logic. A Confirmation Rule is only written on the specific rule to which it applies. Confirmation rules do not have an equivalent Shared Business Rule because each Confirmation Rule requires specific logic.

**TIP:** Shared Finance Business Rules can be called from a Confirmation Rule. Create standard helper functions in a Shared Finance Business Rule and call them from a specific Confirmation Rule creating some reusable logic and improving the overall Confirmation Rule infrastructure maintenance.

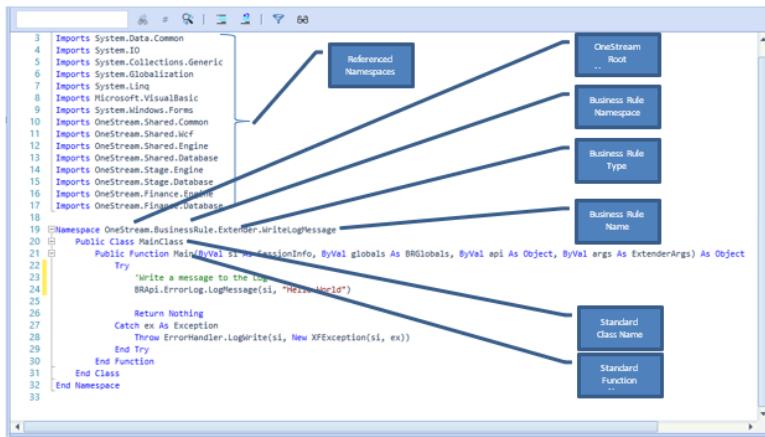
# Business Rule Organization and Referencing

Each OneStream Business Rule has a predefined Namespace, a Public Class and a Public Function that the OneStream platform engines invoke when the Business Rule needs to be called.

## Predefined Object Names

- Namespace: OneStream.BusinessRule.<Business Rule Type>.<Unique Business Rule Name>
- Class: MainClass
- Function: Main

Example Business Rule Structure



## Function Prototypes

Each Business Rule has one standard entry point Function Title called Main. The Function definition below represents the standard prototype used by the Main Function in each OneStream Business Rule. The Main Function always has the same standard parameter layout, but the last two parameters, API and ARGS, contain different object references based on the type of Business Rule being executed.

```
Public Function Main()
    ByVal si As SessionInfo,           à Connection Object Required to use API
    ByVal globals As BRGlobals,        à Global Variable Object Used to Share Values
    ByVal api As Object,              à Specific API object (Different for each Type)
    ByVal args As ExtenderArgs        à Specific Arguments (Different for each Type)
```

```
)  
As Object
```

## Defining a Reference to a Shared Business Rule

The Business Rule framework organizes the Business Rules so that reuse can be maximized. There are many situations where a Business Rule writer may create a standard function that is reused in many other Business Rules. The platform provides a way for Business Rules to be linked and called from other Business Rules. In addition, the platform provides a way for external DLL's to be linked and called from a Business Rule.

This section describes how to reference a Shared Business Rule from a within a Business Rule and how to reference an external DLL from within a Business Rule.

When a Shared Business Rule is created, its public members can be referenced and executed by other Shared and Item Specific Business Rules.

Common reasons to create a shared or referenced Business Rule:

- Create a list of shared constant values
- Create a set of standard helper functions
- Centralize maintenance of shared logic

### Shared Business Rules Referencing Other Shared Business Rules

In order to create a reference from one Shared Business Rule to another, navigate to the rule calling a Public Method of another Shared Business Rule and make a declaration in the Referenced Assemblies property. The syntax used to create a reference to another Shared Business Rule requires a BR\ prefix and the Business Rule name to reference.

**NOTE:** Reference more than one Business Rule by creating a comma-separated list of reference statements.



### Syntax

BR\<Business Rule Name to Reference>

#### Example (Single Reference)

BR\OPS\_PostalServiceHelper

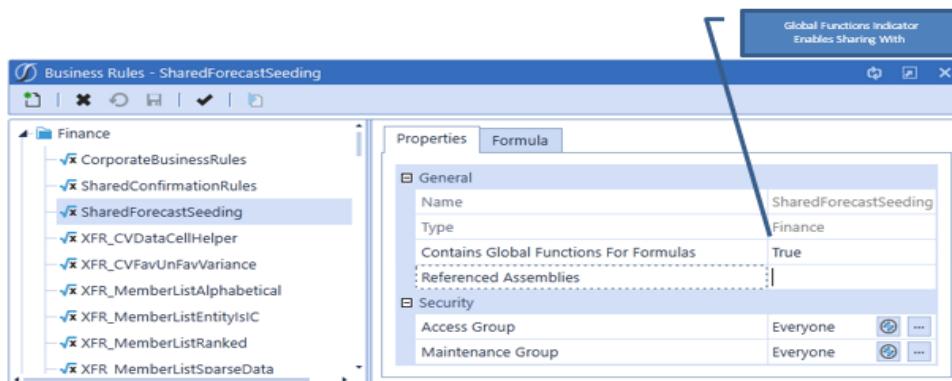
#### Example (Multiple References)

BR\OPS\_PostalServiceHelper; BR\CPP\_SolutionHelper

### Item Specific Rules Referencing Shared Business Rules

Finance, Parser, ConditionalRule and DerivativeRule Shared Business Rules all have equivalent Item Specific Business Rules. When creating a Shared Business Rule, set the Contains Global Functions For Formulas to True in order to make the Shared Business Rule available to Item Specific Business Rules. Setting this property makes the Shared Business Rule available without having to place a reference on the item using the rule. Item Specific Business Rules do not have a Referenced Assemblies property and therefore can only reference Shared Rules of the same engine type with the Contains Global Functions For Formulas property set to True.

In the screenshot below, the SharedForecastSeeding Rule can be called from any other Finance Rule because its Contains Global Functions For Formulas property is set to True.



**NOTE:** When a Finance Business Rule has its Contains Global Functions For Formulas set to True, any changes made to the Business Rule causes a metadata status impact and changes the Calculation Status to OK, MC. This dependency must occur because a global rule can be used by a Member Formula calculation, and therefore can impact the status of the Finance Engine's data (analytic / Cube data).

### Code Declaration

Once a reference is made to a Shared Business Rule, the Business Rule's Public Methods (Functions / Subs) can be called. To access the Shared Business Rule's Public Methods, declare an instance of the rule in the code using the Business Rule's fully qualified Namespace. This creates an object variable that references the Shared Business Rule and calls its Public Methods.

### Example Declaration

```
'Declaring an object variable to reference a Shared Business Rule  
Dim opsHelper As New OneStream.BusinessRule.DashboardExtender.OPS_PostalServiceHelper.MainClass
```

### Example Usage

```
'Executing a Function on the Reference Business Rule Object Variable  
Dim desc As String = opsHelper.GetFieldFromID(si, "Dashboard", "Name", dashName, "Description")
```

## Defining a Reference to an External .Net DLL

Developers can build and reference their own custom Microsoft .Net DLLs from Shared Business Rules. These are written in either VB.Net or C#. Custom business logic can be encapsulated and protected within an external DLL written in Microsoft Visual Studio.

Common reasons to create a custom DLL referenced by a Business Rule:

- Protect domain specific intellectual property (hide value programming logic)
- Separate code with dependencies on other programs (system integration wrappers)
- Complex logic requiring development tools only available within Microsoft Visual Studio (Web Service Discovery and Interface Development)

### DLL Installation and Configuration

This section defines the configuration steps that must be completed before an external DLL can be referenced within a Shared Business Rule. This is a three-step process.

1. Specify the BusinessRuleAssemblyFolder located in the Application Server configuration file

This folder should be shared by all application servers meaning the folder must be accessible by the Account Credentials used to configure the IIS Application Pool on the application server.

This setup process is a best practice but is not required. As an alternative, reference the external DLL from a folder located on each application server and any time the DLL is updated, it needs to be copied to a standard folder on each application server.

2. Identify or create the external DLL to be called and copy it to the BusinessRuleAssemblyFolder

When a Business Rule is executed and an external DLL reference containing the XF\ prefix is found in the Referenced Assemblies property of the rule, the application server will look in the BusinessRuleAssemblyFolder defined in the application server configuration file in order to find the DLL to be referenced.

3. Add a reference specification to the DLL in the Referenced Assemblies property of the Business Rule using it.

### Reference Specification

This section defines the syntax required to reference an external DLL by setting the Shared Business Rule's Referenced Assemblies property. There are three methods available for referencing an external DLL.

#### Method 1

This method uses the XF\ prefix to create a reference to an external DLL located in the BusinessRuleAssemblyFolder folder which is specified in the application server configuration file.

#### Syntax

XF\<External DLL Name to Reference>

#### Example (Single Reference)

XF\ExternalCode.DLL

#### Example (Multiple References)

XF\ExternalCode1.DLL;XF\ExternalCode2.DLL

#### Method 2

This method uses the file system path C:\DLLFolderName\ to create a reference to an external DLL located on each application server.

**NOTE:** The same folder path and DLL must exist on all application servers. This referencing method is not a best practice for custom business logic DLLs because it creates a maintenance and update burden.

Using a file system path reference is a valid method when referencing an external DLL that already exists on an application server. The DLL exists on the application server as part of the operating system or another installed software component.

#### Syntax

C:\DLLFolderName\<External DLL Name to Reference>

#### Example (Single Reference)

C:\DLLFolderName\ExternalCode.DLL

### Example (Multiple References)

C:\DLLFolderName\ExternalCode1.DLL; C:\DLLFolder\ExternalCode2.DLL

### Code Declaration

Once a reference is made to an External DLL from a Shared Business Rule, the Public Methods (Functions / Subs) of that External DLL can be called. In order to access the Shared Business Rule's Public Methods, declare an Import to the Namespaces defined by the DLL, and then create an instance of the desired class to utilize in the code.

### Example Import

Imports YourNamespace.SubNamespace

### Example Declaration

'Declaring an object variable to reference a class on the external DLL  
Dim extHelper As New YourClass

### Example Usage

'Executing a Function on the external DLL  
Dim desc As String = extHelper.YourFunciton("SomeParameter")

### Method 3

This method uses a Windows environment variable to create a reference to an external DLL. All standard Windows paths are supported, and the name is determined by .NET.

### Syntax

%System%\DLLName.DLL

### Example

%userprofile%\documents\WindowsBase.DLL

## Cube View Extender: Advanced Cube View Formatting

To apply advanced formatting to Cube Views, users can apply a Cube View Extender Business Rule. See Business Rules in "Application Tools" on page 727 for more information. A custom formatting formula built inside the Cube View can also be used. Using custom formatting allows the Cube View design to go beyond the standard Cube View formatting properties and provides flexibility for specific formatting needs. See the OneStream API Overview Guide as well as the OneStream API Details and Database Documentation Guide for more details on how this Business Rule is used.

# Cube View Extender Business Rule Structure

When creating a new Cube View Extender Business Rule or Formula within a Cube View, some example logic is presented in order to have a starting point rather than starting the rule from scratch. This lays out a general structure for what can be accomplished within a custom rule like this, but the logic is commented out initially. There are Cube View Extender Snippets available for download from the MarketPlace Store under the Snippet Editor Solution. These will also help with a starting point for common uses.

Below is an example of a Cube View Extender Business Rule for review and understanding:

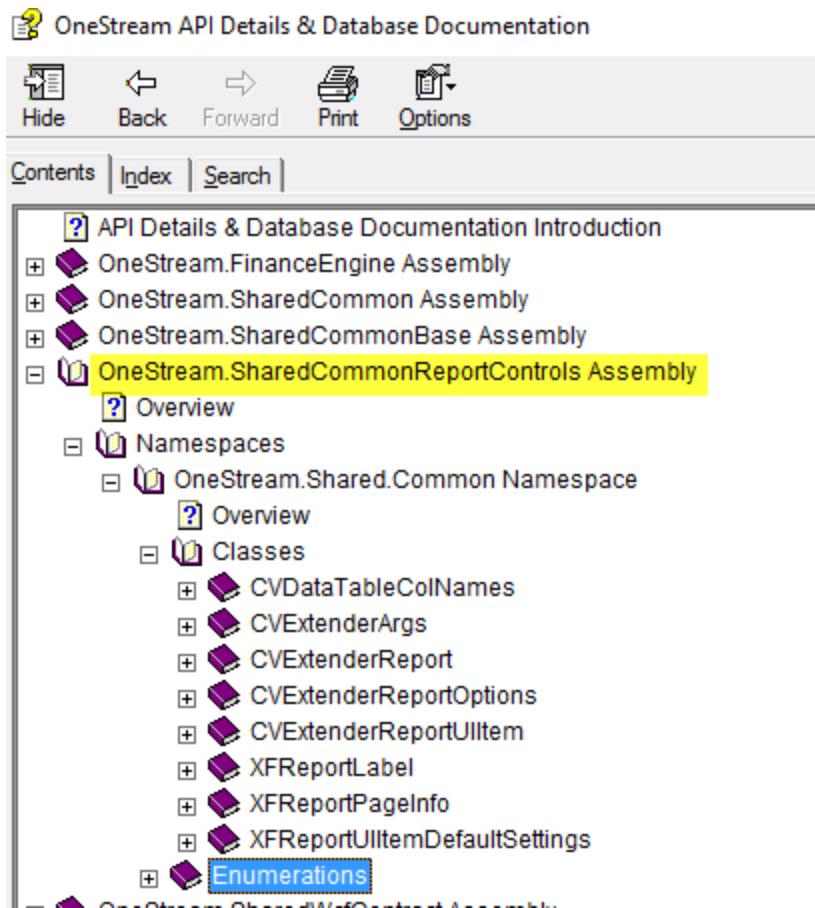
```
Select Case args.FunctionType
Case Is = CVExtenderFunctionType.GetReportOptions
    Dim reportOptions As New CVExtenderReportOptions()
    reportOptions.ReportMarginTop = 90
    reportOptions.ReportMarginBottom = -1.0
    reportOptions.PageHeaderTitlesHeight = -1.0
    reportOptions.PageFooterHeight = -1.0
    Return reportOptions
End Select
```

Case Is = CVExtenderFunctionType.FormatReportUIItem  
Dim uiItem As CVExtenderReportUIItem = args.Report.CurrentUIItem

If uiItem.UIItemType = XFRReportUIItemType.DataCellLabel Then  
 Dim cvRow As CubeViewRow = uiItem.GetCubeViewRow()  
 If Not cvRow Is Nothing Then  
 If cvRow.Name.XFContainsIgnoreCase("Total") Then  
 uiItem.BackgroundColor = XFColors.Red  
 End If  
 End If  
 If uiItem.XFHasData Then  
 If uiItem.XFAmount < 50.0 Then  
 uiItem.BackgroundColor = XFColors.Yellow  
 End If  
 Else  
 uiItem.Text = "NODATA"  
 uiItem.TextColor = XFColors.XFLightBlue  
 uiItem.FontSize = 14  
 End If  
End If

End Select

See the *OneStream API Overview Guide* for more details on this Business Rule including the various lists of potential options available for setting properties. Search in the below highlighted section of the API Guide:



The breakdown of each section for this type of rule are below:

### **Args.FunctionType**

Select Case args.FunctionType is the expression used when a certain process needs to be isolated and run special logic. The Case statement in the example Business Rule above is necessary to determine one of two operations that are being performed to generate the Cube View Report. These are GetReportOptions or FormatReportUIItem.

### **CVExtenderFunctionType.GetReportOptions**

This is for the retrieval of and setting of properties such as margins and height of title and footer. A value of -1 means to use the default value and is not necessary if that property is not being overridden. Any of these numeric settings is represented in pixels, roughly the width of a human hair, so these are precise measurements.

Select Case args.FunctionType

Case Is = CVExtenderFunctionType.GetReportOptions

Dim reportOptions As New CVExtenderReportOptions()

```
reportOptions.ReportMarginTop = 100  
reportOptions.ReportMarginBottom = -1  
reportOptions.PageHeaderTitlesHeight = 20  
reportOptions.PageFooterHeight = -1  
  
Return reportOptions
```

Notice in the rule above a new CVExtenderReportOptions object is declared so that a few properties can be set for that object and then written back with the Return statement at the bottom of the section. Without the Return statement, the properties do not get set for this particular instance when this report is run.

These property settings will not override the saved properties for this Cube View, but just override at run time. Properties that can be set in a CVExtenderReportOptions object:

- PageFooterHeight
- PageHeaderTitlesHeight
- ReportMarginBottom
- ReportMarginTop

### **CVExtenderFunctionType.FormatReportUIItem**

The key concept to grasp for how these Cube View Extender Business Rules apply to the formatting of a report is that every item is looped through and is eligible to have its format changed through logic. When a report is run, it plots every label, every line, everything seen on the page one by one. Based on the current item being processed (i.e. Args.Report.CurrentUIItem), the system has the context of many properties (which depend on the type of item it is, or UIItemType) and the Cube View Column or Row to which it is related.

This is for formatting a specific Report User Interface Item, which could be a data cell, line, footer, header, row header, column header or another item on the report. These object types are called a UIItemType.

Case Is = CVExtenderFunctionType.FormatReportUIItem

```
Dim uiltem As CVExtenderReportUIItem = args.Report.CurrentUIItem  
  
If uiltem.UIItemType = XFReportUIItemType.DataCellLabel Then...
```

Based on the UIItemType, apply logic that will set properties as appropriate and based on conditions.

### Args.Report

Can set properties in the report view of the Cube View output. See below for some examples.

- CurrentUllItem

Example: Dim uiltem As CVExtenderReportUllItem = args.Report.CurrentUllItem

This is the key Report Arg as it returns an object called CVExtenderReportUllItem

- A CVExtenderReportUllItem can return:

- UllItemType

This a key property. This will reveal if the Report Item is a Label, Page Header or Footer Label, Row or Column Header Label or a Data Cell Label (meaning an amount or cell on the report). Once it knows what type of Report Item is being analyzed, certain properties can be applied.

Example of this logic:

If uiltem.UllItemType = XFReportUllItemType.DataCellLabel Then...

- Whether the item has data (uiltem.XFHasData) if the type is DataCellLabel
      - The amount of the DataCellLabel (uiltem.XFAmount)
      - Text stored and how formatted (uiltem.Text, uiltem.FontFamily or uiltem.FontSize)
      - The item's Name (uiltem.Name)
      - Change the colors (uiltem.TextColor, uiltem.BackgroundColor, uiltem.BorderColor, etc.) with a statement such as uiltem.BackgroundColor = XFCOLORS.Yellow
      - Change borders and lines (uiltem.BorderSides, uiltem.BorderLineStyle, etc.)
      - Whether the item can grow or shrink based on content, if it needs to stay on one row, or how large that row or column should be (uiltem.CanGrow, uiltem.CanShrink, uiltem.Padding or uiltem.WordWrap)

- Margin sizes

- api.Report.MarginLeft
  - api.Report.MarginRight

- `api.Report.PageWidthMinusMargins`
- Page position and margin width  
`api.Report.CurrentPageInfo` controls the left/right/center positioning as well as page/header width
- Auto Fit settings
  - `api.Report.AutoFitToPageWidth`
  - `api.Report.AutoFitNumPagesWide`

### **Args.CubeView**

These Args can be used to retrieve properties from a Cube View being processed when the report is run, but really are here more for internal use. Some of these properties could be used as conditions, when setting labels or other properties. Examples are:

- Paper Size
- Margins
- Titles
- Headers & Footers
- Cube View POV settings
- Row & Column Height & Width

### **Args.PageInstanceState**

This is for setting Dashboard Page State and is not related to Cube View Extenders.

### **Args.CustomSubstVars**

This is not related to Cube View Extenders for the most part. They can provide the ability to retrieve the name value pairs of a custom Parameter applied when this Cube View was run and the choice the user made. For example, if the user selects an Entity upon running the report, MyEntity = Houston could be returned and used in a custom report, however, there are other methods to apply this same information.

### Manipulating Formats based on the related Cube View Row or Column

When custom formatting should only impact specific Cube View Report Rows or Columns related to the CurrentUIItem being processed, this type of statement can be used to first declare a new row or column object. It can then apply formatting based on the properties retrieved from this part of the Cube View related to that item, such as a Row Name.

Example:

```
If uiItem.UIItemType = XFReportUIItemType.DataCellLabel Then  
    Dim cvRow As CubeViewRow = uiItem.GetCubeViewRow  
    Dim cvCol As CubeViewCol = uiItem.GetCubeViewColumn
```

As shown in the previous example, a test can be performed to see if there is anything to be rendered in this column or row with If Not cvRow Is Nothing Then, which is a good practice.

Conditional logic can be used before applying formatting to labels or data cells such as the Name of the Cube View Row related to the CurrentUIItem:

```
If cvRow.Name.XFContainsIgnoreCase("Total") Then...
```

## Common Cube View Extender Business Rule Examples

### Controlling Logo Display on Cube Views

Logos display in the Cube View Report Header only, but can vary by report, user security, be displayed on some reports and concealed on others, and placed on the left, center or right side of the page.

#### Conceal the Logo on Cube View Reports

Use Case: Users do not want to display a logo on a specific Cube View Report.

Function:

```
Case Is = CVExtenderFunctionType.FormatReportUIItem  
    Dim uiItem As CVExtenderReportUIItem = args.Report.CurrentUIItem  
    If uiItem.UIItemType = XFReportUIItemType.PageHeaderPictureBoxLogo Then  
        uiItem.Visible = False  
    End If
```

#### Vary by User Security

Use Case: Vary the logo that displays on a Cube View Report based on a user's security group. The alternative logos must be in png format and saved in the OneStream File Explorer in order to reference them in the function.

Function:

## Implementing Security

---

```
Case Is = CVExtenderFunctionType.FormatReportUIItem
    Dim uiItem As CVExtenderReportUIItem = args.Report.CurrentUIItem
    If uiItem.UIItemType = XFReportUIItemType.PageHeaderPictureBoxLogo Then
        If BRApi.Security.Authorization.IsUserInGroup(si, "SecurityGroup") Then
            uiItem.SetPictureBoxImage
    (FileSystemLocation.ApplicationDatabase, "Documents/Public/Calculate.png",
        TriStateBool.TrueValue)
End If
```

## Vary Logo by Cube View Report

Use Case: Vary the logo that displays on a specific Cube View Report, based on the Page Caption property. The alternative logos must be in png format and saved in the OneStream File Explorer in order to reference them in the function.

Function:

```
Case Is = CVExtenderFunctionType.FormatReportUIItem
    Dim uiItem As CVExtenderReportUIItem = args.Report.CurrentUIItem
    If uiItem.UIItemType = XFReportUIItemType.PageHeaderPictureBoxLogo Then
        If args.CubeView.PageCaption.Contains("Corp") Then
            uiItem.SetPictureBoxImage
    (FileSystemLocation.ApplicationDatabase, "Documents/Public/Corporate.png",
        TriStateBool.TrueValue)
        Else
            uiItem.SetPictureBoxImage
    (FileSystemLocation.ApplicationDatabase, "Documents/Public/Standard.png", TriStateBool.TrueValue)
End If
```

## Vary Logo by Entity

Use Case: Vary the logo that displays on a specific Cube View Report, based on the Entity mentioned in the Cube View POV. The alternative logos must be in png format and saved in the OneStream File Explorer in order to reference them in the function.

Function:

```
Case Is = CVExtenderFunctionType.FormatReportUIItem Dim uiItem As CVExtenderReportUIItem =
args.Report.CurrentUIItem
    If uiItem.UIItemType = XFReportUIItemType.PageHeaderPictureBoxLogo Then
        If args.CubeView.CubeViewPovMembers.Entity.Name.Contains("Corp") Then
            uiItem.SetPictureBoxImage
    (FileSystemLocation.ApplicationDatabase, "Documents/Public/Corporate.png",
        TriStateBool.TrueValue)
        Else
```

## Implementing Security

---

```
    uiItem.SetPictureBoxImage  
(FileSystemLocation.ApplicationDatabase,"Documents/Public/Standard.png", TriStateBool.TrueValue)  
End If
```

## Move Logo in Header

```
Case Is = CVExtenderFunctionType.FormatReportUIItem  
    Dim uiItem As CVExtenderReportUIItem = args.Report.CurrentUIItem  
  
    If uiItem.UIItemType = XFReportUIItemType.PageHeaderPictureBoxLogo Then  
        uiItem.Left = args.Report.CurrentPageInfo.RightPosition - uiItem.Width  
  
End If
```

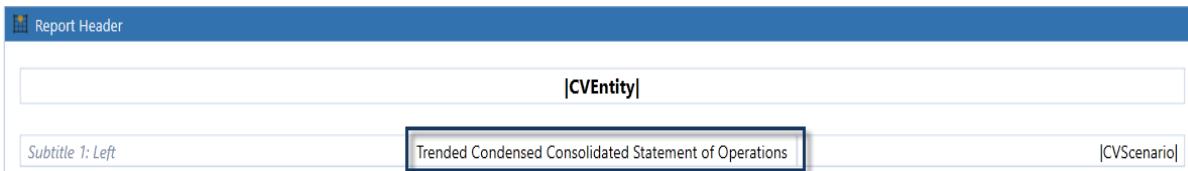
**NOTE:** CurrentPageInfo can also use CenterPosition in order to move the logo to the center of the report.

## Customizing Cube View Report Headers/Footers

The following examples use header settings and can also be applied to footers.

### Controlling the Width of Left, Right and Center Header Subtitles

Use Case: Control how a Report Header or Header Subtitles word wrap. When designing a Report Header using a combination of left, center, and right subtitles, each subtitle will automatically word wrap. For example, the header below has a long center subtitle and a right subtitle.



At runtime, this is how the Cube View Report displays:



## Implementing Security

---

In order to prevent the Center Subtitle from wrapping, apply a Cube View Extender function and control the width.

Function:

```
Case Is = CVExtenderFunctionType.FormatReportUIItem
    Dim uiItem As CVExtenderReportUIItem = args.Report.CurrentUIItem
    If uiItem.UIItemType = XFReportUIItemType.PageHeaderLabelCenter1 Then
        uiItem.Width = 350
End If
```

Result:



### Applying a Color Border to a Cube View Header

Use Case: The Cube View Header or Subtitle(s) need a border around it. Users can control the Border Sides, the Border Color, and the Border Thickness.

Function:

The function below is controlling the Center Subtitle's width and putting a border around it.

```
Case Is = CVExtenderFunctionType.FormatReportUIItem
    Dim uiItem As CVExtenderReportUIItem = args.Report.CurrentUIItem
    If uiItem.UIItemType = XFReportUIItemType.PageHeaderLabelCenter1 Then
        uiItem.Width = 350
        uiItem.BorderSides = XFSides.All
        uiItem.BorderColor = XFCOLORS.XFDarkBlueBackground
        uiItem.BorderThickness = 5
    End If
```

Result:



## Implementing Security

---

### Applying a Background Color/Font Color to a Header

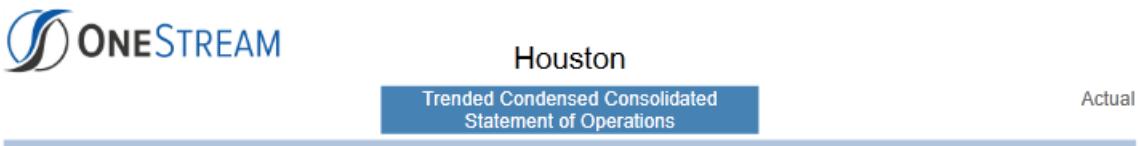
Use Case: The Cube View Header or Subtitle(s) need a specific background color or font color. In the example below, the user would like the Center Subtitle highlighted in blue and the text in white font.



Function:

```
Case Is = CVExtenderFunctionType.FormatReportUIItem
    Dim uiItem As CVExtenderReportUIItem = args.Report.CurrentUIItem
    If uiItem.UIItemType = XFReportUIItemType.PageHeaderLabelCenter1 Then
        uiItem.BackgroundColor = XFColors.SteelBlue
        uiItem.TextColor = XFColors.White
    End If
```

Result:



### Cube View Text Wrapping

You can apply a consistent line width across a Cube View Row when text wrapping occurs in one or more Row expansions.

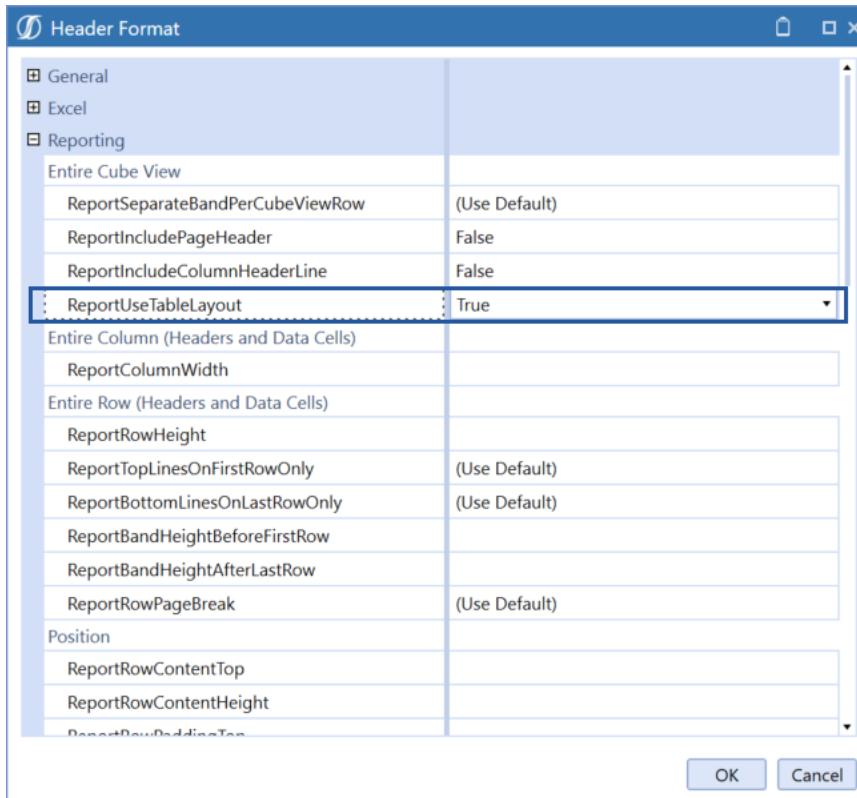
It is enabled on the Cube View Default as Header formatting.

Cube View Header Format		
	Col1	Col2
Default	9,999.99	9,999.99
Row1		

## Implementing Security

---

There is a setting in the Reporting section of Header Formatting called ReportUseTableLayout. The ReportUseTableLayout default setting is False, which is the standard Cube View Label structure.



If set to True, the Labels are changed to a Table structure which allows the row height to be formatted across rows as text wrapping occurs within Row expansions.

**GolfStream™**  
Get Back to the Green

Cube View Text Wrap Formatting

Jan 2018	
Earnings Before Taxes	4,270,990.88
Earnings Before Interest and Taxes	4,511,155.38
Interest Income	52,758.50
IC Interest Income	0.00
Interest Expense	292,923.00
IC Interest Expense	0.00

# Conditional Formatting

Use Case: Format cells based on the cell content. In the example below, the function will highlight any value less than \$500,000 and place the text NODATA in any cell without a value.

Function:

```
Case Is = CVExtenderFunctionType.FormatReportUIItem
    Dim uiItem As CVExtenderReportUIItem = args.Report.CurrentUIItem
    If uiItem.UIItemType = XFReportUIItemType.DataCellLabel Then
        Dim cvRow As CubeViewRow = uiItem.GetCubeViewRow()
        If Not cvRow Is Nothing Then
            End If
            If uiItem.XFHasData Then
                If uiItem.XFAmount < 500000.0 Then
                    uiItem.BackgroundColor = XFColors.SteelBlue
                End If
            Else
                uiItem.Text = "NODATA"
                uiItem.TextColor = XFColors.DarkBlue
                uiItem.FontSize = 9
            End If
        End If
    End If
```

Result:



Total GolfStream

### Balance Sheet

	Mar 2011	Mar 2010
10000 - Petty Cash	28,172,900.04	4,193,051.98
10100 - Cash Deposits	95,423,421.34	76,714,574.05
10200 - Other Cash Equivalents	753,451.40	NODATA
10300 - Marketable Securities	12,123,388.00	9,941,178.16
10400 - Restricted Cash	11,128,389.00	9,125,278.98
<b>10999 - Total Cash</b>	<b>147,601,549.78</b>	<b>99,974,083.17</b>
11000 - Trade Receivables	81,983,391.73	36,709,795.02
11100 - Other Receivables	289,822.35	NODATA
11300 - Allowance for Doubtful Accounts	4,022,733.30	3,180,163.19
<b>11999 - Net Accounts Receivable</b>	<b>78,250,480.78</b>	<b>33,529,631.83</b>
12000 - Raw Materials Inventory	16,386,567.83	11,703,935.14
12100 - Work in Progress Inventory	11,397,331.20	6,470,147.82
12200 - Finished Goods Inventory	37,669,079.03	7,371,835.94
12300 - Supplies - Inventory	1,005,568.65	614,295.43
12400 - In Transit Inventory	3,534,894.22	164,273.88
<b>12999 - Total Inventories</b>	<b>69,993,440.92</b>	<b>26,324,488.20</b>
13000 - Prepaid Insurance	186,701.96	133,498.16
13200 - Prepaid Taxes	992,833.00	814,123.06
13300 - Prepaid Other	612,352.05	427,272.96
<b>13999 - Total Prepaid Expenses</b>	<b>1,791,887.01</b>	<b>1,374,894.18</b>

## Page Number Display in Footer

Use Case: Format the page number in the Footer to display Page x of xx or Page x. Below is the standard page number format. Note that the setting to add a page number to Cube Views running in Data Explorer Report mode is determined under OnePlace/Application/Application Properties.

8/27/2017 12:25:14 PM

1

```
SetPageNumberDisplayInfo Function: Page x of xx
Case Is = CVExtenderFunctionType.FormatReportUIItem
    Dim uiItem As CVExtenderReportUIItem = args.Report.CurrentUIItem
    If uiItem.UIItemType = XFReportUIItemType.PageFooterPageNumber Then
        uiItem.SetPageNumberDisplayInfo(True, "")
        uiItem.SetPageNumberDisplayInfo(True, "Page {0} of {1}")
    End If
```

Result:

8/27/2017 12:28:40 PM

Page 1 of 2

```
SetPageNumberDisplayInfo Function: Page x
Case Is = CVExtenderFunctionType.FormatReportUIItem
    Dim uiItem As CVExtenderReportUIItem = args.Report.CurrentUIItem
    If uiItem.UIType = XFReportUIItemType.PageFooterPageNumber Then
        uiItem.SetPageNumberDisplayInfo(False, "")
        uiItem.SetPageNumberDisplayInfo(False, "Page {0}")
    End If
```

Result:

---

8/27/2017 12:31:01 PM

Page 1

## Dashboard Extender Business Rule Examples

### Using Page State

A page state is used in a Business Rule to store information about a specific Dashboard page instance. If a user has multiple tabs open with two different Dashboards, different state information can be stored for each page. For example, when a Dashboard is first opened, a Dashboard Extender Rule can run in the Dashboard's Load Dashboard event and then store the results of some database queries. Once the Parameter's selection changed, another extender could read the page state.

To use page state in a Dashboard Extender Rule, get the Page Instance ID using

args.PageInstanceStateInfo.PageOrDlgInstanceId

Then pass that using one of the following BRApi functions:

- BRApi.State.DeletePageOrDlgState
- BRApi.State.DeleteAllStateForPageOrDlg
- BRApi.State.SetPageOrDlgState
- BRApi.State.GetPageorDlgState

## Dashboard DataSet Business Rule Examples

### Map Dashboard Component Business Rule Examples

The Map Component is used to display specific locations on a geographical map via a Dashboard DataSet Business Rule. The XFMapItemCollection objects provide the ability to pass in Parameters in order to drill down on these locations and display data.

#### Pinpoint Example

This places a clickable pinpoint at each geographical location. A Parameter value can be included in the string in order to generate an action upon clicking the pinpoint.

The string defines the following:

(Latitude, Longitude, Location Label, Image to display on map, latitude pixel shift (if necessary), longitude pixel shift (if necessary), Parameter Value, Image to display when hovering over the location)

```
Dim pinPoints As New List(Of XFMappinPoint)
pinPoints.Add(New XFMappinPoint(42.68342, -83.13702, "Rochester", XFIImageFileType.ClientImage,
"StatusGreenBall",
0, 0, "Clubs", XFIImageFileType.ClientImage, "StatusRedBall"))
```

**NOTE:** See Business Rule Client Image Types for a list of available status images.

Result:



#### Ellipses Example

This places a clickable ellipse at each geographical location. A Parameter value can be included in the string in order to generate an action upon clicking the ellipsis.

## Implementing Security

---

The string defines the following:

(Latitude, Longitude, ellipse width, ellipse height, ellipse color, opacity, stroke color (border), stroke thickness, Parameter Value, color when hovering over the ellipse, hover opacity, hover stroke, hover stroke thickness)

```
Dim ellipses As New List(Of XFMapEllipse)
ellipses.Add(New XFMapEllipse(50.5, .5, 4, 2, XFCOLORS.Yellow.GetHexString(), .5,
    XFCOLORS.Red.GetHexString(), 2, "ellipseParam1", XFCOLORS.Green.GetHexString(),
    1, XFCOLORS.Blue.GetHexString(), 4))
```

Result:



## Polyline Example

This creates a continuous line composed of several pre-determined line segments.

First define the coordinates for each polyline segment. The string then defines the following:  
(Polyline name, reference to the local variables above, line color, line thickness, Parameter Value, hover color, hover thickness)

```
Dim polylines As New List(Of XFMapPolyline)
Dim points As New List(Of XFMapPoint)
points.Add(New XFMapPoint(20, 20))
points.Add(New XFMapPoint(30, 25))
points.Add(New XFMapPoint(40, 15))
polylines.Add(New XFMapPolyline("Polyline1", points, XFCOLORS.Black.GetHexString(), 3,
    "polylineParam1", XFCOLORS.Green.GetHexString(), 6))
```

## Polygons Example

This creates a polygon shape which outlines a specific location on the map.

First define the coordinates to apply to the polygon. The string defines the following:  
(Polygon name, reference to the local variables above, polygon color, opacity, stroke, stroke thickness, Parameter Value, hover color, hover opacity, hover stroke, hover stroke thickness)

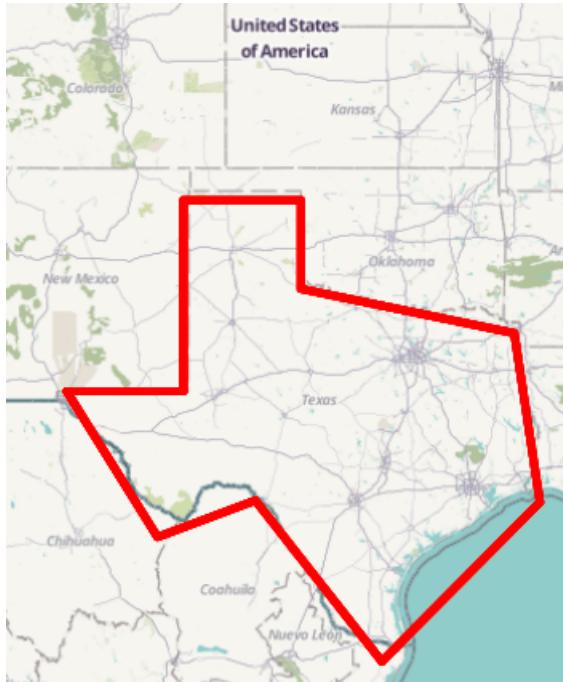
```
Dim texasPoints As New List(Of XFMapPoint)
texasPoints.Add(New XFMapPoint(36.29, -103.2))
texasPoints.Add(New XFMapPoint(36.29, -100))
texasPoints.Add(New XFMapPoint(34.33, -100))
```

## Implementing Security

---

```
texasPoints.Add(New XFMapPoint(33.33, -94.2))
texasPoints.Add(New XFMapPoint(29.41, -93.5))
texasPoints.Add(New XFMapPoint(25.57, -97.8))
texasPoints.Add(New XFMapPoint(29.45, -101.24))
texasPoints.Add(New XFMapPoint(28.58, -103.9))
texasPoints.Add(New XFMapPoint(32, -106.37))
texasPoints.Add(New XFMapPoint(32, -103.2))
polygons.Add(New XFMapPolygon("Texas", texasPoints, XFColors.white.GetHexString(), 0,
XFColors.Red.GetHexString(), 5, "Texas", String.Empty, 0, XFColors.Blue.GetHexString(), 5))
```

Result:



## Gantt Dashboard Component Business Rule Example

This Dashboard Data Set Business Rule demonstrates how to create a hard-coded list of tasks to display in a Gantt View Control.

```
If args.DataSetName.XFEqualsIgnoreCase("GanttDataSet") Then
    'Create a DataSet from the XFGanttTaskCollection
    '-----
    'Define the properties that we need to set for each task
    Dim now As DateTime = DateTime.UtcNow
    Dim taskName As String = String.Empty
    Dim taskTitle As String = String.Empty
    Dim taskDesc As String = String.Empty
    Dim wfStatusType As WorkflowStatusTypes = WorkflowStatusTypes.Unknown
```

```
Dim imageSource As XFIImageFileType = XFIImageFileType.ClientImage
Dim imageNameOrPath As String = ""
Dim taskStartTime As DateTime = now
Dim taskEndTime As DateTime = now.AddDays(2)
Dim taskDeadline As DateTime = now.AddDays(5)
Dim isMilestone As Boolean = False
Dim percentComplete As Double = 0
Dim isHighlighted As Boolean = False
Dim taskParameters As String = String.Empty
Dim dependencies As New List(Of String)
Dim children As New List(Of XFGanttTaskItem)

'Create Task 1
taskName = "MyTask1"
taskTitle = "MyTask1 Title"
taskDesc = "MyTask1 Description"
wfStatusType = WorkflowStatusTypes.Unknown
imageSource = XFIImageFileType.ClientImage
imageNameOrPath = XFClientImageTypes.StatusGrayBall.Name
taskStartTime = now
taskEndTime = now.AddDays(3)
taskDeadline = now.AddDays(6)
isMilestone = False
percentComplete = 25
isHighlighted = False
taskParameters = "MyParam=HiTask1"
dependencies = New List(Of String)
children = New List(Of XFGanttTaskItem)
Dim task1 As New XFGanttTaskItem(taskName, taskTitle, taskDesc, wfStatusType, imageSource,
imageNameOrPath, taskStartTime, taskEndTime, taskDeadline, isMilestone, percentComplete,
isHighlighted, taskParameters, children, dependencies)
```

## System Extender Business Rules

This feature will be made available in a future release.

See "System Business Rules" on page 894 in System Tools.

## Direct Load Business Rules

### Special Workflow Rule BRAPI Functions

The Direct Load Workflow includes supporting BRAPI's to aid in programmatical Business Rule development.

### Determine Workflow Types

Boolean to determine the Workflow Type

- BI Blend - BRApi.Workflow.General.IsBiBlendWorkflow(si, wfClusterPk)
- Direct Workflow - BRApi.Workflow.General.IsDirectLoadWorkflow(si, wfClusterPk)

### Information Tables

Retrieve details from BI Blend's StageBiBlendInformation and Direct Load's StageDirectLoadInformation tables

- As StageBiBlendInfo = BRApi.Import.Data.GetBiBlendInfo(si, wfClusterPk)
- As StageDirectLoadInfo = BRApi.Import.Data.GetDirectLoadInfo(si, wfClusterPk)

### Error Management

Details for Transformation and Intersection errors relative to the current Direct Workflow process. Direct Load's in-memory processing only supports 1000 errors per load/import.

- As DataTable = BRApi.Import.Data.GetDirectLoadTransformationErrors(si, wfClusterPk)
- As DataTable = BRApi.Import.Data.GetDirectLoadIntersectionErrors(si, wfClusterPk)

### Summarized Target Data

The Direct Load Workflow has two settings to manage summarizing Stage records as Row or Blob. The Blob method does not physically write records to the StageSummaryTarget Table. These BRAPI's are built to automatically determine the storage method and retrieve the records.

- As DataTable = BRApi.Import.Data.ReadSummaryTargetDataTable(si, wfClusterPk)
- As DataTable = BRApi.Import.Data.ReadSummaryTargetDataTableTimeRange(si, wfClusterPk, cubeStartTimeld, cubeEndTimeld)

# Data Management Automation With PowerShell

PowerShell is an object-oriented programming language and interactive command line shell for Microsoft Windows. It was designed to automate system tasks, such as batch processing and create systems management tools for common processes. PowerShell includes over 130 standard command line tools for functions that formerly required users to create scripts in VB, VBScript or C#.

PowerShell offers a variety of ways to automate tasks which include:

- Cmdlets : Very small .NET classes that appear as system commands.
- Scripts: Combinations of cmdlets and associated logic.
- Executables: Standalone tools.

## Instantiation of Standard .NET Classes

PowerShell integrates with the .NET environment and can be embedded in other applications. Over one hundred cmdlets are included and can be used separately or combined to automate more complex tasks. You can also create and share cmdlets.

PowerShell is built into Windows Server 2008 and Windows 7, provided as an optional feature during installation. You can use Windows Task Scheduler to automate PowerShell script execution.

## Using PowerShell Script Editor

To run PowerShell on Windows 7 or a later version:

- Select Windows > Start, then enter "PowerShell".
- Select All Programs > Accessories |Windows PowerShell.

You can use these programs with PowerShell:

- Windows PowerShell ISE: An integrated script editor you can use to type PowerShell commands and to edit and run PowerShell script files. These are text files with a ps1 extension.
- Windows PowerShell: A command line execution tool similar to a DOS prompt. You can use this tool to run commands or script files but you cannot create or modify scripts.

## Configuring PowerShell for the Client API

Perform these steps on each machine that will use PowerShell before it can be used with the Client API:

1. Execute a PowerShell command enabling the execution of unsigned scripts.
2. Create or alter the PowerShell execution and IDE configuration files, so the script engine understands how to use the .Net Framework v4.0.
3. Install OneStream Studio on each machine executing PowerShell scripts.

## Running Unsigned Scripts

Initially, this code needs to run in a PowerShell command prompt to enable PowerShell to run unsigned scripts created on the local computer:

```
set-executionpolicy remotesigned
```

## Configuration for .Net Framework v.4.0

To use the OneStreamClientApi with PowerShell, PowerShell must be configured to use the .NET Framework v4.0. Perform these tasks:

1. Modify or create these files if they do not exist in C:\Windows\System32\WindowsPowerShell\v1.0:

- powershell.exe.config
- powershell\_ise.exe.config

2. Add this required content to each file:

```
<?xml version="1.0"?>
<configuration>
    <startup useLegacyV2RuntimeActivationPolicy="true">
        <supportedRuntime version="v4.0.30319"/>
        <supportedRuntime version="v2.0.50727"/>
    </startup>
</configuration>
```

For more information, see: <http://tfi09.blogspot.com/2010/08/using-newer-versions-of-net-with.html>.

3. Install OneStream Studio.

## Learning PowerShell

Microsoft provides extensive resources to help IT professionals leverage PowerShell. For more information, see: <http://technet.microsoft.com/en-us/scriptcenter/powershell.aspx>

## Using the Client API in a PowerShell Script

OneStreamClientApi specifically enables PowerShell scripts to call a function. This API exposes functions for authentication and Data Management. OneStream expanded the number of functions exposed to this API. The Client API component is installed with the Client API Installer. This API offers a simple set of functions enabling script writers to connect to the server, authenticate, execute OneStream Data Management Sequences, and perform basic data retrieval.

Client API Object Hierarchy

OneStreamClientAPI

LogonInfo

Type

LogonInfo

SI

Type

SessionInfo

Authentication

Logon

Parameters

string webServerUrl

string userName

string password

XFClientAuthenticationType clientAuthenticationType

Return Value

LogonInfo

Logoff

Parameters

None

Return Value

None

OpenApplication

## Implementing Security

---

Parameters

string Application

Return Value

LogonInfo

LogonAndOpenApplication

Parameters

string webServerUrl

string userName

string password

string application

XFClientAuthenticationType clientAuthenticationType

Return Value

LogonInfo

EncryptPassword

Parameters

string clearTextPassword

XFClientAuthenticationType clientAuthenticationType

Return Value

string

DataManagement

ExecuteSequence

Parameters

string sequenceName

string customSubstVarsAsCommaSeparatedPairs

Return Value

DataMgmtResult

ExecuteStep

Parameters

string dataMgmtGroupName

string stepName

string customSubstVarsAsCommaSeparatedPairs

Return Value

DataMgmtResult

DataProvider

GetAdoDataSetForCubeViewCommand

Parameters

```
string CubeViewName  
bool dataTablePerCubeViewRow  
CubeViewDataTableOptions dataTableOptions  
string resultDataTableName  
Dictionary<string, string> customSubstVars  
bool throwExceptionOnError
```

Return Value

DataSet

GetAdoDataSetForMethodCommand

Parameters

```
XFCommandMethodType xfCommandMethodType  
string methodQuery  
string resultDataTableName  
Dictionary<string, string> customSubstVars  
bool throwExceptionOnError
```

Return Value

DataSet

# Exposing Data Management Automation through OneStream Web API

OneStream Web API is a RESTful web service designed to expose OneStream Data Automation functions when interacting with third-party API client applications.

OneStream Web API must be installed on a web server. It also must be configured for external authentication providers supporting OAuth2.0/OpenID Connect authorization protocol. Identity Providers currently supported are Okta, Azure AD and PingFederate.

OneStream Web API is API client agnostic. It accepts and outputs data in JSON format making it possible for every API client application that supports this format to also interact with this service. One of the most widely used API clients is Postman, a Windows app. For more information about how to configure OneStreamWeb API to interact with Postman see the autogenerated documentation at [http\(s\)://\[servername\]:\[port\]/onestreamapi](http://[servername]:[port]/onestreamapi).

# OneStream Web API endpoints:

Authentication endpoint. Represents a RESTful service for Authentication.

- POST api/Authentication/LogonAndReturnCookie

Used primarily by the Enablement Team to verify Web API installation completed successfully. Returns a one-time cookie value that holds authentication state or a message indicating failure along with a proper HTTP code.

DataManagement endpoint. Represents a RESTful service of Data Management.

- POST api/DataManagement/ExecuteSequence:

Executes a Data Management Sequence and returns a success/failure message along with a proper HTTP code.

- POST api/DataManagement/ExecuteStep

Executes a Data management Step and returns a success/failure message along with a proper HTTP code.

DataProvider endpoint. Represents a RESTful service of Data Provider

- POST api/DataProvider/GetAdoDataSetForAdapter

Returns a JSON representation of a DataSet a given Dashboard Adapter or a failure message along with a proper HTTP code.

- POST api/DataProvider/GetAdoDataSetForCubeViewCommand

Returns a JSON representation of a DataSet for a given Cube View or a failure message along with a proper HTTP code.

- POST api/DataProvider/GetAdoDataSetForSqlCommand

Returns a JSON representation of a DataSet for a given SQL Query or a failure message along with a proper HTTP code. Administrator role is required for this functionality.

- POST api/DataProvider/GetAdoDataSetForMethodCommand

Returns a JSON representation of a DataSet for a given pre-defined list of method commands used by XFDataProvider to fill a DataSet or a failure message along with a proper HTTP code. Administrator role is required for this functionality.

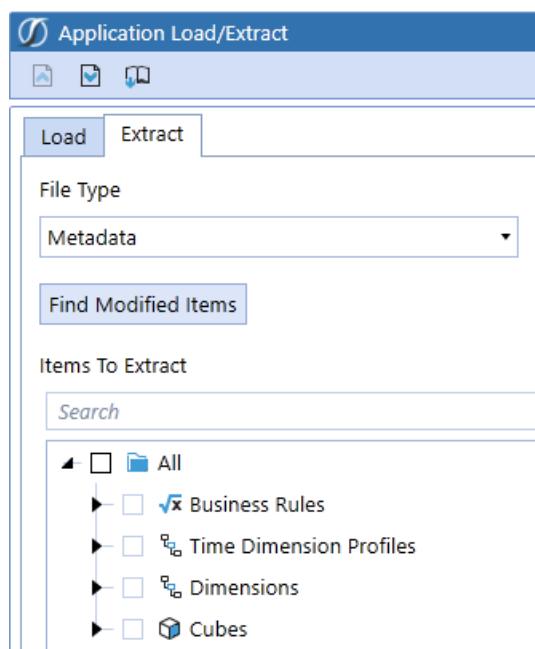
## Extracting and Loading Dimensions

The Application Load/Extract can find Dimension changes and export them into an xml in order to make the metadata migration process easier.

You can extract applications as an XML or zip file. If the file size is larger than 2Gb, you must use a zip file.

When importing, you may have extracted an XML file that was larger than 2Gb in size. You must do a zip extract to successfully load the file.

**NOTE:** This feature only applies to Dimensions.



## Extracting Dimension Changes

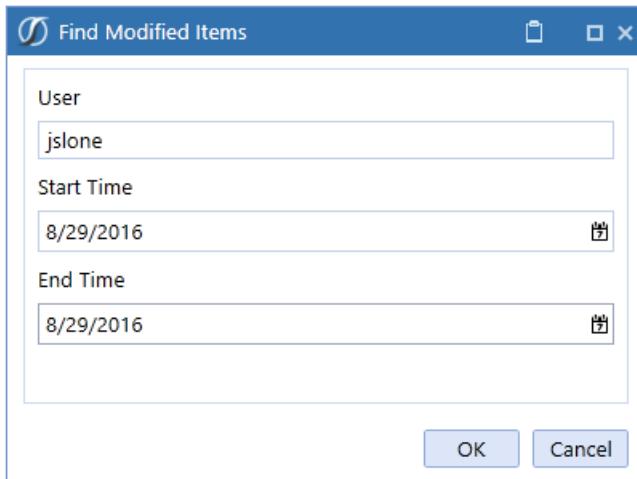
**Find Modified Items**

Modified Items can consist of adding/deleting Dimension Members and/or adding/deleting Dimension Relationships. Click Find Modified Items to launch the dialog. This allows users to find Dimension changes by time and user.

## Implementing Security

---

1. Enter a user name and a range of time in order to find Dimension Member and Dimension Member Relationship changes. Leave the User field empty in order to find changes by all users.



For example, in the application's Dimension Library, a Member was created, another Member was deleted, and a relationship was deleted. Once the user and time filters are entered, click **OK**, and these changes will be highlighted.

The hierarchy will indicate where changes were made. The example below indicates there was a partial change to the Dimensions.

A screenshot of a software interface titled "Items To Extract". At the top is a search bar with "Search" placeholder text. Below it is a tree view of items:

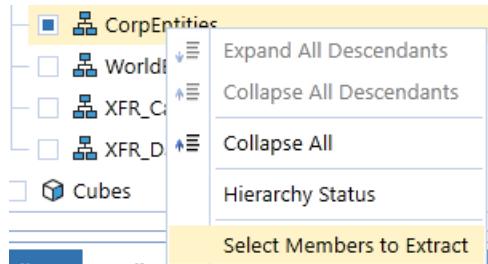
- All
  - Business Rules
  - Time Dimension Profiles
    - Dimensions
  - Cubes

The "Dimensions" node under "Time Dimension Profiles" is highlighted with a red rectangular box.

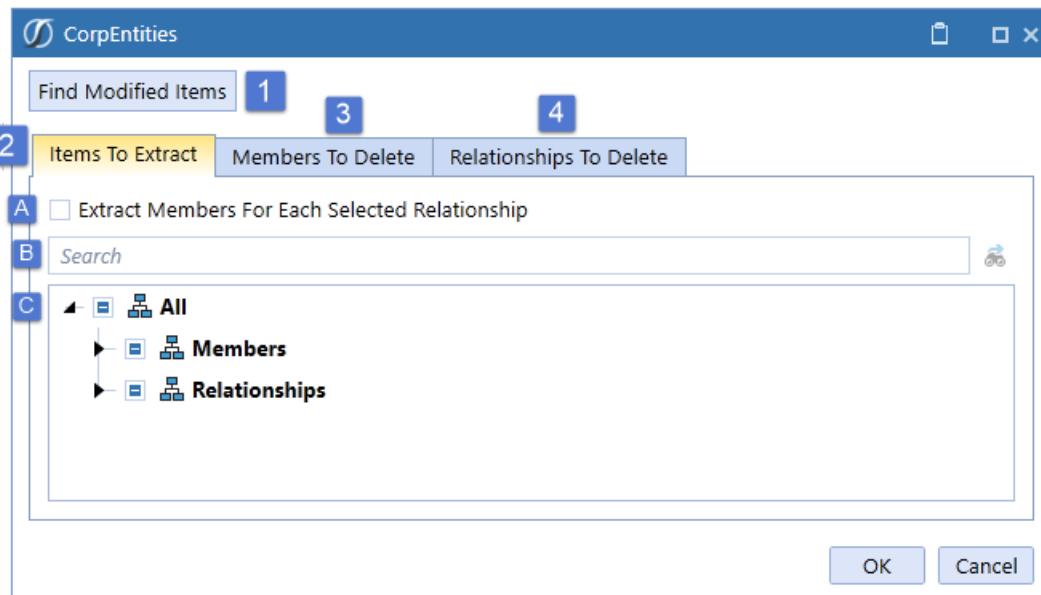
2. Expand this to see what Dimensions were changed.



3. Right-click on the Dimension name and click **Select Members to Extract**.



This launches an extraction dialog for the specific Dimension selected.



4. Find Modified Items

By default, everything is selected. Click Find Modified Items and enter the username and desired date range in order to find the specific changes made to the selected Dimension.

5. Items to Extract Tab

This tab displays the changes found in the hierarchy. Users can also manually select Members and Relationships to extract.

6. Extract Members for Each Selected Relationship Check Box

Check this box in order to include the Members used in the selected relationships.

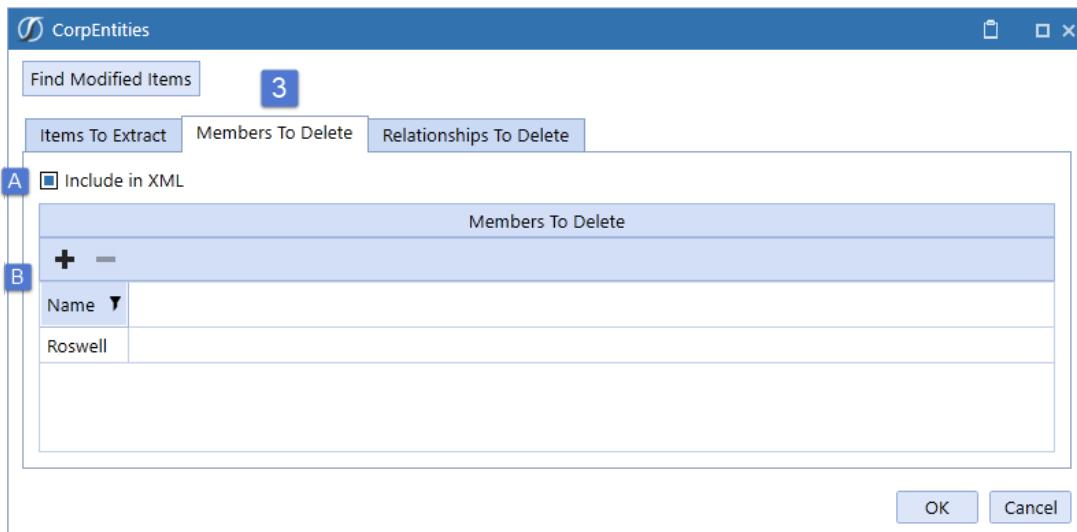
### 7. Search

Search for a specific Member in the selected Dimension.

### 8. Dimension Hierarchy

Scroll through the Dimension hierarchies in order to see where changes occurred and/or manually select or de-select Members or Relationships to extract.

**NOTE:** Right-click on any Parent Member under Relationships and click Select All Descendants in order to select all Child Members within the hierarchy and include them in the extraction.



### 9. Members to Delete Tab

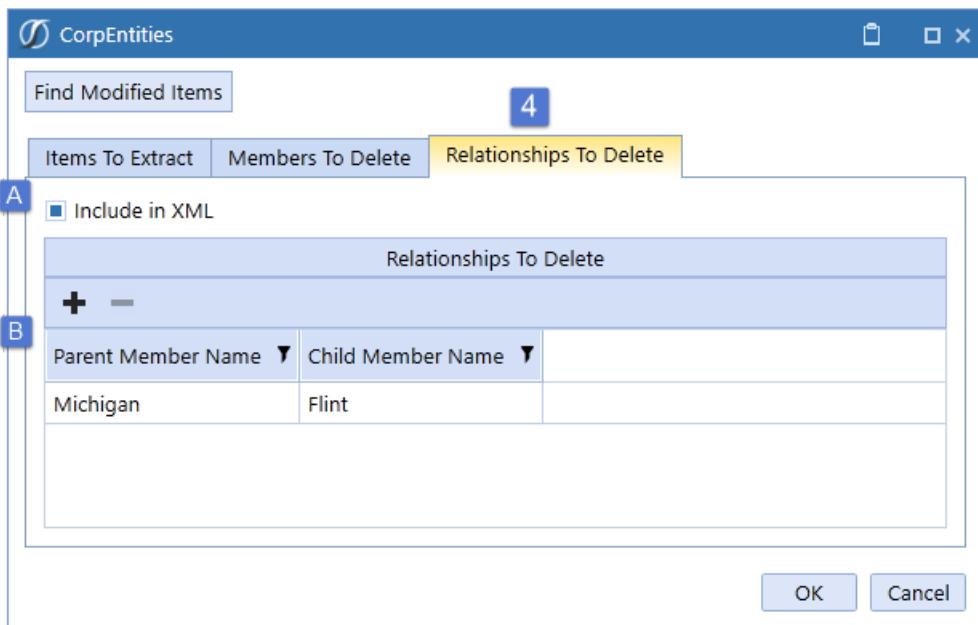
The Members to Delete tab displays any deleted Members found within the User/Time Parameters.

### 10. Include in XML Check Box

By default, Include in XML is not checked. Check this box in order to include the displayed Members in the extract. If checked, any Members to Delete will have an Action=Delete in the xml file and be deleted upon loading the xml into another application.

### 11. Any deleted Members found within the User/Time Parameters will display here. Click the plus sign to manually add additional Members to delete. To exclude a particular entry from the xml, select the line to exclude and click the minus sign.

**NOTE:** Members with stored data cannot be deleted from an application.



### 12. Relationships to Delete

The Relationships to Delete Tab displays any deleted relationships found within the User/Time Parameters.

### 13. Include in XML Check Box

Check this box in order to include the displayed Member relationships in the extract. If checked, any Relationships to Delete will have an Action=Delete in the xml file and be deleted upon loading the xml into another application.

### 14. Users can manually add additional relationships to delete by clicking the plus sign and indicating the Parent and Child. To exclude a particular entry from the xml, select the line to exclude and click the minus sign.

## Loading the XML File

When importing an extracted xml file, it processes in the following order:

1. New or changed Members
2. Deleted Members

3. New Relationships
4. Deleted Relationships

**NOTE:** The processing order is important because if any errors occur during the xml file load, OneStream applies as many modifications as it can up to the point of the error. For example, an xml file contains ten Member changes, three deleted Members and one new relationship. If an error occurs when trying to delete the first Member, the ten Member changes will still take place because they are processed first during the xml load. Any modifications in the xml prior to the error will occur and any after the error will not.

If a user receives an error during the load process, the error must be resolved in order to complete the metadata migration.

## Common Load Errors and Resolutions

### 1. Deleting a Member with data

If a Member without data is deleted from the source application, but that same Member contains data in the destination application, an error occurs and the Member will not be deleted. To resolve the error, do one of the following:

- a. Clear the Member's data in the destination application and reload the xml file.
- b. Create a new xml extract excluding that Member from the file.
- c. Edit the xml file to exclude the Member and the action.

### 2. Deleting a Member without Data

Care should be taken when deleting Entity members. Even though the member may not have data, it may be in use as an Intercompany Partner. Ensure the member is not in use on any data records, or on records as an Intercompany Partner.

### 3. Undefined Security Group

If a security group is assigned to a Dimension Member in the source application, but does not exist in the destination application, an error will occur. To resolve the error, do one of the following:

- a. Create the security group in the destination application and reload the xml file.

b. Create a new xml extract excluding this Member and its changes from the file.

c. Edit the xml file to exclude the Member and the action.

### 4. Undefined FX Rate Type

If a FX Rate Type is assigned to a Scenario Member in the source application, but does not exist in the destination application, an error will occur. To resolve the error, do one of the following:

a. Create the FX Rate Type in the destination application and reload the file.

b. Create a new xml extract excluding this Member and its changes from the file.

c. Edit the xml file to exclude the Member and the action.

### 5. Invalid Characters in the XML File

If the xml file was edited and invalid characters were entered for a Member name, an error will occur. To resolve the error, do one of the following:

a. Make the Member modifications in the source application and extract an xml file without invalid characters.

b. Edit the xml file and remove the invalid characters.

## Project Extract and Load

This extract is for Application Project Designers who are building solutions that span many artifacts, such as Dashboard Maintenance Units, Business Rules, Cubes, Dimensions, Cube Views, etc. A good example is a person designing a solution to be hosted on MarketPlace. This application Extract and Load option allows all the defined objects, such as Dashboards and Business Rules, to be collected as a single file export package and to be later reloaded as a package. XFProject is used as a convenient way to organize MarketPlace or similar solutions into a folder structure which can be integrated with a version control system such as "Git." Doing so could enable more than one team member to work on a solution simultaneously. The developer must create an XML file which is the definition for the contents of the Project export.

## Project File

The Application Designer must first manually define an XML file to support the export of objects as a Project File. The file is saved with the file extension of .xfProj and saved to a local project folder which could also support a version control system.

Sample File:

## Implementing Security

---

```
7 <xfProject topFolderPath="" defaultZipFileName="">
8   <projectItems>
9     <projectItem itemType="DashboardMaintenanceUnit" folderPath="" name="Corporate" includeDescendants="true" />
10    <projectItem itemType="DashboardMaintenanceUnit" folderPath="" name="Corporate Templates" includeDescendants="true" />
11    <projectItem itemType="DashboardComponent" folderPath="" name="btnSaveAndCalculate" includeDescendants="true" />
12    <projectItem itemType="BusinessRule" folderPath="" name="FXRates" includeDescendants="true" />
13  </projectItems>
14 </xfProject>
```

### File Structure:

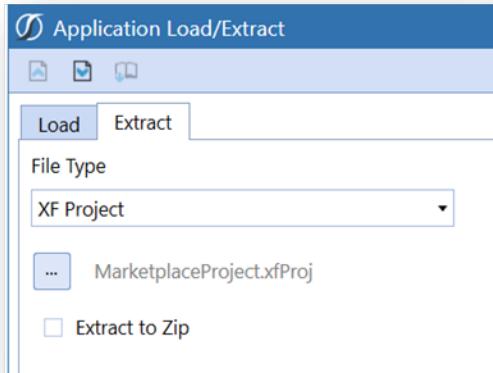
- xfProject: The root node to start a .xfProj which contains two attributes:
  - TopFolderPath: Will create and define the starting folder location of where the specified files are extracted to.
  - DefaultZipFileName: Will create a standard default file name for .zip file extracts.
- projectItems: A list structure containing the project items needed to extract (no attributes needed).
- projectItem: The item reflecting what is needed to extract from OneStream or load from the file system. It has 4 attributes:
  - ProjectItemType:
    - BusinessRule
    - Cube
    - CubeViewGroup
    - CubeView
    - CubeViewProfile
    - DashboardMaintenanceUnit
    - DashboardFile
    - DashboardString
    - DashboardParameter
    - DashboardGroup

- DashboardAdapter
- DashboardComponent
- Dashboard
- DashboardProfile
- DataManagementGroup
- DataManagementStep
- DataManagementSequence
- DataManagementProfile
- DataSource
- Dimension
- TransformationRuleGroup
- TransformationRuleProfile
- FolderPath: The name of the sub folder where the project item type is extracted to.
- Name: The name of the project item.
- IncludeDescendants: Default is true and only affects these project item types:
  - CubeViewGroup
  - DashboardGroup
  - DashboardMaintenanceUnit
  - DataManagementGroup

## File Extract

The .xfProj file is placed in a local folder, such as the user's desktop. The defined folderpath folders will be generated here as the target location for application exports and loads. There are two file extract options available on the Windows App.

- .zip: The export option will collect all the objects defined in the .xfProj file as a Zip file to the location of the .xfproj file.
- File: The standard export will export all the objects defined in the .xfProj file to the folderpath locations defined in .xfproj file.



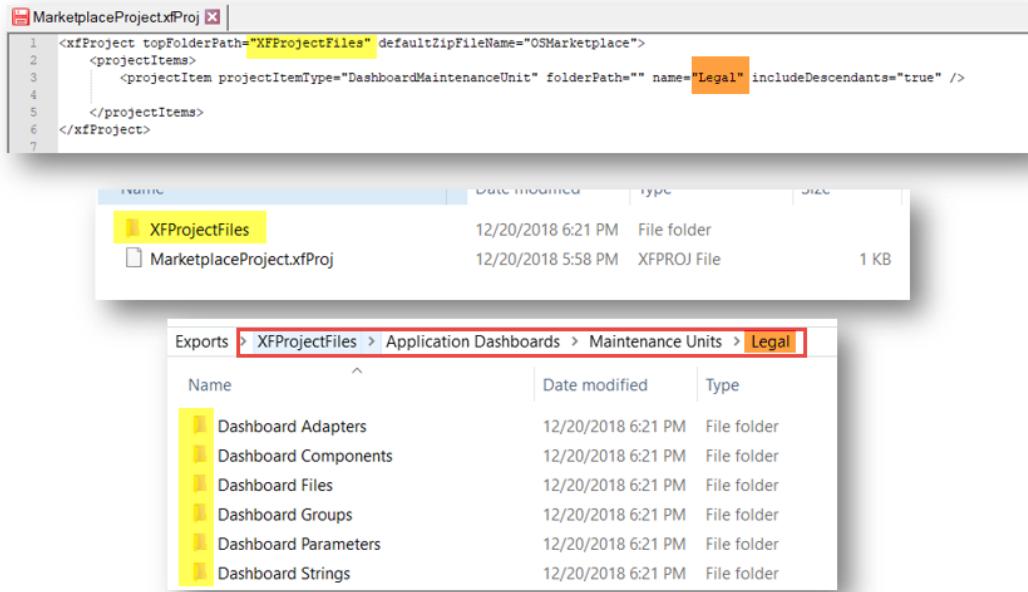
1. Navigate to **Application > Load > Extract**.
2. Select the **Extract** tab.
3. Browse and select the **.xfProj** file.
4. (Optional) From the OneStream Windows Client, select or de-select **Extract to Zip** as required.
5. Click the **Extract toolbar** button.

Example:

The contents will be generated in the defined folder paths.

## Implementing Security

---

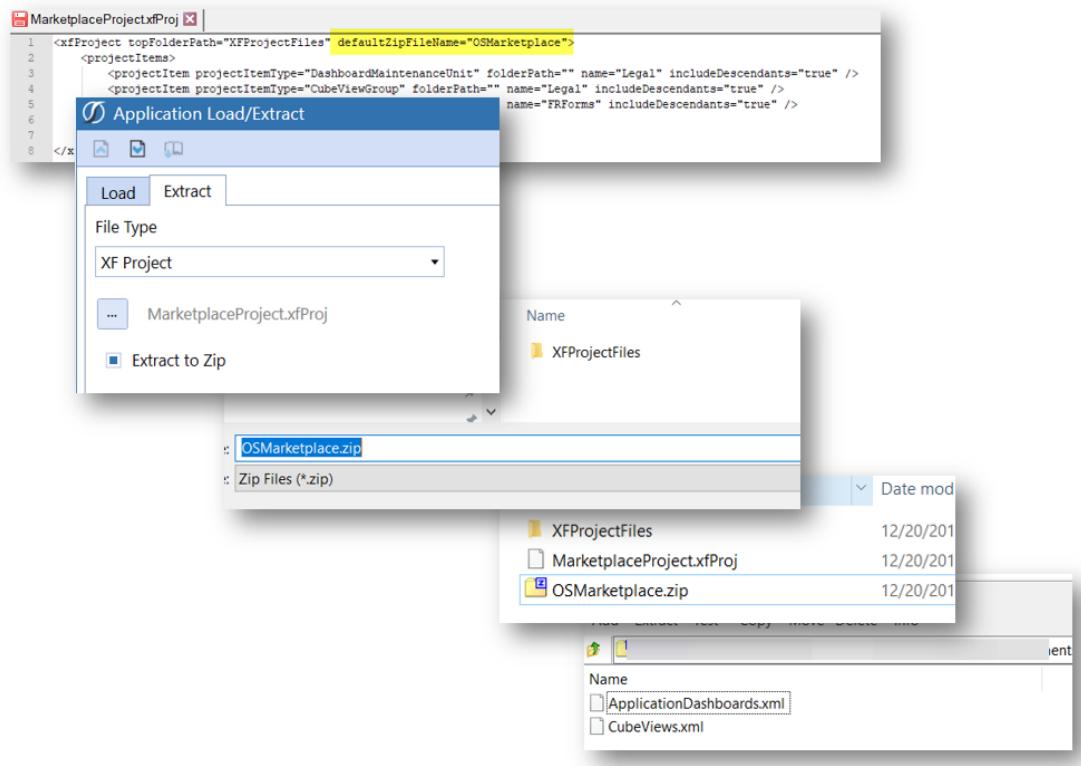


## Zip Extract

The zip file extract will create an application zip file containing all the objects defined.

## Implementing Security

---

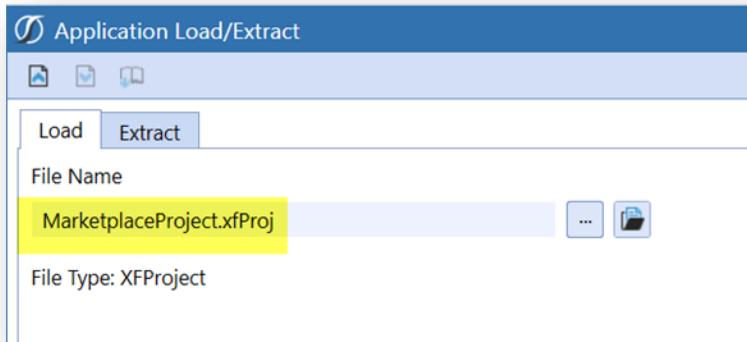


## File Load .xfProj

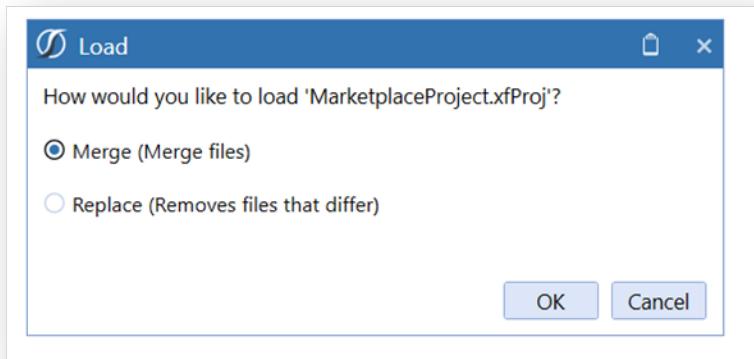
The file load using the defined .xfProj file provides a seamless link to the project files. When loading an .xfProj file, the user is presented with options to “Merge” or “Replace” the target files. The only files affected are those defined by the .xfproj file.

**NOTE:** If you select **Replace**, it will only remove files that differ for CubeViewGroups, DataManagementGroups, DashboardMaintenanceUnits, and DashboardGroups. For all other items (such as, business rules or extensibility rules), if you select **Replace** it will act as a Merge.

1. Navigate to **Application > Load > Extract**.
2. Click the **Load** tab.
3. Browse and select the **.xfProj** file.
4. Click the **Load toolbar** button.



5. Select the Load Method, Merge or Replace.



## Zip Load

The zip file load functions as any other application file load. The contents of the file are merged into the application. The zip file load is not supported by alternative merge or replace file load options.

# Cubes

Cubes are organization structures that contain data. They control how data is stored, calculated, translated, and consolidated based on dimensions assigned to the cube. While flexible and designed to hold multiple types of data, they are generally designed for specific purposes. An application can have several Cubes that share Dimensions, time profiles, business rule functions, and data. In this section, you will learn about cube dimension data, time profiles, and other cube-specific characteristics.

## Dimensions

There are three types of Dimensions available: Customizable, Derived and System Dimensions.

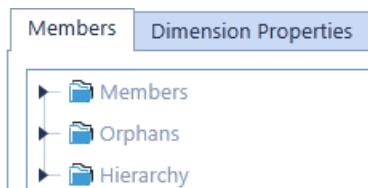
Customizable Dimensions have no preset Members and can be free form. These include Entity, Scenario, Account, Flow, and UD1...UD8 (also known as User Defined Dimensions 1 through 8).

Derived Dimensions are a result of being flagged by a setting in another Dimension. Intercompany is the only Dimension that can be derived. It is made of a None Member and a Member for every Entity that is marked with IsIC = True.

System Dimensions are non-customizable Dimensions and cannot be changed. They are pre-defined as part of the system. These include Consolidation, Time, View and Origin.

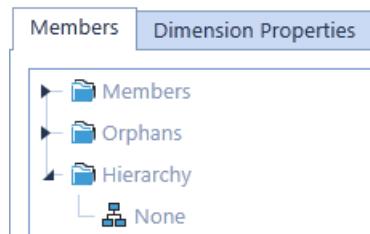
Dimensions can be viewed by Members, Orphans, or in a Hierarchy. All Members will appear in

an alpha-numerical list. Use the search button  to search for specific Members. This search will also produce every hierarchy in which a Member appears. It is possible for different settings to be set based on each Parental roll up. Orphans are a list of Members not assigned to any Dimension hierarchy.



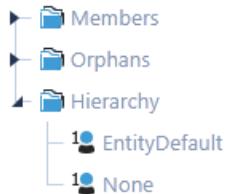
### Root Dimensions

Every Dimension hierarchy has a None Member. This means there is not a selection for this Dimension or it is not applicable. For example, a setting of None in the Intercompany Dimension would be appropriate for an account entry of cash.



### Entity Default

Each User Defined Dimension has an additional Member called EntityDefault used to assign attributes to an Entity. This is set on the specific Entity in the Vary by Cube Type settings for each of the User Defined Dimensions. This allows an Entity to have a specific default and reduce the need to map every Entity to a common tag such as Region or Division as it pertains to data import and form-based data entry. A user can select the EntityDefault Member without knowing the specific Entity setting for each of these UD Dimensions. However, using this setting has a slight impact on consolidation time because there will be more intersections in the financial model.



The Dimension Library allows for all the customizable Dimensions to be defined for your business needs. Dimensions can be shared across multiple Cubes. The defining of Dimensions is extremely important in order to allow sharing between Cubes.

### **UD1 - UD8 Default**

A default Membership for this Entity can be defined to a User Defined Dimension such as Region. For example, if the Entity falls in the U.S. Region every import, form edit, and journal entry classifies this Entity as part of the U.S. Region. The Entity never has to explicitly map to the Region. When data is Validated by the Staging Engine, the Entity's Transformation rule will have set the target as EntityDefault. When using EntityDefault to define the data load to a UD member, all the related source members must use the Transformation Rule target as EntityDefault to properly aggregate the records. As the data records transition from stage to cube during the load process, the EntityDefault target member will resolve to the member definition on the Entity. There can be a negative impact on the consolidation if it is not used in a User Defined Dimension with a limited number of Members.

### **UD1 – UD8 Constraints**

This is the user defined constraint. The Entity can only use the members with this child or members under the selected parent member.

## **Restricted Characters**

There are certain restricted characters in Dimensions, Members, Dashboards, that can not be named with the following characters:

- /
- |
- !
- @
- %
- #
- ,
- ;
- ^
- \*
- +

## Cubes

---

- -
- =
- \
- ?
- < (if using member in Extensible Documents)
- >
- "
- [
- ]
- {
- }
- &

## Reserved Words

These reserved words cannot be used on structural application components, like Cubes and Dimensions. We recommend avoiding these reserved words in the application.

- Account
- All
- Cons
- Consolidation
- Default
- DimType
- Entity
- EntityDefault

## Cubes

---

- Flow Origin
- IC
- None
- Parent
- POV
- Root
- RootAccountDim
- RootEntityDim
- RootFlowDim
- RootScenarioDim
- RootUD1Dim
- RootUD2Dim
- RootUD3Dim
- RootUD4Dim
- RootUD5Dim
- RootUD6Dim
- RootUD7Dim
- RootUD8Dim
- Scenario
- Time
- UD1 – UD8
- UD1Default
- Unknown

- View
- WF
- Workflow
- XFCCommon

## Dimension Library Toolbar and Right-Click Options



### Create Dimension

Use this to create a new Dimension



### Delete Selected Dimension

Use this to delete the selected Dimension



### Save Dimension

Use this to save changes to the selected Dimension



### Rename Dimension

Use this to change the name of the Dimension.



### Move Dimension

Use this to move the dimension. You can move a dimension up one level above a parent or below a sibling. If the move is invalid, you will not be able to move it. Integrity of Member relationships is evaluated and enforced.

**NOTE:** Supported dimensions are Account, Scenario, Flow, and UDs.



### Create Member

Use this to create a new Member under the selected Dimension

**NOTE:** Maximum Member Name Length is 500 characters



### Delete Selected Member

Use this to delete a selected Member

## Cubes

---



### Rename Selected Member

Use this to rename a selected Member



### Save Member

Use this to save changes to a selected Member



### Add Relationship for Selected Member

Use this to move a Member from one hierarchy to another, or to place a Member in more than one hierarchy.



### Remove Selected Relationship without Deleting Member

Use this to remove a Member from a hierarchy without deleting the selected Member. This can also be done by right-clicking on a Dimension Member.



### Cancel All Changes Since Last Save

Use this to undo and unsaved changes.



### Search Hierarchy

Use this to search Member hierarchies within a Dimension.



### Collapse Hierarchy

Use this to collapse a Member hierarchy within a Dimension.

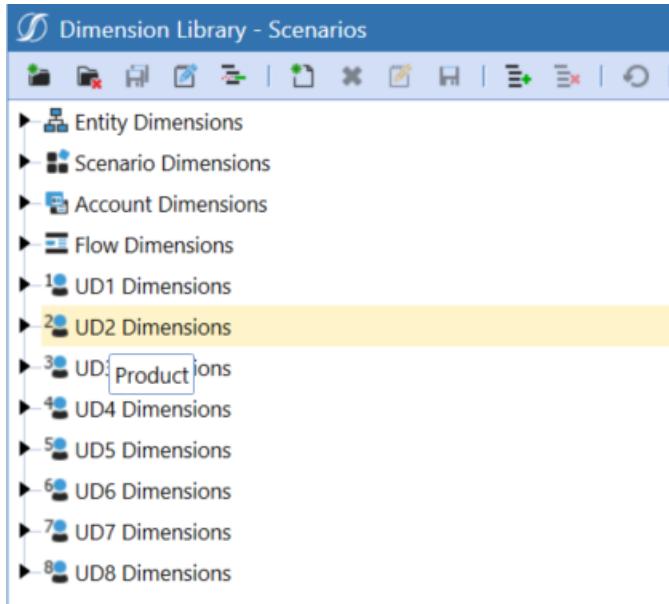


### Navigate to Security

This icon appears in all Dimension Security properties and when clicked it navigates to the Security screen. This is an easy way to make changes to Security Users or Groups before assigning them to specific Dimensions.

### User Defined Description – Dimension Library

Navigating within the dimension Library, hovering over the Dimension Types will display the User Defined description as a tool tip. See Application Properties and then User Defined Dimensions (Descriptions) for more information.



## Dimension Member Right-Click Options

The following options are available when a user right clicks on any Dimension Member:

### Clone Member

This creates a new Member using the same settings as the selected Member. New Members can be cloned and positioned as a First Child, Last Child, Previous Sibling, or Next Sibling of the selected Member.

Note: Formula Types and Formulas do not copy to the new Member.

### Delete Member

This deletes the selected Member from the Dimension Library.

### Copy Selected Members

Select or multi-select (Ctrl+Click) Members in order to copy them and Paste Relationships.

### Paste Relationship (Add)

Select the desired Member and use this to add the copied Member(s) to a new hierarchy. The copied Members can be added as a First Child, Last Child, Previous Sibling, or Next Sibling of the selected Member. This selection creates additional relationships for the copied Members.

### Paste Relationships (Move)

Select the desired Member and use this to move the copied Member(s) to a new hierarchy. The copied Members can be added as a First Child, Last Child, Previous Sibling, or Next Sibling of the selected Member. This will remove the Members from their current relationship.

## Cubes

---

### Remove Relationships

Use this to remove the copied Member(s) from their current relationship without moving them to a new one. If the copied Member is no longer a part of the Dimension structure, it will be placed under Orphans.

### Expand All Descendants

Use this to expand all the descendants for the selected Member.

## Dimension Grid View

The grid view can be very beneficial when changing the settings of numerous Members within a Dimension. Using Member Filters, a user can create a list of Members, and then choose the metadata settings that need to be changed.

The screenshot shows a software interface for managing dimension members. At the top, there are three tabs: 'Members' (selected), 'Dimension Properties', and 'Grid View'. Below the tabs is a search bar containing the query 'A#[Income Statement].TreeDescendantsInclusive'. Underneath the search bar is a toolbar with icons for filter, sort, and refresh. A blue header bar allows dragging a column header to group by that column. The main area is a grid table with the following data:

Name	AccountType	FormulaType	FlowConstraint	UD1Constraint	UD2Constraint
Income Statement	Group	(Not Used)	Root	Root	Root
69000	Revenue	(Not Used)	Root	Root	Root
64000	Revenue	(Not Used)	Root	Root	Root
63000	Revenue	(Not Used)	Root	Root	Root
62000	Revenue	(Not Used)	Root	Root	Root
61000	Revenue	(Not Used)	Root	Root	Root
60999	Revenue	(Not Used)	Root	Root	Root
60000	Revenue	(Not Used)	Root	Root	Root

### Member Filter

Builds a list to see specific Members. See Member Filter Builder Dialog for more details.

### Grid Settings

#### Cube Type

Members can change based on Cube Type. Specifies the Members to look at in the grid view.

#### Scenario Type

Members can change based on Scenario Type. Specifies the Members to look at in the grid view.

### Time

Members can change based on the Time Member. Settings can be turned on or off and the formulas will change in order to look at a specific time frame.

### Grid Columns to Display

Select the columns of metadata to view in the grid.

## Entity Dimension

The Entity Dimension is different from all others. In a multi-Cube application, the Entity Dimension links everything together.

## General Member Properties

These properties are standard across all Dimensions.

### Dimension Type

This indicates what Dimension is currently being used (e.g., Entity).

### Dimension

This indicates the Dimension name (e.g., Houston).

### Member Dimension

The Dimension to which it is a Member (e.g., HoustonEntity Dimension).

### Name

The name of the Member in the Dimension (e.g. Houston Heights).

### Default Description

A description of the Member in the Dimension. Refer to Report Alias descriptions for Members.

## Security

### Display Member Group

This group can see that this Entity exists within a list of Entities.

### Read Data Group

This group can see data from this Entity.

### Read Data Group 2

This is a second group that can see data from this Entity. It is used for additional security granularity

### Read and Write Data Group

This group can see data from this Entity and make changes to it.

### Read and Write Data Group 2

This is a second group that can see data from this Entity and make changes to it. It is used for additional security granularity.

**NOTE:** Click  and begin typing the name of the Security Group in the blank field. As the first few letters are typed, the Groups are filtered making it easier to find and select the desired Group. Once the Group is selected, select CTRL and Double Click. This will enter the correct name into the appropriate field.

### Use Cube Data Access Security

If set to True, Cube Data Access Security will be applied from the Cube level down to this Entity, if set to False, Cube Data Access Security will not be applied.

### Cube Data Cell Access Categories/Cube Conditional Input Categories/

### Cube Data Management Access Categories

Category is used to specify an optional name for a group of Cube Data Access Items. Multiple items can use the same Category name. A comma separated list of Category names can be specified in Entity Properties in the Dimension Library when the data cell security for an Entity should use only a subset of the Cube Data Access Items.

The settings for these are used to specify a comma separated list of category names that will be processed for the Entity. If these settings are left empty, then all categories will be used in the corresponding settings on the Data Access tab of the Cube Administration page. See Data Access under "Cubes" on page 380 for more details.

## Settings

### Currency

The local currency of a particular Entity.

### Is Consolidated

If set to **True**, the data from this Entity's children is consolidated (i.e., this Entity will equal the total of its children).

If set to **False**, the data will not be consolidated. The use case to set the Is Consolidated setting to False is to use the Parent Entity strictly for grouping purposes. Also, by setting Is Consolidated to False helps with consolidation performance times because the consolidation will not be performed at the Parent Entity.

**Is IC Entity**

If set to True, this will make the Entity an Intercompany Entity. An Entity cannot post intercompany transactions to intercompany accounts if this option is True. This is only required for Base Entities where the intercompany intersections roll up and eliminate. If set to False, this will not be an Intercompany Entity.

## Vary by Cube Type

Constraints are Members with a restriction to only specific Members for an Entity or an Account. There can be different settings for each Cube Type because Dimensions can belong to multiple Cubes.

**Flow Constraint**

This is the Flow Dimension constraint. This Entity can only use the Members with this child or Member under a selected Parent Member.

**IC Constraint**

This is the Intercompany Dimension constraint. This Entity can only use the Members with this child or Member under a selected Parent Member. Setting Entity constraints will define the data intersection as a green no input cells.

**IC Member Filter**

This is an additional way to limit intercompany partners of a particular Entity. The IC Member Filter can make a list and the Entity can only have intercompany transactions with this list of partners, and those lists of partners are the only ones that can have a transaction with this Entity. This provides additional protection to the intercompany transaction.

**UD1 Constraint...UD8 Constraint**

This is the User Defined constraint. This Entity can only use the Members with this child or Members under the selected Parent Member.

**UD1 Default...UD8 Default**

A default Membership for this particular Entity can be defined to a User Defined Dimension such as Region. For example, this Entity falls in the U.S. Region and every import, form edit, and journal entry should classify this Entity as being part of the U.S. Region. The Entity never has to explicitly map to the Region. When data is loaded by the Staging Engine, it is directed to the EntityDefault Member and the User Defined setting here will be applied automatically. This can have a negative impact on the consolidation if not used in a User Defined Dimension with a limited number of Members.

## Vary by Scenario Type

### **Sibling Consolidation Pass**

This is typically used for Holding Companies related to Equity Pickup calculations. Specify Pass 2 or greater if calculated data for this Entity is based on calculated data from other sibling Entities during a consolidation. It allows this Entity to be consolidated and calculated after other sibling Entities have been consolidated and calculated. The default behavior for all Entities is either Pass 1 or (Use Default) (both settings do the same thing) which effectively cause all sibling Entities to be consolidated (if they have Child Entities) and calculated at the same time or in an indeterminate order.

### **Sibling Repeat Calculation Pass**

This is typically used for Circular Ownership related to Equity Pickup calculations. Specify Pass 1 or greater to repeat the calculation of this Entity's Local Consolidation Member. Repeat Calculation Passes occur after all sibling Entities have been consolidated and calculated. Repeat calculations are used when calculated data for two Entities rely on each other's calculated data. The default behavior for all Entities is (Use Default) which will not use a repeat calculation.

### **Auto Translation Currencies**

This is typically used for an Equity Pickup calculation when an Entity needs to be translated to a sibling Holding Company's local currency during a consolidation. Enter a comma separated list of currencies. The default behavior is to translate only to the parent Entity's local currency during a consolidation.

See Equity Pickup in "About the Financial Model" on page 2 for more details on this feature.

## Vary by Scenario Type and Time

During the course of normal business, Entities can have different attributes based on Scenario and Time. There are default settings applied from the first time period in the application until there is a change, and there are settings that vary by Scenario Type. The time display for any time varying properties uses the Standard Time Dimension Profile.

### **In Use**

If set to True, the Entity is in use, if set to False, this can turn off the ability to use an Entity based on Time. This keeps historical data available. This is designed to be used when an Entity becomes inactive or is sold. Once an Entity is no longer in use, it will be ignored during consolidation and all intersections including this Entity will be invalid.

## Cubes

---

### Allow Adjustments

If set to True (default), the Journals Module is enabled for the Entity to enter adjustments to the AdjInput Origin member. When set to False, the Journals Module is disabled for the Entity. However, when False, adjustment to the AdjInput Origin member is still allowed on Accounts having the Account Adjustment Type of “Data Entry” which is used in designs where adjustments are performed using Form data entry rather than the Journals Module. To prevent input to AdjInput on Accounts set as “Data Entry” Adjustment Type, NoInput Rules or Data Cell Conditional security can be used. This setting can be applied as True/False a default or it can vary by Scenario Type and/or Time.

### Allow Adjustments From Children

This setting is applied on parent level Entities to allow the direct child Entities to post journal adjustments using the Consolidation dimension members OwnerPostAdj and OwnerPreAdj. If set to True (default), adjustments from the direct child Entities to OwnerPostAdj and OwnerPreAdj is allowed. This setting can be applied as True/False a default or it can vary by Scenario Type and/or Time.

### Text1...Text8

This is open for custom attributes used for multiple purposes such as Business Rules, Member Filters or Transformation Rules. The value can be changed in the tag over time as the business changes or by Scenario Type.

## General Relationship Properties

While Dimensions, like Accounts or Flow, contain Members which can be inherited and extended through Extensible Dimensionality ®, Entities are referenced across Entity Dimensions. This means that a reference of an Entity, either within the same Dimension or another, can be created.

This is done with the Add Relationship for Selected Member  option within the Entity Dimension. In addition, this icon can be used when a Member must be moved out of its current hierarchy and inserted into another one.

These Settings are standard across all Dimensions.

### Dimension Type

This indicates what Dimension is currently being used. (e.g., Entity)

### Dimension

This indicates the name of the Dimension.

### Parent Member Dimension

This displays the name of the Parent Member. This may be blank if it is at the root level. (e.g., NA Clubs, Clubs, etc.)

### **Member Dimension**

This displays the name of the Member. (e.g., CorpAccounts, Scenarios, CorpEntities, etc.)

### **Parent Member Name**

This displays the direct Parent to the Member. In some cases, this may display Root if it is the first branch.

### **Member Name**

This displays the name of the Member as it was defined. (e.g., Actual, Budget, Flash, etc.)

## **Position within Parent**

When a new Member is created, it is added to the bottom of the list of its siblings. Use the Position property to move the Member amongst its siblings. The position will only allow movement in the current tree level. This setting cannot move out of the current hierarchy.

Position

Retain Current Position

Before Sibling Member

After Sibling Member

First Sibling

Last Sibling

Sibling Member

This will show a list of all the siblings in the current hierarchy available to move based on the setting in the Position field.

## **Default Parent**

### **Parent Sort Order**

This setting is used to determine a default Parent when evaluating Member lists (e.g., in Cube Views). If a Parent is not explicitly specified, the Entity's Parent with the lowest sort order is used.

## **Vary by Scenario Type and Time**

### **Percent Consolidation**

Define the percentage of the Entity to be consolidated. This setting can also be used to reduce a Member's value by using a -100 value.

## Cubes

---

### **Percent Ownership**

This is an ownership setting that can be used by Business Rules if need be. By itself, the setting has no effect on the consolidation.

### **Ownership Type**

This is an ownership setting that can be used by Business Rules if need be. By itself, the setting has no effect on the consolidation.

### **Full Consolidation**

Normal setting for Entities that fully consolidate into a Parent.

### **Holding**

This is used to designate the Parent/Child relationship as a holding company situation.

### **Equity**

This is used to help Business Rules determine the value to increase the equity method of Accounting for an investment.

### **Non-Controlling Interest**

This is used for Business Rules to determine the minority interest portion of an Entity into the consolidation.

### **Custom 1...Custom 5**

Open for custom use in Business Rules.

### **Text 1-8**

Use to define custom attributes to modify aspects of business rule, transformation rule and member filter capabilities. These custom attributes act as variable placeholders, activated at run-time to customize views of, or ways to interact with, data. You can change these values by scenario type or to suit evolving business needs.

## Dimension Properties Tab

These properties are standard across all Dimensions.

### **Dimension Type**

Identifies the Dimension it currently is such as Account...UD8.

### **Name**

The current name of the Dimension

### **Description**

The current description of the Dimension

### **Access Group**

This security group has access to the Dimension

## Cubes

---

### Maintenance Group

This security group has access and can make changes to the Dimension.

## Source Type

### Standard

Normal metadata Member must match a Cube.

### Business Rule

Connects to a Business Rule so that a set of Members does not have to match a Cube.

### XBRL

Connects to an XBRL taxonomy.

### Source Path

Enter the Business Rule name or the full File System Path for the XBRL Taxonomy's Link Definition File.

'Internal/XBRL/Taxonomies/EntryPoint/....'

### Name Value Pairs (e.g., Param1=Value1,...)

Enter a comma-separated list of name-value pairs if the Source Type is a Business Rule or XBRL.

## Scenario Dimension

Scenario types offer great flexibility and the following 24 pre-set types are provided in each cube. Each type contains an unlimited number of scenarios. Dimensions are assigned to cubes and can differ by scenario type. You can show scenarios as a part of a hierarchy for organizational purposes. They do not consolidate.

- Actual
- Administration
- Budget
- Control
- Flash
- Forecast
- FXModel
- History

## Cubes

---

- LongTerm
- Model
- Operational
- Plan
- Sustainability
- Target
- Tax
- Variance
- ScenarioType1-8

## Security

These groups grant members access and rights to a scenario:

- Read Data Group: Can view data.
- Read and Write Data Group: Can view and modify data.
- Calculate from Grids Group: Can calculate, translate, and consolidate from a cube view or form.
- Manage Data Group: Membership in this group is required to run Data Management steps such as custom calculate or reset scenario. This prevents unauthorized users from launching steps which could alter or clear data unintentionally.

## Workflow

Under Scenario, you can use the following properties to control the type of periods displayed to end users when they load data.

### Use in Workflow

Set to False to omit the Scenario from the workflow view in OnePlace, making it unavailable to users setting the workflow point of view. To display the scenario, set this field to True.

**NOTE:** Data can still be entered with forms and the Excel Add-In to a hidden scenario.

### Workflow Tracking Frequency

This determines how time displays in the workflow and is based on the type of data being entered for the scenario. See "Input Frequency" below for details about how these properties work together.

#### All Time Periods

This is the standard setting, which displays all periods in the year. A period could be months or weeks depending on the application. Once defined, you cannot change this setting.

#### Monthly

Use to set the workflow periods to monthly, which could be 12 to 16 months depending on the application. Once defined, you cannot change this setting.

#### Quarterly

Use to set the workflow periods to four periods, such as Q1, Q2, Q3, Q4. Once defined, you cannot change this setting.

#### Half Yearly

Use to set the workflow periods to two periods, such as H1, H2. Once defined, you cannot change this setting.

#### Yearly

This sets the workflow periods to one period, such as 2021. Once defined, this setting cannot be changed.

#### Range

Allows an Administrator to define a custom range that is displayed as one-time period including the start and end time. As data loads, each period displays, such as 11 – Jan and 11 – Dec. Once defined, this setting cannot be changed. The next set of properties only become available when the Range option is chosen.

**NOTE:** Workflow Tracking Frequency settings cannot be changed if the scenario is used in a workflow.

### Workflow Time

To define the workflow time, click the ellipsis to the right and choose a time. Year or year and month can be defined.

### Workflow Start Time

To define the workflow start time, click the ellipsis to the far right and choose a time. Year or year and month can be defined.

### Workflow End Time

To define the workflow end time, click the ellipsis to the right and choose a time. Year or Year and Month can be defined.

### Number of No Input Periods Per Workflow Unit

Use this free form option to enter a numerical value for the number of periods to be "no input" such as: 1,2,3,5. This disables input from the Import, Form, and Adjustment Origin Members for the specified periods. For example, if the first three months of data is automatically copied from Actual to Forecast with a user inputting the remaining nine months, enter 3 to make those periods read-only and prevent data entry.

## Settings

### Scenario Type

This property groups similar scenario types in order to share settings or Business Rules. A scenario type can contain many scenarios.

### Input Frequency (Vary By Year)

This is the data frequency for the scenario. The current selections are Weekly, Monthly, Quarterly, Half Yearly, and Yearly. To vary this by year, click the ellipsis, choose the year, and make your selection. If the Quarterly or Yearly Frequency is used, data is not saved at a base (monthly/weekly) level as this would display invalid cells.

**NOTE:** The Input Frequency property with the Workflow Tracking Frequency property. For example, if the Input Frequency in 2017 is Monthly and the Workflow Tracking Frequency is All Time Periods, the workflow displays the following for this scenario in 2017:

## Cubes

---

### ▲ 2017 Periods

- ▶ ● Jan Manchester
- ▶ ● Feb
- ▶ ● Mar
- ▶ ● Apr
- ▶ ● May
- ▶ ● Jun
- ▶ ● Jul
- ▶ ● Aug
- ▶ ● Sep
- ▶ ● Oct
- ▶ ● Nov
- ▶ ● Dec

If the Input Frequency varies by year, the Workflow Tracking Frequency updates the workflow view accordingly. For example, if the Input Frequency is Weekly in 2018, the Workflow Tracking Frequency still displays All Time Periods as shown:

### ▲ 2018 Periods

- ▶ ● Week 1 Manchester
- ▶ ● Week 2
- ▶ ● Week 3
- ▶ ● Week 4
- ▶ ● Week 5
- ▶ ● Week 6
- ▶ ● Week 7
- ▶ ● Week 8
- ▶ ● Week 9
- ▶ ● Week 10

### Use Input Frequency Data in Lower Frequencies

Use to display data from the scenario's input frequency when processing a lower level invalid frequency. For example, in a quarterly scenario, display the amount from Q1 in M1, M2, M3 and in all weeks below these 3 months. In a yearly scenario, display the single yearly amount in all time periods in the year.

### **Default View**

This is the standard default view for calculations, member formulas, and clearing calculated data. The current selections are YTD and Periodic.

### **Retain Next Period Data Using Default View**

The scenario's Default View is either Periodic or YTD. Set to True if a Flow account has a data value in a future period (for example, Feb) and the data is being changed in a prior period (for example, Jan). Either the Periodic or the YTD amount for Feb must be changed to be consistent with the new Jan entry. If this setting is False, the Feb Periodic amount is retained if a Jan Periodic number is entered, or the Feb YTD amount is retained if a Jan YTD number is entered.

### **Input View For Adjustments**

This is the standard view setting for entering data using a journal for the period. This setting is determined by how journals are entered, but typically uses the same view as the Default View setting. Current selections are YTD and Periodic.

**NOTE:** Regardless of the setting, all journals, except auto-reversing, must be entered monthly.

### **Use Input View for Adj in Calculations**

Set to True to have calculations use the setting in Input View for Adjustments as the view even if the calculation attempts to override the default scenario view.

### **Zero No Data**

The following properties determine how to handle zero no data. Periodic places a 0 value in the period without data. YTD places a 0 value in the current YTD month, causing the period to negate values from prior months.

### **No Data Zero View For Adjustments**

This is the standard setting for no data in journals for the period. Current selections are YTD and Periodic. Typically, this setting is the same as the Input View for Adjustments setting.

### **No Data Zero View For NonAdjustments**

This is tied to the data load and is used when there is not data for the period. The current selections are YTD and Periodic.

### **Consolidation View**

This is the standard setting for the view of the consolidation. The current selections are YTD and Periodic. Typically, the Periodic setting is used especially if work is completed by period.

However, all numbers are stored as YTD. Select YTD to enhance consolidation performance.

This property can be changed and will update for the next consolidation.

## Cubes

---

**NOTE:** If a Consolidation View is Periodic and no data is loaded to the IC accounts for a specific month, custom elimination rules are required in the FinanceFunctionType.ConsolidateElimination section of a business rule attached to the cube to calculate expected elimination results. This requires storing the C#Share member, which impacts overall consolidation performance.

The YTD consolidated results related to Consolidation View property on a Scenario is set to Periodic and where the Percent Consolidation Relationship Properties of Entities vary by time period, also known as "Org by Period."

When the Consolidation Algorithm Type property on a cube is "Standard," Scenario Consolidation View property of Period and Org by Period Entity Relationship Properties, the Share Consolidation member is consistently displayed for both YTD and Periodic on reports.

When the Consolidation Algorithm Type property on a Cube is set to "Standard with Stored Share," Scenario Consolidation View property of Period, and Org by Period Entity Relationship Properties, when a time period's Percent Consolidation changes to 0.0 when varying over time, the system now displays stored calculated results rather than derived.

**NOTE:** that the best practice for this type of "Org by Period" design typically drives a financial model design where the Consolidation Algorithm Type is set to "Custom." In rare cases, this Cube setting is set as Standard and even more rare, this is set as Standard with Stored Share. In both rare situations, this Share Member view will have changed.

## Formula

This is the ability to use a formula to execute copying between Scenarios prior to the consolidation of a Scenario. See Formula Guide "About the Financial Model" on page 2 for more details.

### Formula for Calculation Drill Down

This enables drill down on members with attached formulas. A specific formula for drill down is needed to display the original formula and the members that activate the drill down to the original values. You can drill down on calculated data cells and data cells copied via a Copy Data Management Sequence.

See "Formulas for Calculation Drill Down" in "About the Financial Model" on page 2 for more information.

### Clear Calculated Data During Calc

If True, existing data is cleared during the calculation process. If False, data can be cleared manually. The default setting is True.

## Cubes

---

### Allow Input Into the Aggregated Consolidation Member

If True, allows you to submit data for base member values through the aggregated member. The default setting is False.

## FX Rates

### Use Cube FX Settings

If True, the rate type that is the default for the current cube is used. If False, custom rate calculations are required. Rate Type for Revenues and Expenses & Rate Type for Assets and Liabilities.

#### Average Rate

The average currency rate of a period from the first day to the last day of the month.

#### Opening Rate

The currency rate at the beginning of the period.

#### Closing Rate

The currency exchange rate at the end of a period.

#### Historical Rate

The currency rate to use for a particular historical Account calculation open so a special transaction can be valued on a specific date. Rule Type for Revenues and Expenses & Rule Type for Assets and Liabilities.

#### Direct

Calculation is direct with the current value and current rate.

#### Periodic

Calculation is weighted based on a period.

#### Constant Year for FX Rates

This setting is available when a scenario's Use Cube FX Settings is False. When set to False, this setting becomes a drop-down box displaying all available years. Once set, the rates for the selected year are used.

## Hybrid Scenarios

For information about how the following properties impact hybrid scenarios queries and using hybrid scenarios, see "Working With Hybrid Scenarios" on page 109

#### Data Binding Type

Indicates the type of data binding that should occur for the target scenario. Choose an option:

## Cubes

---

- **Share Data from Source Scenario:** shares a read-only cube data set from the source Scenario to the target Scenario.
- **Copy Input Data from Source Scenario:** copies base level cube data from a source Scenario to the target Scenario.
- **Copy Input Data from Business Rule:** copies base level cube data based on a Finance business rule.

### Source Scenario or Business Rule

Enter the Source scenario member or business rule to indicate the source location.

### End Year

This is the only property that controls Time. If a data query must end after a specific year, indicate that year here. For example, if it should not occur in 2021, the End Year is 2020 to exclude all future years from a query. To query all years, leave this field empty. All years containing data will be included in the results.

The database stores data records by year, each year having its own data table and containing the data records for each period. When data is queried using a Hybrid Scenario, it occurs at the database level, and only returns the periods containing data.

### Member Filters

The members listed in a comma separated list are the only ones included in data query results. This can include multiple dimension types, member expansions, or single members. If this field is blank, all source data is included in query results.

### Member Filters to Exclude

The members listed here in a comma separated list, are excluded from query results. This can include multiple dimension types, member expansions or single members. If this includes members from the Member Filters property, those members are excluded.

**NOTE:** You cannot use Data Unit dimensions in the Member Filters or Member Filters to Exclude properties (Entity, Time, Consolidation, Scenario).

### Pre-aggregated Members

## Cubes

---

Use this property to share or copy data from a parent member (source) to a base member (target). For example, if you query the Top member of a large dimension such as a UD, the aggregated total is calculated "on the fly" each time.

If the detail of a dimension is not needed, set the top member to a Base member to pre-aggregate. This alleviates repetitive "on the fly" calculations for the same number.

UD1#Top=UD1#None , UD2#Top=UD2#None , UD3#Top=UD3#None , UD5#Top=UD5#None , UD5#Top=UD5#None , UD6#Top=UD6#None

Name	4_ShareActual	Description	Share Pre-Aggregated
<b>POV</b>			
Parent Member	US Clubs	...	
Consolidation Member	Local	...	
Scenario Member	ShareActual	...	
Time Member		...	
View Member	YTD	...	
Account Member	69000	...	
Flow Member	None	...	
Origin Member	Top	...	
IC Member	Top	...	
UD1 Member	None	...	
UD2 Member	None	...	
UD3 Member	None	...	
UD4 Member	None	...	
UD5 Member	None	...	
UD6 Member	None	...	

**NOTE:** Ensure that the base members shown above are included in the Member Filters property and the parents are included in the Member Filters to Exclude property.

## Options

Hybrid Source Data Options provide additional control when executing a hybrid share or copy. The settings are optional and will vary. Options are name-value pairs. Ensure that the option names, definitions and syntax are accurate and include any custom name-value pairs if a business rule is used to copy. Create a comma separated list if you use multiple options.

### ExecuteCopyDuringCalc

## Cubes

---

A True / False property setting which is False by default. If True, Hybrid Source Data copy is executed during the Data Unit Calculation Sequence (DUCS). If False, the Hybrid Source Data copy will not run during the DUCS and a Calculate Data Management Step type is needed to execute the Hybrid Source Data copy. On the Data Management step, set **Execute Scenario Hybrid Source Data Copy** as **True**. The calculation defined on this Data Management step uses the settings from the **Hybrid Source Data** properties to run the copy.

General (Step)	
Name	CopytoBudgetVersion
Description	
Data Management Group	HybridSourceData
Step Type	Calculate
Calculation	
Calculation Type	Force Calculate
Execute Scenario Hybrid Source Data Copy	True
Data Units	
Cube	Houston
Entity Filter	E#Houston.Base
Parent Filter	
Consolidation Filter	C#Local
Scenario Filter	S#BudgetWhatif
Time Filter	T#2022M1

This setting is helpful for versioning or seeding where constantly copying unchanging numbers adds unnecessary overhead to every calculation. This setting should be used with either of the Copy binding types and is not applicable with the Share binding type.

### CopyAsDurableData

A True / False property setting which is False by default. If True, copied data has a Storage Type of "Calculation". Calculated data is cleared at the beginning of the DUCS (if the Scenario setting of **Clear Calculated Data During Calc** is **True**) and will re-calculate. If this setting is True, copied data has a Storage Type of "DurableCalculation". Durable data is not cleared at the beginning of DUCS (regardless of the Scenario setting for Clear Calculated Data During Calc) and must be cleared manually with a rule.

The **ExecuteCopyDuringCalc** and **CopyAsDurableData** settings are used together. Typically, if **ExecuteCopyDuringCalc** is **True**, **CopyAsDurableData** is **False** (and vice versa). Use this setting with either of the Copy binding types. It is not applicable with the Share binding type.

### RemoveZeros

A True / False property setting that is False by default. If True, zeros are removed when copying data if all periods within the year of the intersection being copied are 0.

### SourceTimePeriodForYearlyResult

## Cubes

---

This property setting is used when the data results only require data from a specific source Time member. Indicate the source Time Period. Define that time period here. For example:

**SourceTimePeriodForYearlyResult=M6**

The hybrid scenario's Input Frequency must be Yearly, the only frequency that accepts a value from one period. The source scenario's Input Frequency can be more granular such as Weekly or Monthly. If a source time period is not defined, the source Year value is used in the hybrid Year Value.

Examples include a weekly period (Wx), a monthly period (Mx), a quarterly period (Qx) or a half year period (Hx). You cannot specify a year. It will match the year between the source and target.

### **SourceViewForYearlyResult**

This setting is used if the Input Frequency of the target scenario is Yearly and the source scenario has a more granular Input Frequency. This setting can be Periodic or YTD. YTD is the default. You can indicate if you want to copy / share the time period being copied / shared as a periodic or YTD value.

The **SourceTimePeriodForYearlyResult** and **SourceViewForYearlyResult** settings are used together.

## Custom Settings

### **Text 1...Text 8**

Enables you to user custom attributes for many purposes such as business rules, member filters or transformation rules. The value can be changed in the tag over time as the business changes, or by Scenario Type.

## Account Dimension

Account Members are meant to be organized in a hierarchical fashion. The calculation engine will aggregate these Members, and then perform math on them as they roll up based on their Account Type (e.g. Revenue is positive and Expense is negative). Be sure to set Is Consolidated to False for those items that do not necessarily need to be consolidated for performance purposes.

## Security

### **Display Member Group**

The group that can see that this Account exists within a list of Accounts.

## Settings

**AccountType**

Account attributes to determine the behavior of the Accounts.

**Group**

This is only an organization Member and does not accumulate data. It is meant to only view and organize data.

**Revenue**

Setting for Income Accounts to tag with a revenue attribute. Amount does not have to be negative because of this attribute.

**Expense**

Setting for Expense Account to tag with an expense attribute.

**Asset**

Setting for Asset Accounts

**Liability**

Setting for Liability Accounts. Amount does not have to be negative because of this attribute

**Flow**

Setting to hold values that act like an Income Statement Account and have a periodic and year to date value. This account does not translate.

**Balance**

Setting to hold values that act like a Balance Sheet Account that are at a particular time. This account does not translate.

**BalanceRecurring**

Setting for a Balance Sheet Account that does not change over time such as an Opening Balance. This account does not translate.

**NonFinancial**

Setting for Informational Accounts that are captured and not financial such as Headcount or Square Footage. This is primarily used for legacy purposes such as upgrading from older systems. This account does not translate. NonFinancial and Balance Account Types are similar in that they are available for legacy purposes, however one difference between them is that a NonFinancial data cell is not affected by the Flow Member's Switch Type setting.

**DynamicCalc**

An Account that calculates on the fly and does not need other formulas to run in order to calculate. An example of these types of Accounts are ratios that can be calculated as needed.

## FormulaType

### **FormulaPass1...FormulaPass16**

The formulas are included in the Account metadata and can be shared between Cubes. The formula pass is to define when the calculation should run and whether it is dependent on a calculation from other formulas to derive its value.

### **DynamicCalc**

A Dynamic Calc formula computes a value for a single cell and runs every time the cell needs to be displayed without storing the result.

### **DynamicCal Text Input**

This runs a Dynamic Calc formula and is open for text annotations on the Cube View data cell.

For more details on using Formulas, refer to the "Formulas" on page 25.

### **Allow Input**

If set to True (default), data input for the Account Member is allowed. This is typically set to False if this Account has a formula. This will make the Account read-only and will not affect the formula. If specific Scenarios or Entities need input, set to True and use the Cube's Conditional Input Settings to control input.

### **Is Consolidated**

If set to **Conditional (True if no Formula type (Default))** it determines which Accounts will be part of the consolidation. If the Account has a Formula Type, the Account Member will be calculated and consolidated only if the setting is True.

If set to **False** the account should not be consolidated.

### **Is IC Account**

If set to True, this account is identified as an Intercompany Account and allows transactions to be processed based on the settings in the constraint section under IC Constraints and IC Member Filter, if set to False, it will not be identified as an Intercompany Account.

### **Use Alternate Input Currency In Flow**

If set to True, this account will use the Historical Currency override, if set to False, it will not use an alternate currency.

### **Plug Account**

Intercompany Plug Account to handle any non-eliminating transactions.

### **Input View for Adjustments**

This setting defines how the Account Member will perform for adjustments. The Scenario's Input View for Adjustment setting dictates how the account will process as an adjustment. The setting defaults to Use Scenario Setting (Default). To override the Scenario's Input View for Adjustment, use this setting to change it to either YTD or Periodic.

## Cubes

---

### No Data Zero View For Adjustments

This setting defines how the Account Member will perform for adjustments with no data zero. The Scenario's Input View for No Data Zero View For Adjustments setting dictates how the account will process as an adjustment with no data zero. The setting defaults to Use Scenario Setting (Default). To override the Scenario's Input View for No Data Zero View For Adjustments, use this setting to change it to either YTD or Periodic.

### No Data Zero View For NonAdjustments

This setting defines how the Account Member will perform for non-adjustments with no data zero. The Scenario's Input View for No Data Zero View For NonAdjustments dictates how the account will process as no adjustment with no data zero. The setting defaults to Use Scenario Setting (Default). To override the Scenario's Input View for No Data Zero View For NonAdjustments, use this setting to change it to either YTD or Periodic.

## Aggregation

This can be used to turn off aggregation for specific Dimensions preventing them to roll up. It might be possible that a Dimension Member is used for informational purposes only and does not need to aggregate. Settings are True or False.

- Used On Entity Dimension
- Used On Consolidation Dimension
- Enable Flow Aggregation
- Enable Origin Aggregation
- Enable IC Aggregation
- Enable UD1...UD8 Aggregation

## Vary by Cube Type

See Entity Dimension

## Vary by Scenario Type

Workflow Channel

### Standard

This is a basic Member with no special purpose other than to act as the default Workflow Channel for Account Members and Workflow Profile Input Children.

### NoDataLock

This is a special Member that only applies to a metadata Member (Account or UDx) and should not participate in a Workflow Channel grouping scheme. This is the default value for any UDx Member.

## Vary by Scenario Type and Time

During normal business, Accounts can have different attributes based on Scenario and Time. The time display for any time varying properties uses the Standard Time Dimension Profile.

### In Use

If set to True, the Account is in use, if set to False, this can turn off the ability to use an Account based on Time. This keeps historical data available but allows the ability to close the Account without having to have No Input rules.

### Formula

An individual formula kept with the Account across Cubes and can vary by Time. For example, a calculation changes over time, but the historical interpretation of that formula needs to be saved. See the Formula Guide, in "About the Financial Model" on page 2 for more details.

### Formula for Calculation Drill Down

This will allow drill down to occur on Members with attached formulas. A specific formula for drill down is needed in order to display the original formula used and the Members that activate the drill down back to the original values.

1. Click the ellipsis button located in the property field.

Vary By Scenario Type And Time	
General	
In Use	True
Formula	
Formula For Calculation Drill Down	

2.

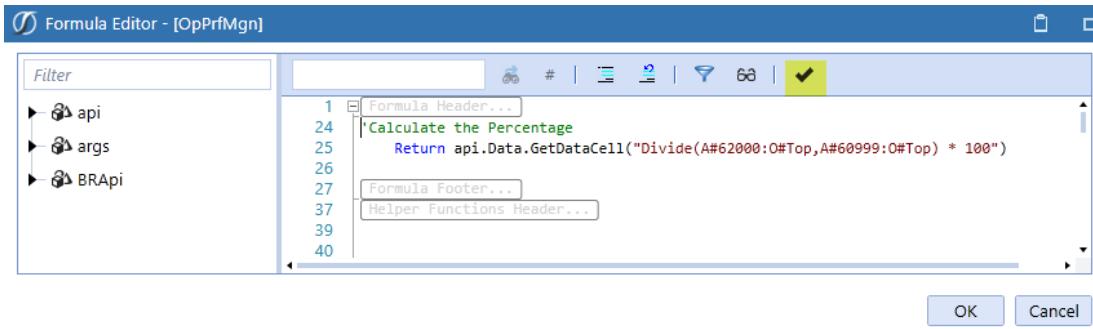
3. Next, click the ellipsis button in the Stored Value field in order to input a formula.

Stored Property Value	
Scenario Type	(Default Scenario Type)
Time	(Default Time)
Stored Value	

4.

5. Type the individual formula in the Formula Editor for the specific Account. Click the check

mark icon in order to ensure that the formula was compiled correctly.



6.

**TIP:** Formulas can be stacked if a particular account intersection needs to have a special formula different from a normal process. Formulas are not limited to one per metadata Member.

### Adjustment Type

This can limit the use of adjustments over time. Settings include Not Allowed, Journals and DataEntry.

### Text 1...Text 8

Open for custom attributes used for multiple purposes such as Business Rules, Member Filters or Transformation Rules. The value can be changed in the tag over time as the business changes, or by Scenario Type.

## Relationship Properties

For information on General and Position within Parent see General Relationship Properties

### Aggregation Weight

The Aggregation setting is available in all Account, Flow and UD Dimensions. This setting can change for a Base Entity based on its Parent. If a Member is reused in a Dimension, but it does not need to sum up more than once, set the weight to 0 data in this node and it will not allow the Member to aggregate to the top.

## Flow Dimension

The Flow Dimension is like the eight User Defined Dimensions in that it can use Extensible Dimensionality® and extend Dimensions across business units and Scenario Types with flexibility. Its Members aggregate up just like the other User Defined Dimensions. The Flow Dimension, however, has some additional settings that help with historical currency overrides and eliminate many common custom scripts in order to perform calculations.

## Security

### Display Member Group

The group that can see that this Dimension exists within a list of Dimensions.

## Settings

For Formula Type, Allow Input and Is Consolidated, see Account Dimension.

### Switch Sign

The Flow Member is linked to an Account through Flow constraints. Depending on the Account Type, this property would be set to True to switch the sign of data for the Flow Member or set to False to keep the sign as is.

### Switch Type

Switch the type of data based on the Account attribute, for example setting an Asset to a Revenue. This is useful when treating roll forward Accounts as Income Statement Accounts in the Balance Sheet. Settings are True or False.

## Flow Processing

These settings are used for dollar override values. In order to be used, an Account must be flagged as true by using the setting UseAltInputCurrencyInFlow. (Except for IsAlternateInputCurrencyforAllAccounts) The Flow Dimension can hold both values of a dollar override and the settings differ based on how this setting is used.

## Flow Processing Type

### Is Alternate Input Currency

This option is used to indicate if data should be input into this Flow Member in a currency other than the entity's local currency. This setting works similarly to **Is Alternate Input Currency for All Accounts**, except it only applies to a specified list of accounts that are configured in the Account dimension.

## Cubes

---

If selected, this setting activates **Alternate Input Currency Flow Member Property**, where you can select a specific currency from the list configured through the Application Properties. Choose the accounts that should be overridden by setting the **Use Alternate Input Currency** property to **true** for a specified list of accounts.

This is used when currency overrides are implemented. Many times OneStream applications spanning multiple currencies have this requirement to override the translated amount with a specific value. The most common use case for this setting is for overriding the USD amount for **Balance Sheet Accounts**.

### **Is Alternate Input Currency for All Accounts**

This option is used to indicate if data should be input into this Flow Member in a currency other than the entity's local currency. This setting works similarly to **Is Alternate Input Currency**, except it applies to all accounts and no further set up is required on the Account dimension.

If selected, this setting activates **Alternate Input Currency Flow Member Property**, where you can select a specific currency from the list configured through the Application Properties.

This is used when currency overrides are implemented. Since this opens currency overrides for all accounts, the use cases will vary. One common use case is if data for an entire entity is loaded in two or more separate currencies instead of translating.

### **Translate using Alternate Input Currency, Input Local**

This will override the translated value with the amount input at the local currency level.

### **Translate using Alternate Input Currency, Derive Local**

This will override the translated value and change the local currency value to be derived based on what the local currency rate would be. This setting probably should not be used in a trial balance unless accounting for the out of balance condition that might result.

### **Alternate Input Currency**

This setting contains a list of all available currencies to put in for the source value override. If this Flow Member has a USD override, then it should be set to USD. If the override is a EUR override, then it should be set to EUR. Typically, this is the default currency of the application, but can also be used in instances of functional currency.

### **Source Member for Alternate Input Currency**

Define the actual Flow Member to override the value for the current Flow Member.

## **Vary by Scenario Type and Time**

See [Account Dimension](#).

## User Defined Dimensions 1-8

Members can take full advantage of Extensible Dimensionality for inheriting and extending across business units and Scenario Types. When determining the order in which to use User Defined Dimensions, it is recommended to define the larger or more significant Dimensions first using UD1 or UD2. Stacking hierarchies in one User Defined Dimension does not affect calculation time because performance is primarily based on the number of stored data cells, rather than the number of possible intersections.

## Security

### Display Member Group

The group of users who can see that this Dimension exists within a list of Dimensions.

## Settings

For Formula Type and Allow Input see "Account Dimension" on page 355.

### Is Consolidated

If set to **Conditional (True if no Formula Type and no Attribute (default))**: The data from this entity's children is consolidated; this entity will equal the total of its children.

If set to **True (regardless of Formula Type and Attribute)**: Consolidate the results of the dimension and the attribute. Consolidate and/or aggregate UD attributes when they reference entity, as the reference Dimension to view results at the parent entity instead of having the Parent entity use the same algorithm as the base entities.

If set to **False**: The data will not be consolidated. Set to False when using the Parent Entity strictly for grouping purposes. Also, by setting Is Consolidated to False helps with consolidation performance times because the consolidation will not be performed at the Parent Entity.

### Alternate Currency for Display

This is a setting to change the Cube View grid currency value to another value. It does not recalculate the Member based on the currency. This requires a formula on the Member to recast the transaction from another Member to the current currency.

## Vary by Cube Type

### **UD1 Default for Other User Defined Dimensions**

Set a default value on the UD1 Dimension in order to drive the Members for other User Defined Dimensions (2 through 8). For example, if the UD1 Dimension is Cost Centers, an administrator can drive which UD2 Member (such as Department or Product) will be chosen based on the UD1 selection. A constraint can be placed on the other User Defined Dimensions in order to create conditions for valid intersections in the Cube.

### **UD2-UD 8 Constraint**

A constraint is a setting that allows only certain Members to be used. If a Member is outside the UD Constraints applied on the Account Dimension, its cube intersection will show as a red/invalid intersection and any numbers in that intersection will not aggregate. Constraints applied on the Entity and UD1 Dimensions will create a green, no input data intersection.

### **UD2-UD8 Default**

This is the standard default Member that can be mapped for a setting and saves having to map to each Member. If the setting says Default, the mapping will always go to that default.

## Vary by Scenario Type

See [Account Dimension](#).

## Vary by Scenario Type and Time

See [Account Dimension](#).

## User-Defined Members as Attribute Members

A User Defined member can be alternatively defined as an Attribute Member by activating the Is Attribute Member property. When activated as an Attribute, the User Defined Members becomes a “read only” member based on the settings for the related reference properties. Data intersections are not loaded directly to an Attribute, its results are derived from references to properties of other members, such as Names, Descriptions or Text fields. These members act like other dimension members by deriving their values based on the references to other properties and can then be used in reporting.

## Cubes

---

The Account, Flow and User Defined Dimensions, as Related Dimension Types, support the Attribute Members for “calc-on-the-fly” aggregations at parent members. The dynamically generated results within the Attribute Members will be automatically aggregated to the parent members. Being stored members, the Entity and Scenario, as Related Dimension Types, do not support “calc-on-the-fly”. Attribute Member results on Entity or Scenario will only be available on base members.

The use of User-Defined Attribute Member can impact application performance, particularly with respect to Consolidation time. The impact is due to the dynamic generation of User-Defined Attribute members' data intersections adding to the size of the final Data Unit. Therefore, the potential intersections derived from User-Defined Attribute Members should be included in application Data Unit analysis. As a guideline, typical application designs should always consider the performance evaluation of User-Defined Attribute Members should the number of User-Defined Attribute Members approach approximately 2000 items.

The User-Defined member as an Attribute is unique:

- The data will dynamically calculate across each dimension hierarchy.
- The Attribute members will not impact the size of the Data Unit in Consolidation.
- Values derived by Attribute members can be referenced by Business Rules and by member formulas.
- The members are treated as standard dimensions and records in that they will be processed within a Cube View's Allow Sparse Suppression routine supporting large sparse application model reporting.
- The results can be modified quickly and easily by modifying the definition of the reference on the Attribute member or from a change on the properties of source member(s) even if those properties (e.g. Text4) vary by Scenario Type or Time.

The model design and use of Attributes should consider if the feature is appropriate for the application model. Here are some considerations:

- Attributes may not be appropriate in situations where reporting on the Attribute member must be maintained with a high level of data integrity. This is due to the dynamic nature of the Attribute where its results are based on properties of other members.
- Attribute results cannot be locked for data integrity. Although the underlying data being referenced will be locked, modifying the definition of the Attribute or a change on the properties of the referenced source member, this may impact the results.

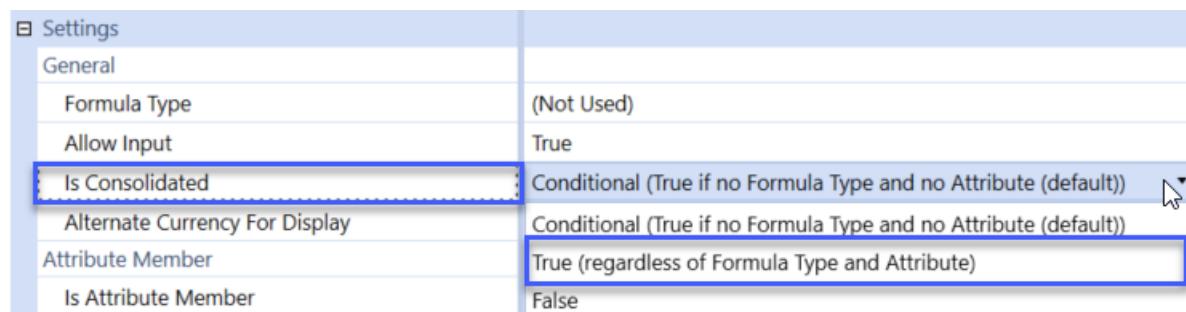
- Since Attribute members do not store data, they can be deleted and are not subject to data integrity restrictions if in use. Therefore, dynamic designs of reports and use in rules should be considered.
- Attributes cannot be input or contain formulas; however, they can reference other input members or calculated members as a source.
- Attribute data cannot be extracted to a data file.
- Drill-Down based on the Attribute member intersection cannot be used to drill-back to the Stage Load Results, the Source Member for Data reference member defined on the Attribute must be used.

## Consolidate UD Attributes

Consolidate UD Attributes, when referencing Entity as the reference Dimension, consolidates Attribute results. The results could consolidate vs the Parent entities using the same algorithm as the Base Entities.

Set Is Consolidated to True on the UD Attribute to enable attribute members to Consolidate to Parent Entities.

- Numbers entered at the None Level settings Attribute Member, referenced Dimension is Entity.
- Uses of the “Is Consolidated” setting.
- Default is “Conditional (True if no Formula Type and no Attribute (default)).
- Consolidation is enabled when “True (regardless of Formula Type and Attribute)” – consolidate the results of the dimension and attribute.



## Cubes

---

Is Consolidated	True (regardless of Formula Type and Attribute)
Alternate Currency For Display	Conditional (True if no Formula Type and no Attribute (default))
Attribute Member	True (regardless of Formula Type and Attribute)
Is Attribute Member	False

UD Attribute will also work with the Aggregated Cons Member.

- Switch Local to Aggregated in the Cons Dimension
- Local, Aggregated and Top
  - At the Base Level Entity Aggregated and Local are the same.
  - At the Parent Entity it switches over and Aggregated is a different Member.
- It works across all of the Consolidation Dimension Members.
- It is the Sum of the Children; not a pointer at the Parent Level.

## Setting Attribute Properties

The User Defined Member Settings under the Attribute Member section are used to define the source of the data used for results.

An Attribute member can define its source reference by up to two physical dimensions using the Related Dimension Type. The interaction between the two references is handled by the Expression Type.

## Expression Types

- Comparison 1 Only: This is used for an Attribute utilizing only a single Related Dimension
- Comparison 1 and Comparison 2: This is to utilize two Related Dimensions where the results will be bound by meeting both Related Dimension conditions
- Comparison 1 Or Comparison 2: This is to utilize two Related Dimensions where the results

## Cubes

---

will need to meet one of the Related Dimension conditions

Settings	
General	
Formula Type	(Not Used)
Allow Input	True
Is Consolidated	Conditional (True if no Formula Type (default))
Alternate Currency For Display	(Not Used)
Attribute Member	
Is Attribute Member	False
Source Member For Data	
Expression Type	Comparison 1 Only
Related Dimension Type 1	(Select One)
Related Property 1	(Select One)
Comparison Text 1	
Comparison Operator 1	(Select One)
Related Dimension Type 2	(Select One)
Related Property 2	(Select One)
Comparison Text 2	
Comparison Operator 2	(Select One)

### Is Attribute Member

True/False to enable the User Defined Member as an Attribute.

### Source Member For Data

This represents the member within the User Defined Dimension, containing the current attribute, which will be used to define the data returned. This member must be a base level member, such as the None member or a calculated base member. A hierarchy parent member cannot be used.

Members   Dimension Properties   Grid View

The screenshot shows the 'Members' node in the SSAS Management Studio navigation pane. Under the 'Members' node, there are several categories: 'Members', 'Orphans', 'Hierarchy', 'AttributeMembers', and 'CalculatedMembers'. The 'AttributeMembers' category is expanded, showing members like 'EntityDefault', 'UD1Default', and 'None'. The 'None' member is highlighted with a yellow box and has a red callout pointing to it with the text 'Source Member for Data'.

### Expression Type

Used to utilize one or two Related Dimensions and the conditional relationship between two to return results.

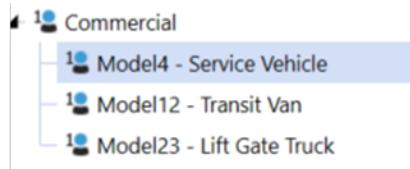
### Related Dimension Type

Identify the Dimension to use as a source to be evaluated. To support on-the-fly aggregations, an Attribute can reference any Account, Flow or User-Defined dimension. Entity and Scenario are stored members; therefore, Attributes will only be reflected as base members. They cannot reference the User Defined Dimension in which it is contained.

### Related Property

The Related Dimension can be evaluated on its Name, Description, Name or Description, Name and Description, Text1-8. The Related Property will be evaluated only on the Base Members of the Related Dimension Type.

## Related Property Examples



The Related Property as:

- **Name:** Model4
- **Description:** Service Vehicle
- **Name and Description:** Model4 – Service Vehicle. Note the hyphen that OneStream creates when presenting the Name and Description will need to be included in the Comparison Property when evaluating as Property = Comparison Text
- **Name or Description:** This property is to conditionally evaluate by Description. If no Description is present, it then evaluates using the Name. The condition is resolved by first evaluating the Description field. If there is a Description field, the evaluation is performed using the Description field. If there is not a Description, it will evaluate the property using the Name field.

### Comparison Text

The text condition being evaluated against the defined Related Property as the Name, Description or Text Field. When referencing a Text Field, the source Text Field can vary by Scenario and Time. When referencing the Description, only the Default Description can be referenced.

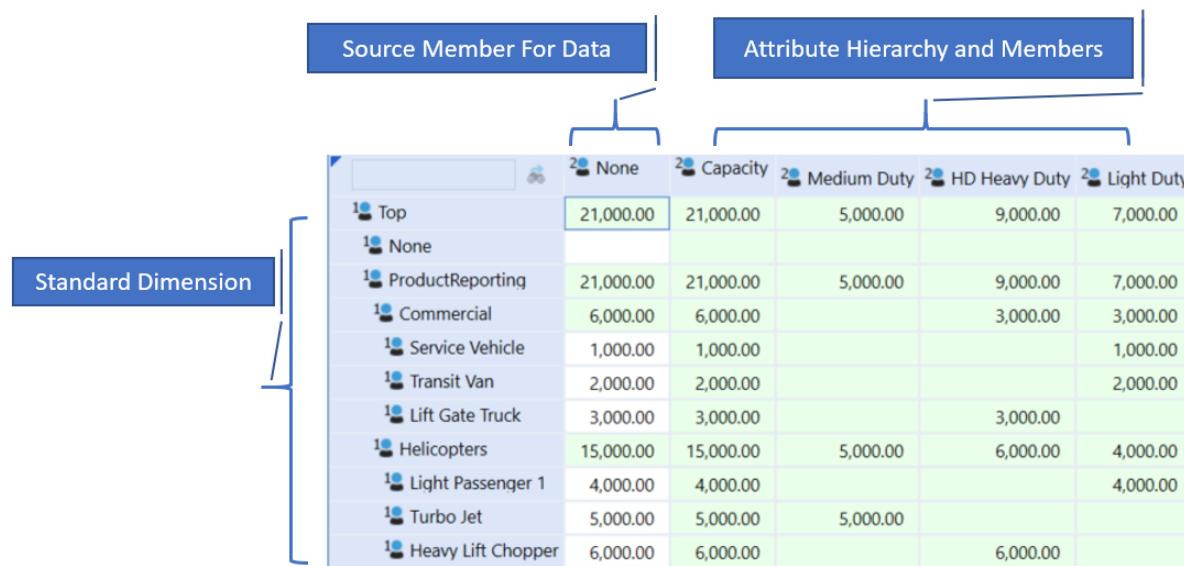
## Cubes

---

### Comparison Operator

Sets the evaluation method to compare the Related Property to the Comparison Text. This can be done explicitly with “=” or “<>” as well as dynamically using the Starts With, Ends With, Contains or Does Not Contain operators.

Once the Attribute members and are defined, and any required Comparison Text is applied to the Source Related Dimension Type, the data will render dynamically. Neither Consolidation nor Calculation is required to render the results. Any change applied to the Source definition of the Comparison text, such as a change to a Text property, will immediately be reflected in the results on the Attribute members. Similarly, modifying the Attribute Member’s Setting Properties will immediately change the Attribute results.



## Defining Source Members

Source dimensions for Attributes, which is the assigned Related Dimension Type on the Attribute Member, are evaluated only on the Base member, not parent members. Therefore, the Attribute’s Property, Related Property, and the corresponding Comparison Text, are designed against collecting data at the base level member on the source dimension. When referencing the Text1 through Text8 field, the vary by Scenario Type and Time can be used to vary the results by Scenario and Time.

## Cubes

---

The screenshot shows the Cubes application interface. On the left, the 'Members' tab is selected, displaying a tree view of dimension members under 'Hierarchy'. A blue box labeled 'Only Base Members' points to the tree. On the right, the 'Dimension Properties' tab is selected, showing a grid of properties for a specific member. A red box highlights the 'Name' field, which is set to 'Model4'. Another red box highlights the 'Default Description' field, which is set to 'Service Vehicle'. A third red box highlights the 'Text' section, which contains a table with two rows: 'Text 1' and 'Text 2', both of which have 'Light mfg' assigned to them.

## Using Attributes within Business Rules

Attributes can only be called through a business rule using a DataBuffer. The function property, includeUDAttributeMembersWhenUsingAll, can be enabled to allow rules to reference the Attribute results for use in formulas.

```
Dim objDataBuffer As DataBuffer =  
    api.Data.GetDataBuffer(scriptMethodType,  
        sourceDataBufferScript,  
        changeIdsToCommonIfNotUsingAll, includeUDAttributeMembersWhenUsingAll, expressionDestinationInfo)
```

## Consolidation Dimension

The Consolidation tree of Dimension Members is how the data rolls up from the local currency to the final numbers. The tree gives opportunities to go from local currency to translated currency showing any intercompany eliminations and allows for the Entity to adjust the tree before or after the final numbers. Also included are all the currencies assigned to this application. See Consolidation in "About the Financial Model" on page 2 for more information on the Consolidation Dimension.

## Cubes

---

Dimension  ▾

Filter

Hierarchy

The screenshot shows a user interface for managing cube dimensions. At the top, there's a dropdown menu labeled "Dimension" with the value "Consolidation - Consolidation". Below it is a "Filter" input field and a "Refresh" button. A "Hierarchy" button is also present. The main area displays a tree structure of dimensions:

- None
- Currencies
- Top
  - OwnerPostAdj
  - Elimination
  - Share
  - OwnerPreAdj
  - Translated
  - Local

## Time Dimension

Time is a fixed Dimension and is based on the Time Dimension Type associated with the application. The Time Dimension type determines whether data is stored at a monthly or weekly level and how many months are in a year, months are in a quarter, and weeks are in a month. The Time hierarchy is driven by this and will display in the Point of View accordingly.

See Time Dimensions in "System Tools" on page 894 for more details on Time Dimension Types.

## Monthly Time Dimension Hierarchy

Dimension **Time - Time** ▾

Filter  Refresh

Hierarchy Search

```
graph TD; 2014[2014] --> 2014H1[2014H1 - H1 2014]; 2014 --> 2014Q2[2014Q2 - Q2 2014]; 2014H1 --> 2014Q1[2014Q1 - Q1 2014]; 2014Q1 --> 2014M1[2014M1 - Jan 2014]; 2014Q1 --> 2014M2[2014M2 - Feb 2014]; 2014Q1 --> 2014M3[2014M3 - Mar 2014]; 2014Q2 --> 2014M4[2014M4 - Apr 2014]; 2014Q2 --> 2014M5[2014M5 - May 2014]; 2014Q2 --> 2014M6[2014M6 - Jun 2014]
```

# Weekly Time Dimension Hierarchy

Dimension ▼

Filter  Refresh

Hierarchy

```
graph TD; 2017[2017] --> 2017H1[2017H1 - H1 2017]; 2017 --> 2017Q1[2017Q1 - Q1 2017]; 2017 --> 2017M1[2017M1 - Feb 2017]; 2017Q1 --> 2017M1Jan[2017M1 - Jan 2017]; 2017M1Jan --> 2017W1[2017W1 - Week1 2017]; 2017M1Jan --> 2017W2[2017W2 - Week2 2017]; 2017M1Jan --> 2017W3[2017W3 - Week3 2017]; 2017M1Jan --> 2017W4[2017W4 - Week4 2017]; 2017M1 --> 2017W5[2017W5 - Week 5 2017]
```

The Time hierarchy can be used by fiscal year and is not tied to a calendar year by using the fiscal year and Month1 (M1), Month2 (M2), etc. to designate the time frame.

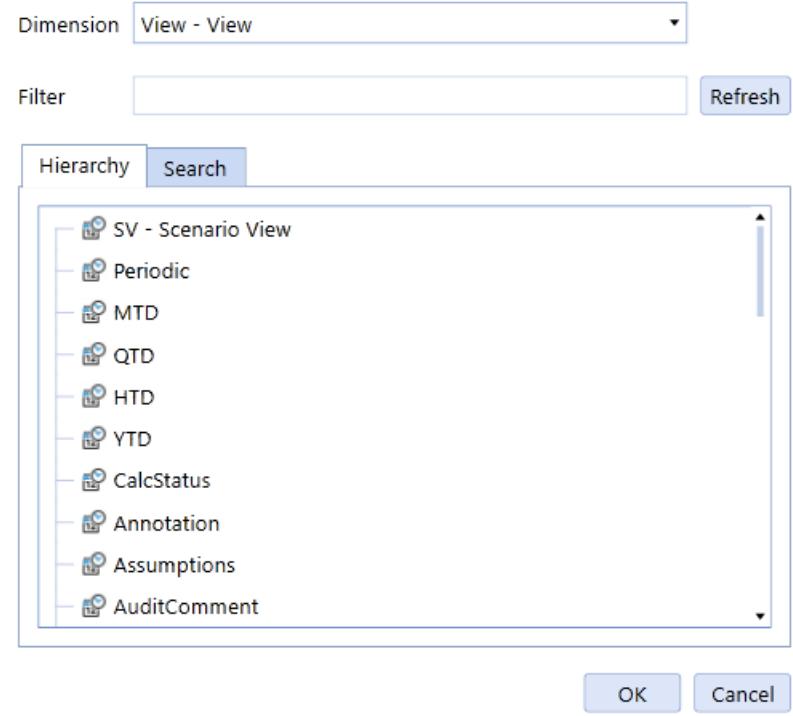
**NOTE:** There are many issues with rolling retained earnings and beginning balances for corporations that do calendar reporting and fiscal year reporting in the same application. For example, if a large corporation had two companies using different reporting years, the books would have to be separate because each company is based on a different method. In these instances, two applications would need to be used for each reporting year.

## View Dimension

The View is how to see the data or look at the text that was entered in certain Dimension Members. Common calculations are included because the data is easily available, and it saves having to create custom formulas. Also included is CalcStatus which shows the calculation status for a Data Unit and several views into different types of Data Attachment comments, such as Annotation, Assumptions, Footnote and VarianceExplanation.

## Cubes

---



## View Member Abbreviations

When using View Members, shorthand versions of the Members are recognized.

Periodic	Trailing2MonthAverage-Trailing12MonthAverage
Per	Trailing2MonthAvg-Trailing12MonthAvg, T2Avg-T12AVG
Month to Date	QuarterAverage
MTD	QuarterAvg, QAvg
Quarter to Date	HalfYearAverage
QTD	HalfYearAvg, HAvg
Half Year to Date	YearAverage
HTD	YearAvg, YAvg
Year to Date	YearSumToDate
YTD	YSumTD
Calculation Status	YearSumToDatePrior

## Cubes

---

CalcStatus, CS	YSumTDPrio
AuditComment	Trailing2MonthTotal-Trailing12MonthTotal
Audit	T2Total-T12Total
VarianceExplanation	
VarExp	

## Origin Dimension

Origin is the channel where data is loaded for each Workflow. Data can be a file import, data entry, forms or journal entries that can be adjusted based on input currency or consolidated data. Eliminated data is from eliminations at common Parents between Entities' intercompany transactions. See Consolidation in "About the Financial Model" on page 2 for more information on Consolidation and Elimination.

The screenshot shows a user interface for managing the Origin dimension. At the top, there is a dropdown menu labeled "Dimension" set to "Origin - Origin". Below it is a "Filter" input field and a "Refresh" button. Underneath these, there are two tabs: "Hierarchy" (which is selected) and "Search". The main area displays a hierarchical tree structure:

- Top**
  - BeforeElim**
    - BeforeAdj**
      - Import**
      - Forms**
    - Adjustments**
      - AdjInput**
      - AdjConsolidated**
    - Elimination**
  - DirectElim**
  - IndirectElim**

## Intercompany (IC) Dimension

Intercompany is a derived Dimension based on the Entity Dimension. A setting must be selected in order to have intercompany account transactions with another Entity. Both Entities must be selected as an intercompany to have the ability to record a transaction with each other. This protects the integrity of the intercompany data ensuring that an Intercompany Account cannot have an intercompany transaction with an Entity if it cannot receive an intercompany transaction. See Consolidation in "About the Financial Model" on page 2 for more information.

## Time Profiles

Time Profiles are used to create different Fiscal Years in order to apply them to different Cubes. Use the Standard Time Dimension Profile if the Fiscal Year begins in January, otherwise create a new Time Dimension Profile with the desired Fiscal Year start date. Once the Time Dimension Profile is created, assign it to the specific Cube to which it is being used. Time Dimension Profiles are included during Metadata load/extract. The default Time Dimension Profile for Cube Views is determined by the Cube POV settings. The time display for the Workflow POV is determined by the Cube assigned to the Workflow Profile. The Standard Time Dimension Profile is used for all time varying properties such as Formulas, In Use, Text Fields, etc.

Time Dimension Profiles can also be specified in a Substitution Variable or Member Filter Functions and used in Cube Views. See Substitution Variables for Cube Views or Dashboards and Commonly Used Member Filter Functions for more details.

## Profile

### Name

The name of the Time Dimension Profile.

### Description

Include a description for the Time Dimension Profile.

### Fiscal Year Start Date

The start date of this Time Dimension Profile's Fiscal Year.

### Fiscal Year is First Period's Calendar Year

Set this to True to specify the Fiscal Year is based on the first period of the Calendar Year. The default setting for this property is False.

## Fiscal Year Month Type

### Calendar Month

Standard 12-month calendar

## Cubes

---

### Fixed Weeks 4-4-5

A weekly calendar determined by four weeks in first month, four weeks in second month, and five weeks in third month.

### Fixed Weeks 4-5-4

A weekly calendar determined by four weeks in first month, five weeks in second month, and four weeks in third month.

### Fixed Weeks 5-4-4

A weekly calendar determined by five weeks in first month, five weeks in second month, and four weeks in third month.

### Custom Start Dates

Customize the start date of each month. Select the Time Periods tab, select each month and specify the start date.

**NOTE:** For Fixed Weeks, each month is either 28 or 35 days except for the twelfth month which will have an extra one or two days.

### Vary Settings By Year

**NOTE:** Set this to True to vary time descriptions by year.

If Custom Start Dates is the selected Fiscal Year Month Type, monthly start dates can also vary by year.

**NOTE:** There are many issues with rolling retained earnings and beginning balances for corporations that do calendar reporting and fiscal year reporting in the same application. For example, if a large corporation had two companies using different reporting years, the books would have to be separate because each company is based on a different method. In these instances, two applications would need to be used for each reporting year.

## Time Periods

Descriptions use |fy|, |fyfy|, |cy|, or |cycy| to include Fiscal or Calendar Year.

### Default Year

#### Year Description

This displays the Year as either Fiscal Year or Calendar and defaults as |fyfy|.



## Half Years

### HY1, 2 Description

This displays the Half Year as either Fiscal Year or Calendar Year and defaults as H1 |fyfy|.

- ▶  2014H1 - H1 2014
- ▶  2014H2 - H2 2014

### HY1, 2 Short Description

This displays the Half Year's short description and defaults as H1.

## Quarters

### Q1, 2, 3, 4 Description

This displays the Quarters as either Fiscal Year or Calendar Year and defaults as Q1 |fyfy|.

- ▶  2014Q1 - Q1 2014
- ▶  2014Q2 - Q2 2014

### Q1, 2, 3, 4 Short Description

This displays the Quarters short description and defaults as Q1, Q2, Q3, Q4.

## Months

### M 1-12 Description

This displays the Months as either Fiscal Year or Calendar Year and defaults as this format Jan|cycy|.

-  2014M1 - Jan 2014
-  2014M2 - Feb 2014
-  2014M3 - Mar 2014

### M1-12 Short Description

This displays the Months short description and defaults as Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec.

## Cubes

---

### Weeks (if applicable)

#### Description

This displays the Weeks as either Fiscal Year or Calendar year.

Example: Week1 |fyfyl|

- 2017W1 - Week1 2017
- 2017W2 - Week2 2017
- 2017W3 - Week3 2017
- 2017W4 - Week4 2017

#### Short Description

This displays the week's short description and defaults to W1, W2, W3, etc.

## Cubes

A Cube is the organizational structure that holds data. A Cube is flexible and can be designed for a specific purpose or type of data. An Application can have multiple Cubes that share Dimensions and data.

For more information on Cubes and Extensible Dimensionality®, see Extensible Dimensionality® in "About the Financial Model" on page 2.

### Cube Properties

#### Name

Input field for creating a new Cube

#### Description

Input field for description of a new Cube.

#### Cube Type

This is an optional setting that creates tags for different types of Cubes which may be used to separate and capture types of data without affecting other Cubes. This is used in order to have different settings for Default and Constraint settings that apply to certain Dimensions and can vary by Cube Type, such as Entity. The Cube Type names are arbitrary and do not have functional differences but are there to represent different kinds of Cubes that may be created.

#### Standard

Cube used for normal Consolidation

#### Tax

Cube specifically used for Tax

## Cubes

---

### Treasury

Cube specifically used by Treasury

### Supplemental

Cube specifically used to capture supplemental data

### What If

Cube specifically used to capture various What If Scenarios.

### Cube Type 1...Cube Type 8

Custom Cube Types

### Time Dimension Profile

Assign the Time Dimension Profile created for the Cube. If this Cube uses January as the start year, select Standard. See Time Profiles earlier for more details on this feature.

## Security

### Access Group

A user can see the object and read its contents.

### Maintenance Group

A user can see the object, create new objects in groups, edit, and delete them.

**NOTE:** Click  in order to navigate to the Security screen. This is useful when changes need to be made to a Security User or Group before assigning it to a

Cube. Click  and begin typing the name of the Security Group in the blank field. As the first few letters are typed, the Groups are filtered making it easier to find and select the desired Group. Once the Group is selected, click CTRL and Double Click. This will enter the correct name into the appropriate field.

## Workflow

### Is Top Level Cube for Workflow

If additional Cubes are used that roll their data into the main Cube via Extensible Dimensionality®, other Cubes should be set to False. Each Cube listed as True for this setting can be set as a (top level) Cube Root Profile for Workflow Profiles, meaning that Cube can have different Workflows than the others.

### Suffix for varying Workflow by Scenario Type

This is active only for Cubes that have Is Top Level Cube for Workflow set to True. There is one setting for each of the Scenarios Types. A suffix is expected to be added onto a Cube Root Profile when Workflow Profiles begin to be built which means by default there is no suffix, so all Scenario Types will follow the exact same Workflow Profile structure.

For example, in order to have Budget, Plan and Forecast Scenario Types follow a different Workflow Profile structure than any other Scenario Types, add a suffix such as Plan to these three Scenario Types. When the Workflow Profiles are built, there will be two Cube Root Profiles. If the main Cube is called GolfStream, and there is a suffix of Plan for the three Scenarios Types listed above, and no suffix for the other Scenarios Types, two Cube Root Profiles will be created called GolfStream and GolfStream\_Plan.

## Calculation

### Consolidation Algorithm Type

#### Standard (Calc-on-the-fly Share and Hierarchy Elimination)

This is the default Consolidation algorithm. Amounts for the Share Consolidation Member are calculated dynamically and amounts for the Elimination Consolidation Member are calculated using built-in algorithms.

#### Stored Share (Stored Share and Hierarchy Elimination)

This is similar to Standard, but it stores the values for the Share Member instead of calculating them dynamically. It only considers the position in the Entity hierarchy tree.

#### Org-By-Period Elimination (Calc-on-the-fly Share and Org-By-Period Elimination)

When determining if the data cell's IC member is a descendant of the Entity being consolidated, this option considers the position in the Entity hierarchy and also checks the percent consolidation for every relationship down the hierarchy. If a percent consolidation is zero, the IC member is determined to not be a descendent of the entity.

#### Stored Share And Org-By-Period Elimination

Stores the values for the Share Member instead of calculating them dynamically. When determining if the data cell's IC member is a descendant of the Entity being consolidated, this option considers the position in the Entity hierarchy and also checks the Percent Consolidation for every relationship down the hierarchy. If a Percent Consolidation is zero, the IC member is determined to not be a descendent of the Entity.

#### Custom

The Consolidation will utilize custom Business Rules to calculate amounts for the Share and Elimination Consolidation Members using the Finance Function Types of Consolidate Share and Consolidate Elimination.

## Translation Algorithm Type

### Standard

This is the default Translation algorithm. Amounts for a foreign currency Consolidation Member are generated from the Local Consolidation Member using the FX rate tables.

### Standard Using Business Rules for FX Rates

This is similar to the Standard Translation algorithm. However, it provides the ability to use a custom Business Rule to specify different FX Rates for any data cell intersection.

### Custom

The translation will be run entirely through Business Rules assigned to the Cube. It is assumed that the Business Rule will calculate translated amounts from data in the Local Consolidation Member.

### No Data Calculation Rules

These settings specify whether the calculation engine will execute Member Formulas and Business Rules for the specified Consolidation Member even if there is currently no data stored in the Data Unit (i.e., the data for a specific Entity, Parent, Consolidation, Scenario, and Time). There is a potential performance benefit of using False to skip the formulas for Data Units that have no data. However, if part of the calculation is to pull data from another Scenario or prior Time Period, then the setting needs to be True. Otherwise, the calculation will not run for an empty Data Unit, so the data will never be pulled from another Scenario or Time Period.

Calculate None Cons Member If No Data

Calculate Local Currency If No Data

Calculate Translated Currencies If No Data

Calculate OwnerPreAdj If No Data

Calculate Share Cons Member If No Data

Calculate Elimination Cons Member If No Data

Calculate OwnerPostAdj If No Data

## Business Rules

Custom Business Rules can be attached to a Cube. Eight different Business Rules can be defined. See Business Rules in "Application Tools" on page 727 for more details. This method of definition allows for extreme flexibility to share or not share certain Business Rules between Cubes. See Consolidation in "About the Financial Model" on page 2 to see the calculation order of Member formulas vs. Business Rules attached to each Cube.

### BusinessRule1...BusinessRule8

These settings are limited to attaching the Finance Business Rules only.

**NOTE:** Click  in order to navigate to the Business Rules screen. This is useful when changes need to be made to a specific Business Rule before assigning it to a Cube.

## FX Rates

### **Default Currency**

This setting is the default reporting currency for the Cube. This is used for FX rate triangulation if the Cube currency is the common currency. This is also used for Intercompany Matching's reporting currency.

### **Rate Type for Revenues and Expenses & Rate Type for Assets and Liabilities**

Note that these are the default settings for this Cube and can be overridden at the Scenario-level with the same settings.

### **Average Rate**

The average currency rate of a period from the first day to the last day of the month.

### **Opening Rate**

The currency rate at the beginning of the period.

### **Closing Rate**

The currency exchange rate at the end of a period.

### **Historical Rate**

The currency rate to be used for a specific historical Account calculation open for a specific special transaction to be valued on a specific date.

### **Rule Type for Revenues and Expenses & Rule Type for Assets and Liabilities**

Note that these are the default settings for this Cube and can be overridden at the Scenario-level with the same settings.

### **Direct**

Calculate direct with current value and current rate.

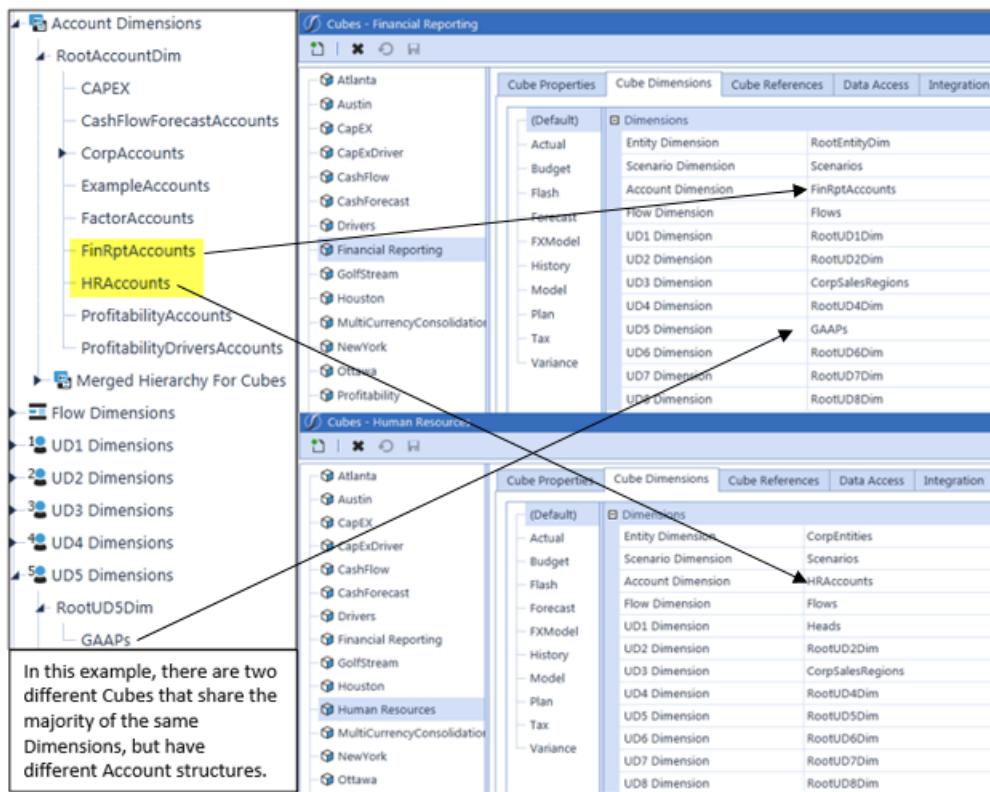
### **Periodic**

Calculate periodic value translation method. This method considers the translation rates for prior time periods and calculates a form of average.

**NOTE:** Click  in order to navigate to the FX Rates screen. This is useful when changes need to be made to FX Rates before assigning them to a Cube.

## Cube Dimensions

Cube Dimensions determine what metadata Dimension Member trees will be used as a default and for a specific Scenario Type. The Dimensions are defined in the Dimension Library. There is a default Scenario Type and a specific Scenario Type. Different Dimension Members can be used for different Scenario Types. For example, a different Region Dimension may be used for Budget that has more detail (through Extensible Dimensionality®) than the Dimension being used for Actual.



## Cube Reference

This is for sharing Cubes through Extensible Dimensionality®. Multiple Cubes can be linked together and extended for different purposes. See Entity Dimension for more details. These options appear in a Top-Level Cube for Workflow where there are references in the assigned Entity Dimension to other Entity Dimensions. In the example below for the GolfStream Cube, the CorpEntities Dimension have references to the five other Entity Dimensions following it. Therefore, in this example, the GolfStream Cube needs to be told which Cube to find data for those other Entity Dimensions.

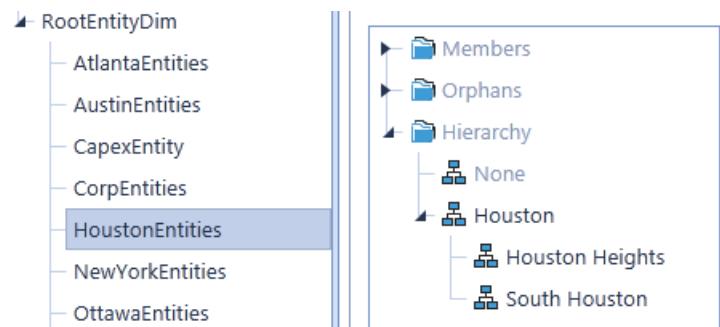
## Cubes

---

The screenshot shows the 'Cubes - GolfStream' application window. On the left is a tree view of cubes: Atlanta, Austin, CapEX, CapExDriver, CashFlow, CashForecast, Drivers, Financial Reporting, and GolfStream. The 'GolfStream' cube is selected. The main area has tabs: Cube Properties (selected), Cube Dimensions, Cube References, Data Access, and Integration. The 'Cube References' tab displays a table titled 'Cube For Referenced Entity Dimensions':

HoustonEntities	Houston
OttawaEntities	Ottawa
AtlantaEntities	Atlanta
AustinEntities	Austin
NewYorkEntities	NewYork

GolfStream acts as a Super-Cube that consolidates data from five other Cubes because the Houston Entity and its children are referenced within the CorpEntities Dimension and the Cube References are updated to point to the Houston Cube for their data (shown below).



## Data Access

These are security settings that control access to the Entity:

### Data Cell Access Security

Blocking a User Group from knowing the existence of a certain Entity or Account can be accomplished right on the Member in the Dimension Library. Data Cell Access Security is where that access rule can be made more granular than the Application/Cube/Entity/Scenario level. Here, No Access, Read Only or All Access can be granted to an intersection of data.

## Cubes

---

A typical use case is to use the Read-Only and Read-Write security group settings on Entities and Scenarios to specify the users that need access to any data for those Dimensions. Then, Cube Data Access Security can be used to further refine which data certain users can access. For example, if restricting read and write access by Cost Center (which may be set up as UD1 in the application) is wanted, this can be done by having entries in Cube Data Access Security that specify which users have access to certain cost centers.

If security needs to be controlled for combinations of Members involving multiple Dimensions, “slices” can be defined using Member Filters when providing access. For example, an Entity typically has a primary group of people responsible for that Entity, but an administrator might also want to provide limited access to that Entity for another larger group of people. If the larger group of people is only allowed to view data for summary level accounts and only for a specific product segment set up in UD2, Member Filters can be used to provide access only to the corresponding data cell intersections. In addition, the ability to reference the current Entity’s name and text properties using Substitution Variables can also be used to simplify security maintenance when product segments and users are different for each Entity.

First, choose a User Group, the level of access, and then enter a Member Filter. For example, a User Group that includes Senior Management and Human Resources can have All Access to actual compensation figures (S#Actual, A#[Total Compensation].Tree), but everyone else will have No Access.

Note that each of these Data Cell Access Security rules either grants or takes away access. This depends on the Action, Behavior and Access Level and the order in which the rule appears in the list.

## General

### **Category**

This is an optional Category name by which access rules can be named and grouped. If these categories are created, more than one can be applied to an Entity’s security settings. If the category is left blank in the Entity’s security settings, then all these rules will apply.

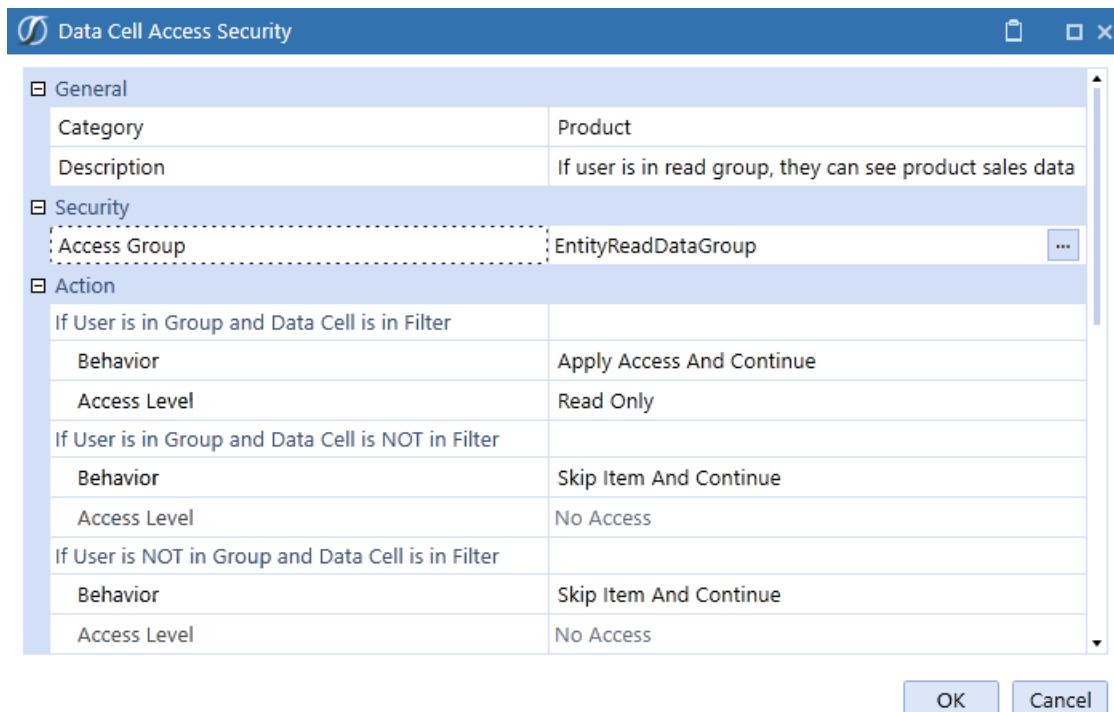
### **Description (Optional)**

Description for the rule.

## Security

### Access Group

This is the group of users to which particular security roles apply. It can be an actual named security group or refer to an Entity or Scenario group. The first four options refer to the Entity's Read Data Group, Read Data Group 2, Read Write Data Group or Read Write Data Group 2. The 5<sup>th</sup> and 6<sup>th</sup> group are the Scenario Read Data Group or Read Write Data Group. For example, if a user is in the Read Data Group for an Entity, and he/she needs to be given access to Product Sales data for that Entity, the rule would be set up as follows:



Further down in the dialog:

UD1		...
UD2	U2#TotalProducts.DescendantsInclusive	...
UD3		...

All Access groups from the 7<sup>th</sup> Access Group down are the full list of security groups from the specific Framework database.

## Cubes

---

EntityReadDataGroup
EntityReadDataGroup2
EntityReadWriteDataGroup
EntityReadWriteDataGroup2
ScenarioReadDataGroup
ScenarioReadWriteDataGroup
Administrators
Everyone

## Action

Action	
If User is in Group and Data Cell is in Filter	
Behavior	Apply Access And Continue
Access Level	Read Only
If User is in Group and Data Cell is NOT in Filter	
Behavior	Skip Item And Continue
Access Level	No Access
If User is NOT in Group and Data Cell is in Filter	
Behavior	Skip Item And Continue
Access Level	No Access

## Actions

There are three cases that will drive different behaviors and access levels for this particular Data Cell Access Security rule in relation to other rules that came before or after it in the list. First, it depends on whether the user trying to query or update data is in a particular User Group and second, it depends on if the cell of data in question falls within a certain Member Filter. These are the three cases:

- If User is in Group and Data Cell is in Filter
- If User is in Group and Data Cell is NOT in Filter
- If User is NOT in Group and Data Cell is in Filter

## Behavior

There are eight possible behaviors that coincide with the three action cases. For example, the “Increase Access...” rules will increase support while going down the list of rules. The rules in the list will continue until it either reaches the end of the list or it reaches a Behavior that includes the word “...Stop.”

## Cubes

---

### Skip Item and Continue

Default for If User is in Group and Data Cell is NOT in Filter or If User is NOT in Group and Data Cell is in Filter

### Skip Item and Stop

Choose this behavior to skip a Cube Data Access Item and stop evaluating the remaining Cube Data Access Items.

### Apply Access and Continue

Default for If User is in Group and Data Cell is in Filter

### Apply Access and Stop

Choose this behavior to apply access to a Cube Data Access Item and stop evaluating the remaining Cube Data Access Items.

### Increase Access and Continue

Choose this behavior to increase access to a Cube Data Access Item and then continue evaluating the remaining Cube Data Access Items.

### Increase Access and Stop

Choose this behavior to increase access to a Cube Data Access Item and then stop evaluating the remaining Cube Data Access Items.

### Decrease Access and Continue

Choose this behavior to decrease access to a Cube Data Access Item and then continue evaluating the remaining Cube Data Access Items.

### Decrease Access and Stop

Choose this behavior to decrease access to a Cube Data Access Item and then stop evaluating the remaining Cube Data Access Items.

## Access Level

### No Access

Cannot read or write to the cell.

### Read Only

Can read the cell.

### All Access

Can read and write to the cell.

These properties work in conjunction with the security that is placed on an Entity. Refer to the Security section under Entity Dimension to get a better understanding of how this works.

## Data Cell Conditional Input

Data Cell Conditional Input is not a security setting, i.e., the same setting applies to all users. A good use of Data Cell Conditional Input is when a Dimension Member is intended to be used for input sometimes, but used for a calculation elsewhere. For example, if users want to manually type in F#OpeningBalance in the Budget Scenario, but use a formula in the Actual Scenario, Data Cell Conditional Input could be used to enable write access to the data cell appropriately.

### Category

This is an optional Category name by which access rules can be named and grouped. If these categories are created, more than one can be applied to an Entity's security settings. If the category is left blank in the Entity's security settings, then all of these rules will apply.

### Description (Optional)

Description for the rule.

### Action

There are two cases that will drive different behaviors and access levels for this particular Data Cell Conditional Input rule in relation to other rules that came before or after in the list. This depends on whether the cell of data in question falls within a certain Member Filter. These are the two cases:

If Data Cell is in Filter

If Data Cell is NOT in Filter

Based on the Action case, a series of Behaviors and Access levels will apply. See Action under Data Cell Access Security for more information on Access Level choices.

### Member Filters

These are the areas of the Cube that are affected by this rule.

## Data Management Access Security

Data Management Access Security helps determine what areas of a Cube can be modified through a Data Management Sequence or step being launched by a user.

### Category

This is an optional Category name by which access rules can be named and grouped. If these categories are created, more than one can be applied to an Entity's security settings. If the category is left blank in the Entity's security settings, then all these rules will apply.

### Description (Optional)

Description for the rule.

### Security

See Data Cell Access Security for a description.

### Action

See Data Cell Access Security for a description.

### Member Filters

These are the areas of the Cube that are affected by this rule.

See Security Best Practices in "Implementing Security" on page 252 for more information on Data Access Security.

## Integration

This section controls the Dimensions that can be updated via the Import Input Type in Workflow and the columns of data that appear in the Stage database when imported data is viewed. It can determine the order of processing Dimensions and which Dimensions are turned on for data integration. If one Dimension has to process before another (rare case), the transformation sequence could be changed.

### Cube Dimensions

This is a list of Dimensions to which a particular Cube can be mapped.

### Cube Dimension Name

This cannot be changed.

### Transformation Sequence

Order in which a Dimension's Transformation Rules will be processed for a given Scenario Type.

### Enabled

When a Dimension is Enabled, it will appear in the Stage area and data can be imported into it for a particular Scenario Type.

### Special Dimensions

There are also special fields that can be mapped to the Stage area but cannot be mapped to the Cubes. They can be drilled back from the Cube or used in Derivative Rules. The use of these fields does increase processing times. Here are the choices:

### Label

This would be the description for any given Account that was related to a particular line of data. It is imported just for reference purposes.

### Source ID

This is one of the keys to the data in the Stage and should be enabled. It can be mapped in via a Data Source and can be set to a particular value in a file, the file name, or even a tab name from an Excel file. Max length is 100 characters.

### TextValue

This is to store large amounts of textual data. Max size is 2GB of text.

## Cubes

---

### Attribute Dimensions

These 20 Dimensions can each store 100 characters of text.

### Attribute Value Dimensions

These 12 Dimensions can store numeric data. By default, the Zero Suppression settings for Attribute Value Dimensions are set to True. When developing a Data Source using Attribute Value Dimensions where a specific value is 0, be aware that the data may be suppressed. Therefore, in some cases, Zero Suppression for an Attribute Value Dimension may need to be set to False.

These are the properties for each item:

#### **Alias**

This controls the column header that appears for this Dimension when viewing data in the Import Input Type in Workflow.

#### **Transformation Sequence**

This is the order in which a Dimension's Transformation Rules will be processed for a given Scenario Type.

#### **Enabled**

When a Dimension is Enabled, it will appear in the Stage area and data is allowed to be imported into it for a particular Scenario Type.

## Foreign Exchange Rates

Foreign Exchange rates are stored in the system Cube as a central repository for all other Cubes to reference the currency rates. This reduces the need to place the rates in each Cube as a default. Exchange rates give the system the ability to convert currency values in the database from one currency into another. (e.g., Converting USD to Euro)

The settings in the Global Point of View essentially give a Cube View the currency rates that can be input or viewed in the system.

## Grid Settings Intersection

### FX Rate Type

#### **Average Rate**

The average currency rate of a period from the first day to the last day of the month.

#### **Opening Rate**

The currency rate at the beginning of the period.

#### **Closing Rate**

The currency exchange rate at the end of a period.

## Cubes

---

### Historical Rate

The currency rate to be used for a specific historical Account calculation open for a specific special transaction to be valued on a specific date.

### Time

Select Time frame

### Source Currency

Select type of currency

### Destination Currency

Select Destination Currency for viewing in this Cube View

## Rows and Columns

### Row Axis

This drop down defines what to display in the rows

### Column Axis

This drop down defines what to display in the columns.

Example of FX Rates:

Grid Settings				
Intersection				
FX Rate Type		AverageRate		
Time		2013		
Source Currency		USD		
Destination Currency		USD		
Rows And Columns				
Row Axis		Destination Currencies		
Column Axis		Source Currencies		
	CAD	EUR	GBP	USD
CAD				
EUR				
GBP				
USD				

## Lock FX Rates

Use the Lock FX Rates feature to lock FX Rate Types. The feature prevents or allows all FX Rate Types rates to be changed. This may help eliminate mistakes and provides Audit and Task Activity information around FX Rate activities, adding integrity to the FX Rate Type data.

File Loads or XFSetRate functions will fail if an FX Rate Type is locked.

The application locks the FX Rate Type by Time only, not by details of source or destination currency.

All locking is performed from the interface or a business rule.

Security Administrators have full rights and all functionality including locking and unlocking, by default. To assign non-administrators lock and unlock rights requires standard security group assignments in Application > Security Roles.

Non-administrators:

- Must be granted access to Application User Interface Roles / FxRatesPage. This will grant View access to the Rates and the Locking Page.
- Access to the FX Rate Types, creating and deleting, and modifying/loading rates has not changed. Users must be granted Application Security Roles / ManageFXRates.
- Users that will lock rates manually or via BRAPI Rules will require Application Security Roles / LockFxRates.
- Users that will unlock rates manually or via BRAPI Rules will require Application Security Roles / UnLockFxRates.

## Security Roles that Manage the Locking Features

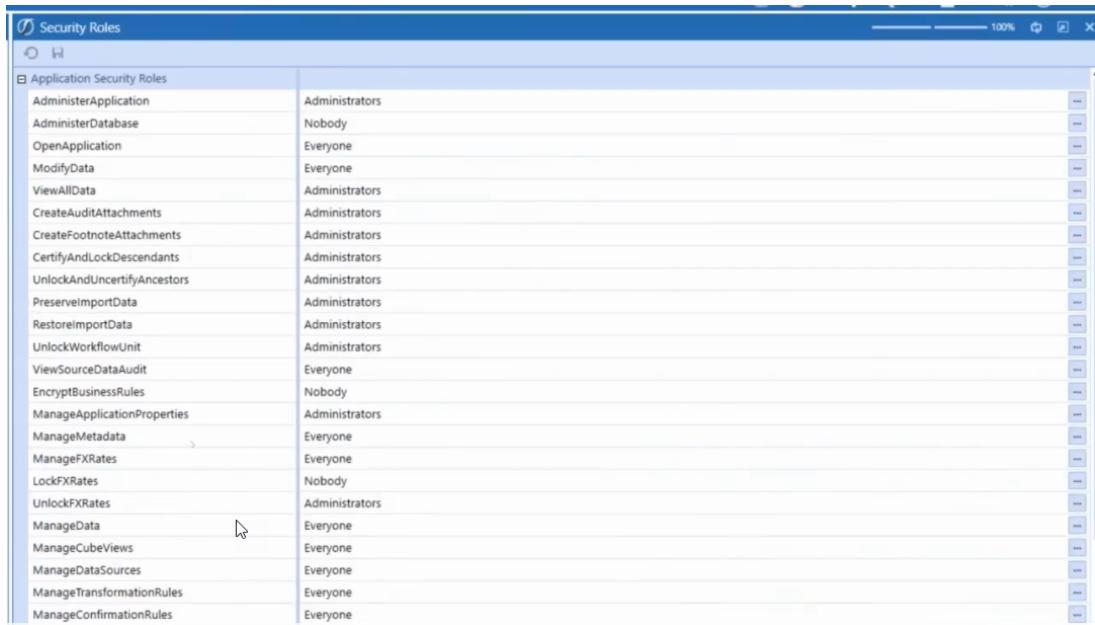
Depending on the settings for LockFXRates and UnlockFXRates, users will have access to the Manage FX Rate Locking window.

The Administrator User Group by default has both roles. Other users can be assigned permissions for lock only, unlock only or both. You do not need to be able to manage the rates, lock/unlock is a separate role.

Users need security rights to each of these roles to lock and unlock.

## To access FX Rates Security Roles

1. From the Application tab, go to **Tools > Security Roles**.



The screenshot shows a software interface titled "Security Roles". On the left, there is a tree view under "Application Security Roles" with several items listed. To the right of the tree view is a table with two columns: "Role" and "User Group". The table contains 20 rows, each corresponding to one of the roles listed in the tree view. The "User Group" column lists either "Administrators" or "Everyone" for most roles, except for "ManageFXRates" which is listed under "Everyone". The last three rows ("ManageConfirmationRules", "ManageTransformationRules", and "ManageDataSource") are collapsed, indicated by a small arrow icon next to their row number.

Role	User Group
AdministerApplication	Administrators
AdministerDatabase	Nobody
OpenApplication	Everyone
ModifyData	Everyone
ViewAllData	Administrators
CreateAuditAttachments	Administrators
CreateFootnoteAttachments	Administrators
CertifyAndLockDescendants	Administrators
UnlockAndUncertifyAncestors	Administrators
PreserveImportData	Administrators
RestoreImportData	Administrators
UnlockWorkflowUnit	Administrators
ViewSourceDataAudit	Everyone
EncryptBusinessRules	Nobody
ManageApplicationProperties	Administrators
ManageMetadata	Everyone
ManageFXRates	Everyone
LockFXRates	Nobody
UnlockFXRates	Administrators
ManageData	Everyone
ManageCubeViews	Everyone
ManageDataSources	Everyone
ManageTransformationRules	Everyone
ManageConfirmationRules	Everyone

2. Change ManageFXRates to allow users visibility to the FX Rates functionality.
3. (Optional) Change LockFXRates to allow users to lock rates.
4. (Optional) Change UnlockFXRates to allow users to unlock rates.

**NOTE:** Users can have access to either LockFXRates or UnlockFXRates. They do not need access to both.

## System Changes: Administrator Access to Lock FX Rates

- Task Activity: as the rates are locked and unlocked, the application tracks who performed the lock or unlock and when.

## Cubes

---

- Audit FX Rate Lock Table: track the current status of the Rate Type as to who performed the unlock or lock.

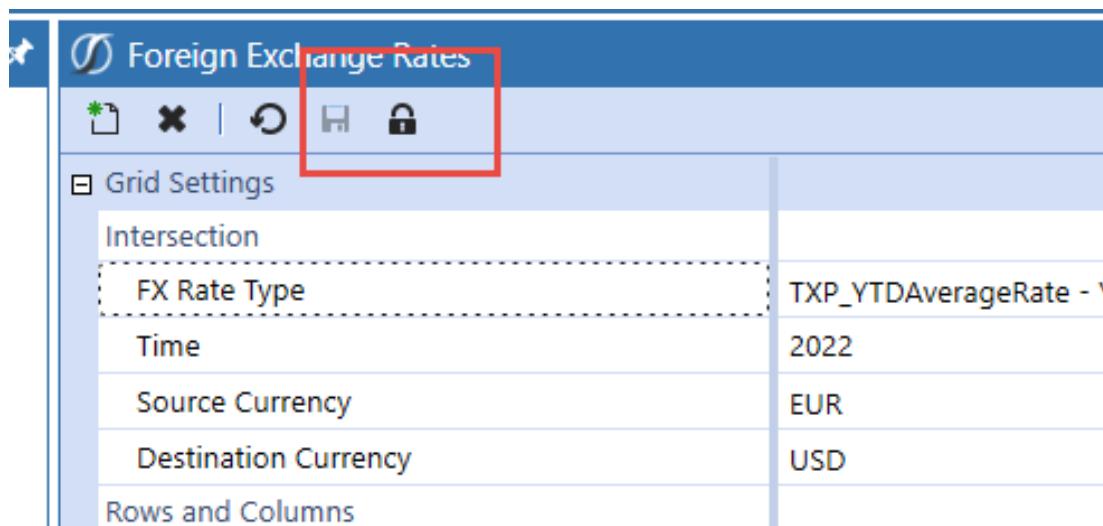
## View Task Activity

There are two new Task Activity logs produced. Lock Fx Rate and Unlock Fx Rate will be generated whenever the Lock status is modified.

1. From the application, click **Task Activity**.
2. In the task list you will see which users have locked or unlocked rates and the time.

## Lock and Unlock Rates

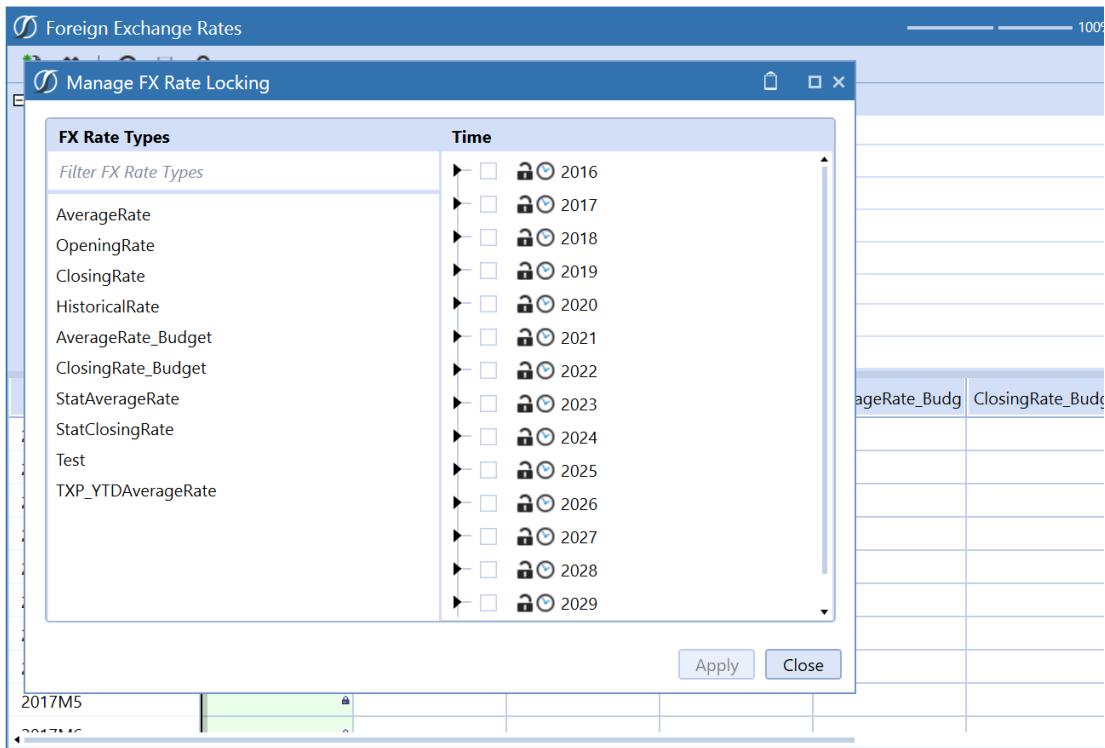
1. From the Application tab, go to **Cube > FX Rates**.
2. Click the lock icon at the top of the page.



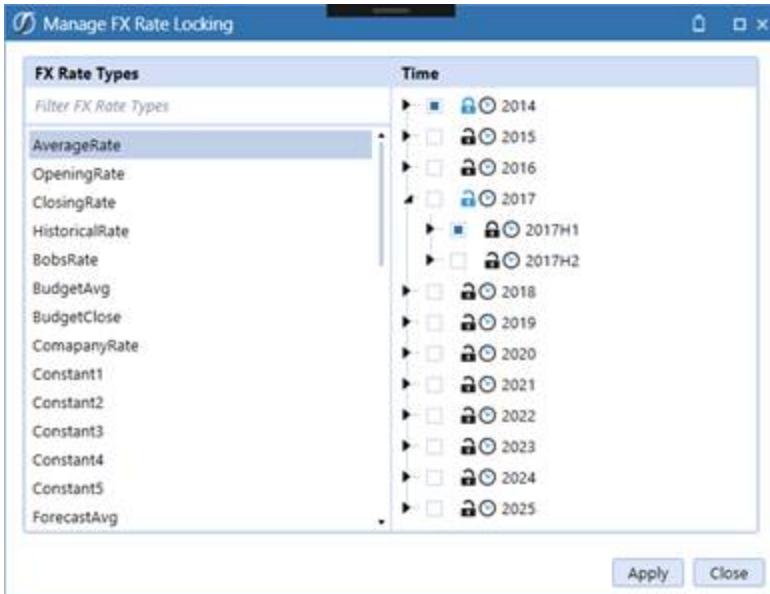
3. The Manage FX Rate Locking window opens.

## Cubes

---



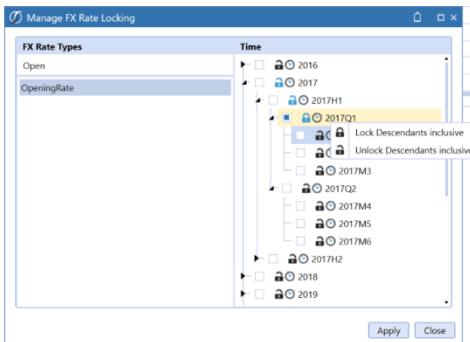
4. In FX Rate Types, select a rate type from the list or enter a type in the search box.
5. In Time, select a year or expand the year to view the full hierarchy.



6. You can navigate the Time structure using the expansion icons. If you select an item using the selection “check box”, it will set the Lock status as a Lock or Unlock icon. Also, any lock that appears blue indicates that a descendant beneath it has a differing “Lock” status.
  - If an item is Locked, the padlock is locked.
  - If an item is Unlocked, the padlock is unlocked.
  - If all the descendants are in the same lock state, the padlock is black.
  - If any of the descendants are in a different state, the padlock is blue.
7. You can lock an entire tree or lock or unlock specific items.
8. You can right-click to apply to a hierarchy and choose Lock Descendants inclusive to lock all within the time frame or Unlock Descendants inclusive to unlock all the within the time frame.

## Cubes

---



9. Click **Apply** and then click **Close**.
10. The FX Rates Grid is displayed. If the rate type and period is locked, you will see a green background and a lock icon in the corner of the cell.

**Foreign Exchange Rates**

**Grid Settings**

	AverageRate	OpeningRate	ClosingRate	HistoricalRate	AverageRate_Budg
2017	🔒				
2017H1	🔒				
2017Q1	🔒				
2017M1	🔒				
2017M2	🔒				
2017M3	🔒				
2017Q2	🔒				
2017M4	🔒				
2017M5	🔒				

## BRAPI Rule Functions

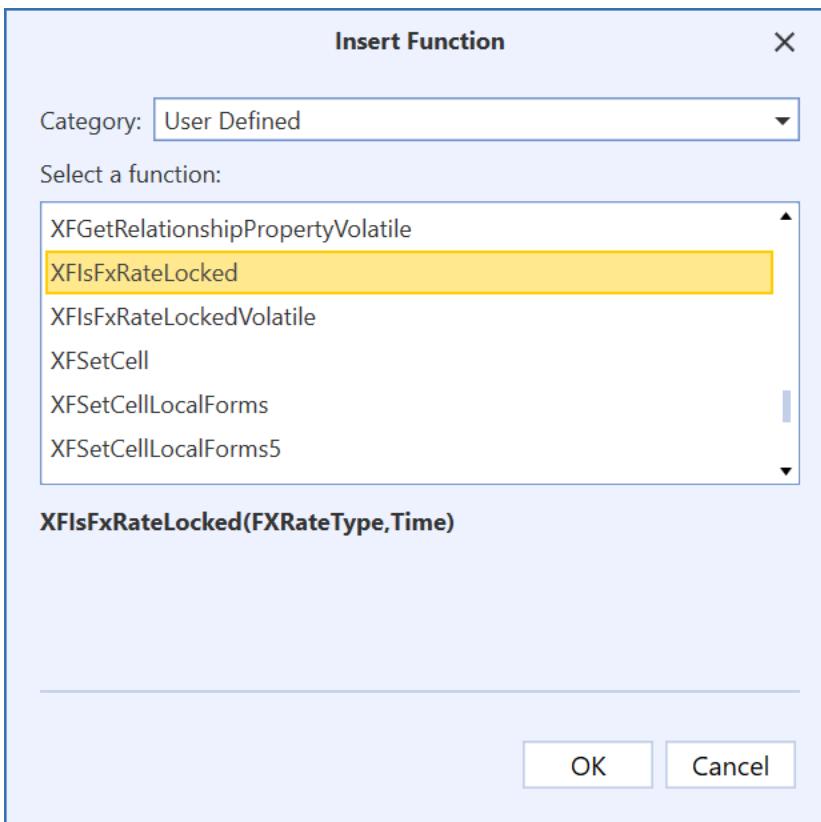
There are three BRAPI's that allow for automation and dashboard development, ISFxRateLocked, LockFxRate, and UnlockFxRate.

- **x= IsFxRateLocked**
- **x= LockFxRate**
- **x= UnlockFxRate**

## Spreadsheet or Excel Functions

Retrieve the status in reports. In spreadsheet or Excel, you can go to functions and select XFIsFXRateLocked for your rate sheets.

1. From the Application tab, go to **Tools > Spreadsheet**.
2. Click the **OneStream** tab.
3. Click the **Insert Function** button.
4. In Category, click **User Defined**.
5. In Select a function, click **XFIsFxRateLocked**.



6. Click **OK**. The Function Arguments window opens.
7. Enter a value in FxRateType.
8. Enter a value in Time.
9. Click **OK**. The spreadsheet will populate with the new data.

**NOTE:** If you change the lock or unlock rate types in the spreadsheet, when you Refresh the application and return to the Rates grid, the data will be updated to reflect your changes in the spreadsheet.

## Member Filters

A Member Filter can be used to filter the data down to what an administrator wants to see by creating a list of restricted Members. Member Filters contain multiple Member Scripts separated by commas.

## Cubes

---

A Member Script is a brief statement typed to query one defined set of Dimensional Members. Members can be specified for any or all Dimensions and the primary Dimension can also specify a Member Expansion Formula (e.g., .Descendants).

Here is an example of a simple Member Script that returns the year 2012:

T#2012

Here is an example of a Member Script with a Member expansion that returns all the Income Statement Accounts.

A#[Income Statement].Descendants

If one or just a few Dimensions in the Member Script are qualified, the remaining Dimensions are pulled either from the Cube View POV, the Global POV, or Workflow, Time, and Scenario. Separate each Dimension qualified in the Member Script with a colon. Here is an example of a fully qualified Member Script:

Cb#GolfStream:E#Houston:P#Texas:C#Local:S#Budget:T#2012M3:V#YTD:A#60000:F#None:  
O#Forms:  
IC#None: U1#Sales: U2#HybridXL: U3#Northeast: U4#SportsCo: U5#None: U6#None:  
U7#None: U8#None

Here is an example of a Member Filter made up of three different Member Scripts that returns the Actual, Budget and Forecast Scenarios.

S#Actual, S#Budget, S#Forecast

## Member Script Abbreviations and Examples

There are 18 Dimensions that can be used to filter data. This section briefly explains each Dimension and gives an example. See Dimension for a full description of each Dimension. Square brackets are optional around Member names, but are mandatory if the Member name contains a space, period, comma, or semi-colon.

### **Cube**

Cb#GolfStream

The specific Cube being referenced.

### **Entity**

E#Houston

The specific Entity being referenced.

## Cubes

---

### **Parent**

P#Texas

The Parent of the Entity being referenced. This is important because Entities can be rolled up into multiple hierarchies.

### **Consolidation**

C#Local

The specific Member of the Consolidation Dimension being referenced.

### **Scenario**

S#Actual

The specific Scenario being referenced.

### **Time**

T#2012Q1

Specific time period in focus. There are several selections available relative in nature to certain boundaries like the time period set in the POV, the selected Workflow time period, or the Global time period. More of these constants are explained in the Member Script Constants section later in this document.

### **T#POV**

This returns the current period of the selected year in the Cube POV.

### **T#W1**

This returns the first week of the selected year in the Cube POV (this only applies to weekly applications).

### **T#M1**

This returns the first month of the selected year in the Cube POV.

### **T#Q1**

This returns the first quarter of the selected year in the Cube POV.

### **T#H1**

This returns the first half of the selected year in the Cube POV.

### **View**

V#YTD

The specific view of the requested information.

### **Account**

A#60000

The specific Account being referenced.

### **Flow**

F#BegBal

Specific Flow Member being referenced. There is a dedicated Flow Dimension.

### Origin

O#Import

The Origin of the data being referenced and then deciding if it was imported, entered as a Journal, or entered into a Form.

### UD1 to UD8

U2#[Fairway Woods] or U3#TotalRegions

Specific Members from the eight User Defined Dimensions are being referenced. If all eight of these Dimensions are not being used, they still need to be qualified in the POV, in a Cube View, or in the Member Script to get a proper intersection. Each User Defined Dimension has a Member called None or the reserved word All that can be used (e.g. U5#All) in some areas of the product such as Intercompany Reporting settings.

## Member Script Constants

Member Script Constants are used as a point of reference when querying data for Members and data. Below are the three constant types:

### Point of View (POV)

Select Members based on their literal or relative position to the selected main Point of View (POV) or the Cube View POV. The main POV settings are available under the Application Tab| Tools | Application Properties. Call out a Dimension to pull its Member value from what is selected in the POV in this way: E#POV, C#POV, T#POV, V#POV, A#POV,... (all Dimension types).

### Workflow (WF)

Refer to the time period or Scenario currently selected in the Workflow module. Examples are T#WF and S#WF.

### Global

The Application has a Global Time and Global Scenario setting found under the Application Tab|Tools|Application Properties. These settings can be used optionally and can be referenced (e.g., T#Global and S#Global). An example of using this as a reference is to build a Cube View and have Global Time and everything before it appear in columns and the Actual Scenario and everything after it appear as the Forecast Scenario.

## Member Expansion Functions

Additional commands can be added at the end of a Member Script to help expand the presented results. For example, E#US will return just the US Entity, while E#US.Descendants will return all of the child Entities that fall under the US Entity such as California, Texas, New York, etc.

## Member Expansions

To illustrate these expansion concepts, the following examples will use a portion of the Entity structure from GolfStream Corporation, which is a golf supply company. This is the management rollup, but there is also a geographical rollup where Houston rolls up to Texas, Carlsbad rolls up to California, Frankfurt rolls up to Europe, etc.

- Total GolfStream
  - Clubs
    - NA Clubs
      - Canada Clubs
        - Montreal
        - Quebec City
      - US Clubs
        - Augusta
        - Carlsbad
        - Houston
    - Frankfurt
  - Golf Balls
    - Europe Golf Balls
    - NA Golf Balls
  - Accessories & Apparel
  - Course Management
  - Corporate

Add these expansions after a Member name to return the desired Members like this:  
E#Houston.Ancestors.

## Cubes

---

**TIP:** Use square brackets, [ ], to reference any name with a space. For example, E#[NA Clubs].

### Member

This lists the Member requested. E#Frankfurt.Member is the same as stating E#Frankfurt.

### Base

E#[NA Clubs].Base queries only the base level Entities and excluding any other aggregate Members:

- Montreal
- Quebec City
- Augusta
- Carlsbad
- Houston

### Children

E#Clubs.Children returns the first level children in a flat list:

- NA Clubs
- Frankfurt

### ChildrenInclusive

E#[NA Clubs].ChildrenInclusive returns both NA Clubs and its children:

- NA Clubs
  - Canada Clubs
  - US Clubs

This can be chained together with another .Children statement to see the next level as well. This may also be referred to as Children.Children elsewhere in the documentation.

E#[NA Clubs].ChildrenInclusive.Children returns:

## Cubes

---

- NA Clubs
  - Canada Clubs
    - Montreal
    - Quebec City
  - US Clubs
    - Augusta
    - Carlsbad
    - Houston

### **Descendants**

E#[NA Clubs].Descendants lists every Member under NA Club in a flat, non-hierarchical list excluding NA Clubs:

- Canada Clubs
- Montreal
- Quebec City
- US Clubs
- Augusta
- Carlsbad
- Houston

### **DescendantsInclusive**

E#[US Clubs].DescendantsInclusive lists every Member under US Clubs in a flat, non-hierarchical list including US Clubs:

- Augusta
- Carlsbad
- Houston Heights
- South Houston

## Cubes

---

- Houston
- US Clubs

### Tree

E#[US Clubs].Tree returns the specified Member and all Members below it in a hierarchy:  
This expansion automatically includes expandable rows.

- US Clubs
  - Augusta
  - Carlsbad
  - Houston

### TreeDescendants

E#Clubs.TreeDescendants lists every Member under Clubs in hierarchical tree excluding Clubs:

- NA Clubs
  - Canada Clubs
    - Montreal
    - Quebec City
  - US Clubs
    - Augusta
    - Carlsbad
    - Houston
- Frankfurt

### TreeDescendantsInclusive

E#[US Clubs].TreeDescendantsInclusive lists every Member under US Clubs in a hierarchical tree including US Clubs:

## Cubes

---

- US Clubs
  - Augusta
  - Carlsbad
  - Houston

### Parents

E#Houston.Parents returns the direct Parents of the given Member regardless of how many hierarchies to which the Member belongs:

- US Clubs (from the management rollup)
- Texas (from the geographical rollup)

The Parent is derived from the Cube View's POV setting by default. If a Member is used in multiple hierarchies, specify a specific Parent using the following syntax in order to override it: E#Houston.Base:P#USClubs or E#Houston.TreeDescendants:P#Texas.

**NOTE:** P# works differently when using it with an expandable Tree filter because that filter processes the children as individual queries, so the P# will only apply to the top-level Members. Use E#Houston.TreeDescendants:P#Texas instead, or specify a Parent on the Cube View's POV settings.

### Ancestors

E#[NA Clubs].Ancestors returns all Members up the chain from NA Clubs:

- Total GolfStream
- Clubs

### Branch

Expand multiple Members of a given expansion by finding specific items and then performing additional expansions on those items. Refer to the Samples Tab in the Member Filter Builder for an example of the syntax.

E#[Total GolfStream].Children.Branch(Find([Clubs]).Children, Find(Golf Balls).Children) returns Total GolfStream's Children, it then finds Clubs' and Golf Balls' Children and returns them in the hierarchy as well.

## Cubes

---

- Total GolfStream
  - Clubs
    - NA Clubs
    - Frankfurt
  - Golf Balls
    - Europe Golf Balls
    - NA Golf Balls
  - Accessories & Apparel
  - Course Management
  - Corporate

This is also used with Quick Views in order to expand several hierarchies at one time.

### Find

Find will apply Member expansions to a nested subset of results:

A#[Income Statement].Descendants.Find(64000).Children will return all of the descendants of the Income Statement as a flat list, but will find account 64000 (i.e. Earnings Before Taxes in this example) and then indent and present that account's children.

### FindAt

This returns a specific Member of a given expansion using a zero-based position index and performing an additional expansion on the specific item. Refer to the Samples Tab in the Member Filter Builder for an example of the syntax.

E#[Total GolfStream].Children.FindAt(1).Children returns Total GolfStream's Children, locates the Member in the first position, and returns its children as well.

- Total GolfStream
  - Clubs
  - Golf Balls

## Cubes

---

- Europe Golf Balls
- NA Golf Balls
- Accessories & Apparel
- Course Management
- Corporate

### First

First will find the first items in the list of results and allow additional Member Filters to be applied:

E#[Total GolfStream].Children.First.Children will list all of the children of Total GolfStream, go to the first Entity (Clubs) and then show its children:

- Clubs
  - NA Clubs
  - Frankfurt
- Golf Balls
- Accessories & Apparel
- Course Management
- Corporate

### Last

Similar to First, Last will find the last items in the list of results and allow additional Member Filters to be applied.

### Keep

Keep will search the results from a Member Filter and only keep certain values:

E#[Total GolfStream].Children.Keep(Clubs, [Golf Balls]).Children will list all of the first level children of Total GolfStream, only keeping Clubs and Golf Balls and then show their children:

## Cubes

---

- Clubs
  - NA Clubs
  - Frankfurt
- Golf Balls
  - Europe Golf Balls
  - NA Golf Balls

### Remove

This will remove some of the Members from the results:

E#[Total GolfStream].Children.Remove(Corporate).Find(Clubs).Children.Find([NA Clubs]).Children will list several Entities and then remove the Corporate Entity from the results:

- Clubs
  - NA Clubs
    - Canada Clubs
    - US Clubs
  - Frankfurt
- Golf Balls
- Accessories & Apparel
- Course Management

### List

This will create a list of specific Members:

A#Root.List(63000, 64000) will return these two accounts.

An Indent Level can also be specified when defining a list of Members. Refer to the Samples Tab in the Member Filter Builder for an example of the syntax.

E#[Total GolfStream].List(Clubs.IndentLevel(0), [Golf Balls].IndentLevel(2), [Course Management].IndentLevel(1))

## Cubes

---

- Clubs
  - Golf Balls
  - Course Management

### Where

The Where clause in a Member Filter will allow further qualification of the results. Use AND, OR and Parentheses to provide detailed conditions for including Members. Use the IN or NotIN qualifiers to see if Members belong to a certain list:

E#Root.Children.Find(US).Children.Find(Michigan, Texas).Children.Where((Name = Flint) Or (Name = Rochester) Or (Name = Dallas)) will return Michigan and Texas and only specific child city Entities

E#Root.Children.Find(US).Children.Find(Michigan, Texas).Children.Where(Name In Flint, Rochester, Dallas) will return Michigan and Texas and only a few child Entities.

See Member Expansion Where Clause later in this section for more examples.

### Options

Use this to reference Sub-Cubes and specify how Dimensions should be processed. Options must immediately follow the expansion function for which it is being used. For example, A#19999.Children might return a different list when looking at an extended dimension associated with a different cube.

A#19999.Options(Cube=[Total GolfStream], ScenarioType=Actual,  
MergeMembersFromReferencedCubes=False)

### Combined Expressions

Stack multiple Member Expressions to display Children and Parent Members. The example below is using the List and Find expressions to display Base and Parent Members.

E#[Total GolfStream].List(Clubs, Corporate, Frankfurt).Find(Clubs).Base.Find(Carlsbad).Parents

- Clubs
- Montreal
- Quebec City
- Augusta
- Carlsbad

## Cubes

---

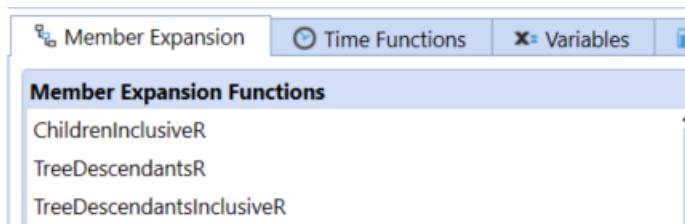
- California
- US Clubs
- Total Product Rollup
- Houston Heights
- South Houston
- Frankfurt
- Corporate
- Frankfurt

For more Member Expansion examples, refer to the Samples Tab in the Member Filter Builder.

## Reverse Order Member Expansions

Functionality to support reversing the direction of the order of results from certain member filters are supported in Cube Views and Quick Views. These reverse order selections include ChildrenInclusiveR, TreeDescendantsR, and TreeDescendantsInclusiveR.

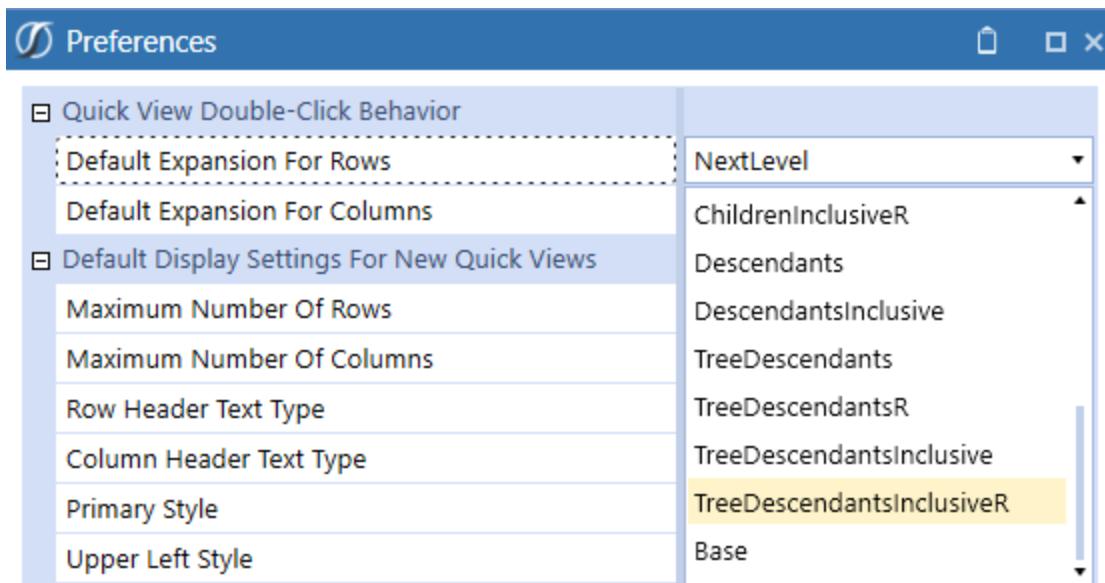
In Cube Views, this functionality can be set in the Member Filter Builder using the Member Expansion Functions tab for Rows (and/or) Columns.



In Quick Views, this can be set in Preferences under the Quick View Double-Click Behavior section in the Default Expansion for Rows (and/or) Columns.

## Cubes

---



This example will demonstrate the TreeDescendantsR being used in a Quick View. This Gross Income view below has been created using Next Level expansion (available in a Quick View):

	Jan 2018	Feb 2018
61000 - Gross Income	27,462,816	28,236,051
60999 - Net Sales	56,155,448	56,119,871
60000 - Operating Sales	57,982,342	57,944,272
2000_100 - Third Party Sales	50,945,236	50,985,236
2000_200 - OEM Sales	4,606,157	4,536,127
2000_300 - Subassembly Sales	382,840	382,800
2000_400 - Parts Sales	2,048,108	2,040,108
60100 - IC Sales	15,000	15,990
60200 - Returns & Allowances	1,841,894	1,840,391
2200_100 - Third Party Returns & Allowances	728,117	727,974
2200_200 - OEM Returns & Allowances	37,372	37,312
2200_300 - Subassembly Returns & Allowances	95,952	95,052
2200_400 - Parts Returns & Allowances	980,454	980,054
60250 - Other Outside Sales		
43000 - Cost of Goods Sold	28,692,632	27,883,820
41000 - Operating Cost of Goods Sold	28,677,632	27,868,820
42000 - IC Cost of Goods Sold	15,000	15,000

## Cubes

---

To change the double-click behavior, select Preferences under the Administration menu. In the Quick View Double-Click Behavior section, select the drop-down list for Default Expansion For Rows and select TreeDescendantsR. The result is the reverse direction of the originating account on expansion for rows when using the double click:

	Jan 2018	Feb 2018
2000_100 - Third Party Sales	50,945,236	50,985,236
2000_200 - OEM Sales	4,606,157	4,536,127
2000_300 - Subassembly Sales	382,840	382,800
2000_400 - Parts Sales	2,048,108	2,040,108
60000 - Operating Sales	57,982,342	57,944,272
60100 - IC Sales	15,000	15,990
2200_100 - Third Party Returns & Allowances	728,117	727,974
2200_200 - OEM Returns & Allowances	37,372	37,312
2200_300 - Subassembly Returns & Allowances	95,952	95,052
2200_400 - Parts Returns & Allowances	980,454	980,054
60200 - Returns & Allowances	1,841,894	1,840,391
60250 - Other Outside Sales		
60999 - Net Sales	56,155,448	56,119,871
41000 - Operating Cost of Goods Sold	28,677,632	27,868,820
42000 - IC Cost of Goods Sold	15,000	15,000
43000 - Cost of Goods Sold	28,692,632	27,883,820
61000 - Gross Income	27,462,816	28,236,051

## Time Member Expansions

These Member expansions can be added onto a regular time-based Member Script. They can be used in a Cube View or in the Form Template's Time Filter for 'Complete Form' option, so only the time periods needed will be displayed.

### AllPriorInYear

This returns all the time periods before the specified time period excluding the specified time for its frequency. T#2012M6.AllPriorInYear, T#WF.AllPriorInYear return the previous periods in that year, but not the specified period.

### AllPriorInYearInclusive

This returns all the time periods before the specified time period including the specified time for its frequency. T#2012M4.AllPriorInYearInclusive returns periods 2012M1 through 2012M4.

## Cubes

---

### **AllNextInYear**

This returns all the time periods after the specified time period not including the specified time for its frequency. T#2012M8.AllNextInYear returns periods 2012M9 through 2012M12.

### **AllNextInYearInclusive**

This returns all the time periods before the specified time period including the specified time for its frequency. T#2012M8.AllNextInYearInclusive returns periods 2012M8 through 2012M12.

### **Weeks**

This returns all the weeks associated with the specified time filter. T#2017M2.Weeks returns all the weeks in M2. T#2017.Weeks returns all the weeks in 2017. If the POV Time is 2017M7, T#POV.Weeks returns all the weeks in M7.

### **Months**

This returns all the months associated with the specified time filter. T#2017.Months returns all the months in 2017. If the POV Time is set to 2017Q2, T#POV.Months returns all the months in Q2.

### **MonthsInQuarter**

This returns the months in the specified quarter. If the Workflow period is 2012M2, T#WF.MonthsInQuarter will return 2012M1, 2012M2, and 2012M3.

### **MonthsInHalfYear**

This returns the months in the half year of the specified period. If the Global period is 2012M2, T#Global.MonthsInHalfYear will return 2012M1 through 2012 M6.

### **MonthsInYear**

This returns all of the months in the year of the specified period. If the POV period is 2012M7, T#POV.MonthsInYear will return 2012M1 through 2012M12.

### **Quarters**

This returns the quarters associated with the specified year. T#2017.Quarters will return 2017Q1, 2017Q2, 2017Q3, 2017Q4.

### **QuartersInHalfYear**

This returns the quarters in the half year of the specified period. If the Workflow period is 2012M3, T#WF.QuartersInHalfYear will return 2012Q1 and 2012Q2.

### **QuartersInYear**

This returns all of the quarters in the year of the specified period. T#2012M7.QuartersInYear will return 2012Q1, 2012Q2, 2012Q3 and 2012Q4.

### **HalfYears**

This returns the half years associated with the specified year. T#2017.HalfYears returns 2017H1 and 2017H2.

## Cubes

---

### **HalfYearsInYear**

This returns the half years in the year of the specified period. If the Global period is 2012M5, T#Global.HalfYearsInYear will return 2012H1 and 2012H2.

### **Prior 1-Prior 12**

This returns the prior period(s) in relation to the specified period. T#2010M12.Prior12 will return the 12 months prior to 2010M12 not including 2010M12. T#2017W40.Prior12 will return the 12 weeks prior to 2017W40 not including 2017W40.

### **Next 1-Next 12**

This returns the next period(s) in relation to the specified period. T#2010M12.Next12 will return the 12 months after 2010M12 not including 2010M12. T#2017W40.Next12 will return the 12 weeks after 2017W40 not including 2017W40.

## **Workflow Member Expansions**

These are used in Cube Views that are used in Reports, Forms or Dashboards and presented during the Workflow process. These Method Queries can also be used when setting pop-up Parameters when Dashboards and Reports are run:

### **WFProfileEntities (Entity Dimensions Only)**

This returns all Entities associated with this Workflow Profile.

### **WFCalculationEntities (Entity Dimension Only)**

Filters to the Entities encompassed within the Workflow Profile Calculation Filter and are calculated/translated/consolidated when the user clicks Process during Workflow.

### **WFConfirmationEntities (Entity Dimensions Only)**

Filters to the Entities encompassed within the Workflow Profile Confirmation Filter and are checked when the user clicks Confirm during Workflow.

### **WFTimePeriods**

This returns the time period(s) associated with the Workflow Profile. When using the Standard Workflow Tracking Frequency, it returns the single time period associated with the selected Workflow Unit. When using other Workflow Tracking Frequencies, it returns the range of time periods for the selected Workflow Unit between WFStartTime and WFEEndTime.

### **WFCalculationTimePeriods**

This returns the time periods needed to be calculated for a Workflow Unit. For example, if a range-based Workflow Unit spans two years, it would return the last period of the first year and the last period of the second year. Executing a calculation using those two time periods would cause all 24 time periods to be calculated because the calculation engine automatically calculates all prior periods in a year. WFCalculationTimePeriods is intended to be used in Data Management Steps and Business Rules.

**WFChannelMembers (User Defined Dimensions Only)**

This returns the UD Dimension Members associated with the Workflow Channel that is associated with this Workflow Profile. This can be a one-to-many relationship. For example, UD1 represents Cost Centers for this application and three UD1 Members are associated with the Workflow Channel called Engineering. If the Workflow Profile in the Workflow View is associated with the Workflow Channel of Engineering, UD1#Root.WFChannelMembers will return those three UD1 Members.

## Member Expansion Where Clause

The Where Clause adds an additional level of filtering after the list of Members has been retrieved. The list of supported fields is as follows:

**Name (e.g., (Name StartsWith 'Sales'))**

A#Sales.Descendants(Name Contains '6000') filters the Member list to include only Member names containing '6000.'

**Description (e.g., (Description Contains 'Costs'))**

A#Root.Descendants(Name Contains 'Costs') filters the Member list to include only account names containing 'Costs.'

**MemberDim**

U2#AllProducts.Descendants(MemberDim='HoustonProducts') filters the Member list to the products in the HoustonProducts Dimension under UD2.

**HasChildren (e.g., (HasChildren = True))**

A#Root.Children(HasChildren=False) filters the Member list to include only the Members that have children.

**InUse**

A#Root.Children(InUse=True) filters the results to only the Members in use.

**AccountType**

A#Root.Base(AccountType = Revenue) filters the results to only the Members under the Account Type of Revenue.

**Formula**

F#Root.TreeDescendants(Formula <> "") filters out Members with an empty value for the Formula property.

**User In Security Group Where Clauses**

These **Where Clauses** will restrict the list to only Members the user is allowed to read and/or write to based on the user security settings. OR expressions can be used to refer to multiple properties:

E#[Total GolfStream].Base.where((UserInReadDataGroup = True) or  
(UserInReadWriteDataGroup = True))

## Cubes

---

Choices:

UserInReadDataGroup  
UserInReadDataGroup2  
UserInReadWriteDataGroup  
UserInAnyDataSecurityGroup  
UserInReadWriteDataGroup2

### **Text1 through Text8**

This uses the Text1 property in the Scenario, Entity, Account, Flow, and UD1 to UD8 Dimensions. There are eight TextN properties in these types of Members that can be used for any need. These Where Clauses will query those fields for their values:

A#Root.Children(Text1='Blue')

A#Root.Children(Text2 StartsWith 'Strategic')

A#Root.Children(Text3 Contains 'Tax')

A#Root.Children(Text4 EndsWith 'Old')

A#Root.Children(Text5 <> "") filters out the Members that have an empty value for the Text5 property.

This is the list of supported operators. Operators such as < are not applicable to the text-based fields.

>=

<=

<>

=

>

<

StartsWith

Contains

DoesNotContain

EndsWith

For more Where Clause examples, refer to the Samples Tab in the Member Filter Builder.

## Time Functions

### Time Functions and Pov Member References

When using the POV as a point of reference, use the following constants in the Member Scripts. In the constants below, N refers to the number of periods desired.

#### **T#POV (All Dimension Types)**

Select Members based on their literal or relative position to the selected main Point of View (POV) or the Cube View POV. The main POV settings are available under the OnePlace module. Call out a Dimension in order to pull its Member value from what is selected in the POV in this way: E#POV, C#POV, T#POV, V#POV, A#POV, ... (all Dimension types).

#### **T#PovYear (Uses Time Member Suffix, e.g., T#PovYearM1)**

A specific time period within a current POV year defined by the user's POV. If POV is equal to 2013M3, then T#POVYear returns 2013. Also, T#POVYearM3 returns the third month of the current POV year.

#### **T#PovPriorYear (Uses Time Member Suffix, e.g., T#PovPrior YearM12)**

A specific time period within the prior year defined by the user's POV e.g., T#POVPriorYearM12

#### **T#PovPriorInYear1 (Uses Integer Suffix, e.g., T#PovPriorInYear1)**

This returns the relative period within the same year. If the current period is 2013M3, and T#POVPriorInYear12 is requested, it returns no value.

#### **T#PovPrior1 (Uses Integer Suffix, e.g. T#PovPrior1)**

T#POVPrior3 returns the time period three periods prior to the current POV value. If 2013M1 is selected, 2012M10 is returned.

#### **T#PovNextInYear1 (Uses Integer Suffix, e.g., T#PovNextInYear2)**

T#POVNextInYear5 returns the month five months in the future from the current POV period selected, but in the same year. If 2013M8 is selected, and T#POVNextInYear9 is requested, it returns no value.

#### **T#PovNext1 (Uses Integer Suffix, e.g., T#PovNext1)**

T#POVNext5 returns the month five months in the future from the current POV period selected. If 2013M8 is selected, and T#POVNext9 is requested, 2014M5 is returned.

#### **T#PovFirstInYear**

This returns the first period in the year portion of the POV period. It is the same as T#POVM1, T#POVQ1, or, T#POVH1 based on the Input Frequency setting of the given Scenario. T#M1, T#Q1, or, T#H1 work the same way as well.

### **T#PovLastInYear**

This returns the last period in the year portion of the POV period.

### **T#POVFirstInMonthPrior1**

POV Month Example

This refers to the POV month and returns N months prior. If the POV is 2017M10, and T#POVFirstInMonthPrior5 is requested, 2017M5 is returned.

### POV Week Example

This refers to the POV week and goes back N weeks from the first week in that month. This varies by calendar (445, 454, 544). In a 445 calendar, if the POV is 2017W38 (M9), and T#POVFirstInMonthPrior5 is requested, the result is driven by the first week in the current month which is 2017W35. This returns 2017W30.

### **T#PovFirstInQuarterPrior1 (Uses Integer Suffix, e.g., T#PovFirstInQuarterPrior 1)**

This refers to the POV sub-period's quarter and goes back N periods from the first period in that quarter. If the period is 2013M5, and T#POVFirstInQuarterPrior2 is requested, 2013M2 is returned.

### **T#PovFirstInHalfYearPrior1 (Uses Integer Suffix, e.g., T#Pov FirstInHalfYearPrior2)**

This refers to the POV sub-period's half year and goes back N periods from the first period in that half year. If the period is 2013M5, and T#POVFirstInHalfYearPrior2 is requested, 2012M11 is returned.

### **T#PovFirstInYearPrior1 (Uses Integer Suffix, e.g., T#Pov FirstInYearPrior3)**

This refers to the POV year and goes back N periods from the first period. If the period is 2013M5, and T#POVFirstInYearPrior1 is requested, 2012M12 is returned.

### **T#PovLastInYearPrior1 (Uses Integer Suffix, e.g., T#PovLastInYearPrior4)**

This refers to the POV year and goes back N periods from the last period. If the period is 2013M5, and T#POVLastInYearPrior1 is requested, 2013M11 is returned.

## Time Functions and Workflow References

### **T#WF (e.g., T#WF or S#WF)**

This refers to the time period or Scenario currently selected in the Workflow module. Examples are T#WF and S#WF.

### **T#WFYear (Uses Time Member Suffix, e.g., T#WFYearM1)**

A specific time period within the current Workflow year. If the Workflow period is equal to 2013M3, then T#WFYear returns 2013. Also, T#WFYearM3 returns the third month of the current Workflow year.

### **T#WFPriorInYear1 (Uses Integer Suffix, e.g., T#WFPriorInYear1)**

T#WFPriorInYear1, T#WFPriorInYear2...returns the relative period within the same Workflow year. If the current Workflow time period is 2013M3, and T#WFPriorInYear12 is requested, it returns no value.

### **T#WFPrior1 (Uses Integer Suffix, e.g., T#WFPrior2)**

T#WFPrior3 returns the time period three periods prior to the current Workflow period. If 2013M1 is selected, 2012M10 is returned.

### **T#WFNextInYear1 (Uses Integer Suffix, e.g., T#WFNextInYear1)**

T#WFNextInYear5 returns the month five months in the future from the currently selected Workflow period, but in the same year. If 2013M8 is selected, and T#WFNextinYear9 is requested, it returns no value.

### **T#WFNext1 (Uses Integer Suffix, e.g., T#WFNext 2)**

T#WFNext5 returns the month five months in the future from the current Workflow period selected. If 2013M8 is selected, and T#WFNext9 is requested, 2014M6 returns.

### **T#WFFirstInYear**

This returns the first period in the year portion of the Workflow period. This is the same as T#WFM1, T#WFQ1, or T#WFH1 based on the Input Frequency settings of the given Scenario.

### **T#WFLastInYear**

This returns the last period in the year portion of the Workflow period.

## Time Functions and Global References

### **T#Global (e.g., T#Global or S#Global)**

The Application has a Global Time and Global Scenario setting found under the Application Tab|Tools|Application Properties. These settings can be used optionally and can be referenced (e.g. T#Global and S#Global). An example of using this as a reference is to build a Cube View and have Global Time and everything before it appear in columns and the Actual Scenario and everything after it appear as the Forecast Scenario.

### **T#GlobalYear (Uses Time Member Suffix, e.g., T#GlobalYearM1)**

A specific time period within the current Global year. If the Global period is equal to 2013M3 then T#GlobalYear returns 2013. Also, T#GlobalYearM3 returns the third month of the current Global year.

### **T#GlobalPriorInYear1 (Uses Integer Suffix, e.g., T#GlobalPriorInYear1)**

T#GlobalPriorInYear1, T#GlobalPriorInYear2...returns the relative period within the same Global year. If the current Global time period is 2013M3, and T#GlobalPriorInYear12 is requested, it returns no value.

### **T#GlobalPrior1 (Uses Integer Suffix, e.g., T#Global Prior 3)**

T#GlobalPrior3 returns the time period three periods prior to the current Global Period. If 2013M1 is selected, 2012M10 is returned.

### **T#GlobalNextInYear1 (Uses Integer Suffix, e.g., T#GlobalNextInYear5)**

T#GlobalNextInYear5 returns the month five months in the future from the current Global period selected, but in the same year. If 2013M8 is selected, and T#GlobalNextInYear9 is requested, it returns no value.

### **T#GlobalNext1 (Uses Integer Suffix, e.g., T#GlobalNext5)**

T#GlobalNext5 returns the month five months in the future from the current Global period selected. If 2013M8 is selected, and T#GlobalNext9 is requested, 2014M6 is returned.

### **T#GlobalFirstInYear**

This returns the first period in the year portion of the Global period. This is the same as T#GlobalM1, T#GlobalQ1, or T#GlobalH1 based on the Input Frequency setting of the given Scenario.

### **T#GlobalLastInYear**

This returns the last period in the year portion of the Global period.

For more Time Function examples, refer to the Samples Tab in the Member Filter Builder.

## **Time Functions and General References**

With these time functions, reference the year separate from the sub-period wherever T# is used. For example, if a user selects a time Member from a Parameter prompt. The user's options are a long list of time periods (e.g., 2012M6). The Parameter is named MyTimeParam.

The Cube View can use the selection like this:

```
T#Year(!MyTimeParam!)Period(!MyTimeParam!)
```

The Year function returns the specified year and the Period function returns the period without the year. The result is a valid time Member because they are combined. If the user selected 2012M6, it looks like this after the Parameter substitution.

```
T#Year(2012M6)Period(2012M6)
```

The result used in the Cube View is:

```
T#2012M6
```

This is an example showing last year's values for the same month:

```
T#YearPrior1(!MyTimeParam!)Period(!MyTimeParam!)
```

which ends up being T#2011M6

## Cubes

---

This is an example showing last month's values for the same year:

T#Year(!MyTimeParam!)PeriodPrior1(!MyTimeParam!)

which ends up being T#2012M5

This is an example showing all base Members for the year:

T#Year(!MyTimeParam!).Base

Here are some additional examples:

T#Year(2012M6)Period(2012M6) = 2012M6 T

The 2012M6 could have started as a substitution variable such as !MyTimeParam! or |PovTime|

T#Year(2012M6)M2 = 2012M2

T#YearPrior(2012M6)Period(2012M6) = 2011M6

T#YearPrior2(2012M6)Period(2012M6) = 2010M6

T#YearPrior2(2012M6) = 2010

T#Year(2012M6)PeriodPrior3(2012M6) = 2012M3

T#Year(2012M6)PeriodPrior8(2012M6) = 2011M10

The PeriodPrior8 caused it to change the year too.

T#Year(2012M6)PeriodPriorInYear8(2012M6) = 2012M1

The PeriodPriorInYear8 does not change the year.

If the Year section is skipped, the year comes from the Period section.

T#PeriodPrior3(2012M6) = 2012M3

Time functions to use this way:

T#Year(): T#Year(!TimeParam!)Period(!TimeParam!)

To look at the Time dimension in your workflow and then count back 2 years prior:

T#YearPrior1(): T#YearPrior2(|WFTime|)

WorkFlow Time Points to: 2022M1

Results: 2020 YTD Net Income Value

To look at the Time dimension in your Global POV and count forward 3 years:

T#YearNext1(): T#YearNext3(|GlobalTime|)

Global Time Points to: 2022M3

Results: 2025 YTD Net Income Value

## Cubes

---

T#Period(): T#Year(!TimeParam!)Period(!TimeParam!)

T#PeriodPrior1(): T#PeriodPrior1(|WFTime|)

T#PeriodNext1(): T#PeriodNext2(|GlobalTime|)

T#PeriodPriorInYear1(): T#PeriodPriorInYear2(2012M6)

T#PeriodNextInYear1(): T#PeriodNextInYear1(|Global Time|)

T#FirstPeriodInQuarter(): T#FirstPeriodInQuarter(|WFTime|)

T#FirstPeriodInQuarterPrior1(): T#FirstPeriodicInQuarterPrior1(2012M6)

T#FirstPeriodInQuarterNext1(): T#FirstPeriodInQuarterNext2(|POVTime|)

T#FirstPeriodInHalfYear(): T#FirstPeriodInHalfYear(|WFTime|)

T#FirstPeriodInHalfYearPrior1(): T#FirstPeriodInHalfYearPrior1(2012M6)

T#FirstPeriodInHalfYearNext1(): T#FirstPeriodInHalfYearNext2(|WFTime|)

T#FirstPeriodInYear(): T#FirstPeriodInYear(|WFTime|)

T#FirstPeriodInYearPrior1(): T#FirstPeriodInYearPrior1(!TimeParam!)

T#FirstPeriodInYearNext1(): T#FirstPeriodInYearNext2(|PovTime|)

T#Quarter(): T#Quarter(|WFTime|)

T#QuarterPrior1(): T#QuarterPrior1(2012M6)

T#QuarterNext1(): T#QuarterNext2(|POVTime|)

### Weekly Functions

The following Time Functions apply to weekly applications. The following examples are based on a 52 Week 445 calendar.

T#FirstPeriodInMonth(): T#FirstPeriodInMonth(2017W43)

Returns: 2017W40

T#FirstPeriodInMonthPrior1(): T#FirstPeriodInMonthPrior3(|WFTime|)

Workflow Time: 2017W1

Returns: 2016W50

T#FirstPeriodInMonthNext1(): T#FirstPeriodInMonthNext2(2017W49)

Returns: 2017W50

T#Month(): T#Month(2017W40)

Returns: 2017M10

## Cubes

---

T#MonthPrior1(): T#MonthPrior3(2017W44)

Returns: 2017M8

T#MonthNext1(): T#MonthNext5(|POVTime|)

POV Time: 2017W9

Returns: 2017M8

Changing the Time Label in Headers When Using These Time Functions

When using this Time function to change a header label in a Cube View, use the :Name("") function and substitute what will be seen. For example, to show the value from the last period from the year prior to the Global Time period, use this function:

This example is using 2012M6 as the current Global Time period.

T#YearPrior1(|GlobalTime|)M12

2011M12 will return in the header of the Cube View, such as a column header. In order to return Year End 2011, use the Name function in this way:

T#YearPrior1(|GlobalTime|)M12:Name("Year End |MFYear|")

The MF in the substitution variable above stands for Member Formula.

These are examples of what similar substitution variable will return:

T#YearPrior1(|GlobalTime|)M12:Name("Year End |MFTime|") returns Year End 2011M12

T#YearPrior1(|GlobalTime|)M12:Name("Year End |MFTimeDesc|") returns Year End Dec 2012

T#YearPrior1(|GlobalTime|)M12:Name("Year End |MFTimeShortDesc|") returns Year End Dec

T#YearPrior1(|GlobalTime|)M12:Name("Year End |MFSubPeriod|") returns Year End M12

## Substitution Variables

Substitution Variables can be used to exchange values at run time for Parameters presented to the user before a Dashboard runs, for Members in the POV, for Global variables for Scenario or Time, or for Workflow variables for Time, Scenario or other Workflow Profile attributes.

## Substitution Variables from Current POV

These settings will return the values from the user's current POV for each Dimension.

By adding Desc to the end of many of these substitution variables, the Description for the Member will be returned instead of the Member name:

|PovEntityDesc| will return Akron instead of the Entity name 123.

## Cubes

---

POVCube and POVCubeDesc  
POVEntity and POVEntityDesc  
POVParent and POVParentDesc  
POVConsolidation and POVConsolidationDesc  
POVScenario and POVScenarioDesc  
POVTime, POVTimeDesc and POVTimeShortDesc  
POVView and POVViewDesc  
POVAccount and POVAccountDesc  
POVFlow and POVFlowDesc  
POVOrigin and POVOriginDesc  
POVIC and POVICDesc  
POVUD1-UD8 and POVUD1-8Desc

**NOTE:** If there is an index out-of-range, the Cube View will display an error showing the data POV as invalid. When the Cube View is passed to the Pivot Grid control, it attempts to resolve the error column. It cannot, so in the case of the Time dimension it will instead display the default time member as "Name - Description". To resolve the issue in the Pivot Grid, you must first correct the issue in the source Cube View.

Substitution Periods are defined within a given Year. The number of sub periods are limited to the number of months or weeks within the given application year. For example, if you go prior or next within a given year, it will stop at the year boundary. If PovSubPeriodPrior14 in a given monthly app for a given year will not go past M1. PovSubPeriodNext will not go past M12. In a weekly app, PovSubPeriodPrior will not go past W1 and PovSubPeriodNext will not go past W53.

### **POVTimeDimProfile**

This returns the Time Dimension Profile name associated with the user's current Cube POV.

### **POVCurrency**

This returns the actual currency for an Entity based on the Consolidation Dimension setting in the POV. For example, if Consolidation is set to Local for the Manchester Entity, GBP will return, which is its currency. If Consolidation is set to Translated and the Parent is set to US, USD will return.

### **POVTimePriorInYearN**

This returns the prior (or more) time period if it is still in the current year.

## Cubes

---

### **POVTimePriorN**

This returns the prior (or more) time period.

### **POVTimeNextInYearN**

This returns the next (or more) time period if it is still in the current year.

### **POVTimeNextN**

This returns the next (or more) time period.

### **POVYear**

This returns the Year portion of a POV Time period.

### **POVYearPriorN**

This returns the Year portion of a Time period for the previous year where N is the number of years.

### **POVYearNextN**

This returns the Year portion of a Time period for the upcoming year where N is the number of years.

### **POVSubPeriod**

This returns the Week, Month, Quarter, or Half Year portion of a Time period. If the POV is 2012M5, M5 is the SubPeriod.

### **POVSubPeriodPriorN**

This returns the previous Week, Month, Quarter, or Half Year portion of a Time period.

### **POVSubPeriodNextN**

This returns the next Week, Month, Quarter, or Half Year portion of a Time period.

### **POVSubPeriodNum**

This returns the sub period's number for the current POV, so if the POV is set to M3, the number 3 is returned.

### **POVTimeFirstInYear**

This is the same as T#POVFirstInYear, T# W1, T#M1, T#Q1, or T#H1 based on the Input Frequency of the Scenario.

### **POVTimeLastInYear**

This returns the last time period in the year based on the Input Frequency of the Scenario.

## **Workflow Substitution Variables**

### **WF**

This is used within the Point of View area for Scenario and Time: T#WF.

### **WFProfile**

This provides the current Workflow Profile name.

## Cubes

---

### **WFProfileIndex**

This provides a numeric value of the Workflow Profile in the hierarchy.

### **WFProfileLastDescendantIndex**

This provides an index of bottom descendants in the Workflow Profile tree.

### **Workflow Profile Keys**

These return a numeric internal ID for Workflow Profiles.

WFProfileKey

WFReviewProfileKeys

WFInputParentProfileKeys

WFImportProfileKeys

WFFormProfileKeys

WFJournalProfileKeys

### **WFSceario**

This returns the Scenario in the Workflow View.

### **WFScearioDesc**

This returns a Description of the Scenario in the Workflow View.

### **WFScearioID**

This returns the Numeric ID of the Scenario in the Workflow View.

### **WFTime**

This returns the Time period associated with current Workflow Unit.

### **WFTimeDesc and WFTimeShortDesc**

This returns the Description (e.g. Feb 2011) or Short Description (e.g. Feb) for Workflow Time as defined in Time Profile.

### **WFTimeID**

This returns the Numeric ID of the Workflow Unit Time period.

### **WFTimePriorInYearN**

This returns the prior (or more) Workflow time period if it is still in the current year.

### **WFTimePriorN**

This returns the prior (or more) time period.

### **WFTimeNextInYearN**

This returns the next (or more) time period if it is still in the current year.

### **WFTimeNextN**

This returns the next (or more) time period.

### **WFYear**

This returns the year portion of the Workflow Unit.

## Cubes

---

### **WFYearPriorN**

This returns the Year portion of a Time period for the previous year where N is the number of years.

### **WFYearNextN**

This returns the Year portion of a Time period for the upcoming year where N is the number of years.

### **WFSubPeriod**

This returns the Week, Month, Quarter, or Half Year portion of a Time period

### **WFSubPeriodPriorN**

This returns the previous Week, Month, Quarter, or Half Year portion of a Time period.

### **WFSubPeriodNextN**

This returns the next Week, Month, Quarter, or Half Year portion of a Time period.

### **WFSubPeriodNum**

This returns the period's number for the current Workflow Period, so if the Workflow is set to M3, the number 3 is returned.

### **WFTimeFirstInYear**

This is the same as T#WFFirstInYear, T#W1, T#M1, T#Q1, or T#H1 based on the Input Frequency of the Scenario.

### **WFTimeLastInYear**

This returns the last time period in the year based on the Input Frequency of the Scenario.

### **WFStartTime**

This is the Workflow Start Time entry for this Scenario.

### **WFStartTimeDesc**

This provides a Description of the Workflow Start Time entry for this Scenario.

### **WFStartTimeShortDesc**

This provides a Short Description of the Workflow Start Time entry for this Scenario.

### **WFEendTime**

This is the Workflow End Time entry for this Scenario.

### **WFEendTimeDesc**

This provides the Description of the Workflow Start Time entry for this Scenario.

### **WFEendTimeShortDesc**

This provides the Short description of the Workflow Start Time entry for this Scenario.

### **WFCubeRoot**

This returns the very top level of the Workflow Profile hierarchy.

## Cubes

---

### **WFCube**

This returns the Cube related to this Workflow Profile.

### **WFTimeDimProfile**

This returns the Time Dimension Profile name associated with the user's current Workflow POV.

### **WFEntityDim**

This returns the Dimension in play for the Workflow Entities assigned to this Workflow Profile.

### **WFScearioDim**

This returns the Scenario Dimension for the Cube assigned to this Workflow Profile.

### **WFAccountDim**

This returns the Account Dimension associated with the Cube for this Workflow Profile.

### **WFFlowDim**

This returns the Flow Dimension for the Cube assigned to this Workflow Profile.

### **WFUD1Dim-WFUD8Dim**

This returns the UD1 Dimension-UD8 Dimension for the Cube assigned to this Workflow Profile.

### **WFText1, WFText2, WFText3 and WFText4**

This provides four optional text fields that can be populated and referenced from a Workflow Profile.

## Global Substitution Variables

### **GlobalScenario**

This is the Global Point of View Scenario from Application Properties.

### **GlobalTime**

This is the Global Point of View Time period from Application Properties.

### **GlobalScenarioDesc**

This provides the Descriptions for the Global Scenario.

### **GlobalTimeDesc and GlobalTimeShortDesc**

This provides the Description (e.g. Feb 2011) or Short Description (e.g. Feb) for Global Time, as defined in the Time Profile.

## Cube View Substitution Variables

These settings will return the values from the user's current Cube View in the Point of View Slider for each Dimension. For the Dimensions not specified in the Cube View's Point of View Slider, the Member will be represented by the user's POV settings.

## Cubes

---

By adding Desc to the end of many of these substitution variables, the Description for the Member will be returned instead of the Member name:

|CVAccountDesc| will return Net Sales instead of 61000

|CVTimeDesc| will return Dec 2011 instead of 2011M12

|CVTimeShortDesc| will return Dec instead of 2011M12

The following Substitution Variables will return the Cube View Name, Cube Name, or Dimension Name associated with the specific Cube View as well as a Description.

CVName and CVDesc

CVCube and CVCubeDesc

CVTimeDimProfile

CEntity and CEEntityDesc

CVParent and CVParentDesc

CVConsolidation and CVConsolidationDesc

CVScenario and CVScenarioDesc

CVTime, CVTimeDesc and CVTimeShortDesc

CVView and CVViewDesc

CVAccount and CVAccountDesc

CVFlow and CVFlowDesc

CVOrgin and CVOrginDesc

CVIC and CVICDesc

CVUD1-8 and CVUD1-8Desc

### **CCurrency**

This returns the actual currency for an Entity based on setting in the specific Cube View.

### **Additional Cube View Time Functions**

See Substitution Variables from Current POV to see examples of these Time Functions.

CVTimePriorInYear

CVTimePrior

CVTimeNextInYear

CVTimeNext

CVYear

CVYearPrior

CVYearNext

CVSubPeriod

CVSubPeriodPrior

CVSubPeriodNext

CVSubPeriodNum

CVTimeFirstInYear

CVTimeLastInYear

## General Substitution Variables

The script or variable will use pipe characters to include a pre-defined substitution variable, e.g. |UserName|. These can be used in Business Rules, Cube Views, and Dashboard headers.

### **Null**

|Null| is empty text. |Null| is mostly used in Cube-level security and the Stage parser. It can also be used within a comma-separated list of Parameter values when you need to set a value to be an empty string. However, |Null| is not a commonly used substitution variable.

### **Space**

This is used to replace a string with nothing or use a space along with the Name function in a Member Filter. For example, in order to make a Column Header or Row Header blank, use T#POV:Name(" ") or T#POV:Name(|space|). Both will produce the same result.

### **UserName**

This provides the current user name

### **UserText1-4**

This allows the reference of the Text1 through Text4 properties related to a User account:  
|UserText3|.

### **AppName**

This provides the application name

### **DateForFileName**

This returns the current date and time: 20131208\_102540.

### **DateForFileName**

This returns the current date: 20131208.

### **DateLong**

This returns the current date: Sunday, December 08, 2013.

### **DateMMDDYYYY**

This returns the current Date as Month, Day, Year: 12/08/2013.

### **DateDDMMYYYY**

This returns the current Date as Day, Month, Year: 08/12/2013.

### **DateYYYYMMDD**

This returns the current Date as Year, Month, Day: 2013/12/08.

### **DateTimeHHMMSS**

This returns the current Date/Time as Hour, Minutes, Seconds: 11:00:19.

## Cubes

---

DateTimeForFileNameUTC

DateForFileNameUTC

DateLongUTC

DateMMDDYYYYUTC

DateDDMMYYYYUTC

DateYYYYMMDDUTC

DateTimeHHMMSSUTC

The version of these functions with the UTC suffix returns the same result, but in Coordinated Universal Time.

## Member Filter Substitution Variables

In order to change how a Member appears in results, use the MF functions below. See the section on Changing the Time Label in Headers When Using These Time Functions under Time Functions for several examples of how these are used. These are used in Name and GetDataCell functions only.

MFTime

MFTimeDesc

MFTimeShortDesc

MFYear

MFSUBPeriod

MFSUBPeriodNum

The following Substitution Variables work with the XFMemberProperty function in order to retrieve any Dimension Member Name being used within a Member Filter. Add Desc to any of these Substitution Variables in order to display the Member Description instead of the Member Name. See XFMemberProperty under Commonly Used Member Filter Functions for more details on using these Substitution Variables.

## Cubes

---

|MFCube|  
|MFEntity|  
|MFPARENT|  
|MFConsolidation|  
|MFSceNario|  
|MFTime|  
|MFView|  
|MFAccount|  
|MFView|  
|MFFlow|  
|MFOrgin|  
|MFIC|  
|MFUD1|-|MFUD8|

Loop1-4Variable  
Loop1-4DisplayVariable  
Loop1-4Index  
Variable1-10

See "Presenting Data With Books, Cube Views and Dashboards" on page 547.

## Custom Substitution Variables

For user-defined (i.e., custom) Substitution Variables, the user will enclose the variable name using both pipes and exclamation points: `!|myVariable|!`

This excludes the Member Filter Substitution Variables listed above.

A good use of this is for Parameters added to a Dashboard. A Parameter called `SalesDashboardEntity` might be created and prompt the user with a list of all Entities to choose from before filtering a Report. In the Cube View that drives this Report, limit the Cube View's POV to the selected Parameter Entity by entering `!|SalesDashboardEntity|!`. When designing a Dashboard, `!|SalesDashboardEntity|!` can also be entered in the title of the Dashboard Component in order to be displayed at run time.

## Commonly Used Member Filter Functions

### **GetDataCell**

Use the `GetDataCell` function to retrieve specific cell(s), perform math, or Business Rule operations.

### **Variance Example using GetDataCell**

`GetDataCell(Variance(S#Actual,S#BudgetV1)):Name(Variance)`

### BetterWorse Example Using GetDataCell

This provides a Variance taking Account Type into consideration

```
GetDataCell(BWDiff(S#Actual, S#BudgetV1)):Name("BetterWorse Difference")
```

### VariancePercent Example Using GetDataCell

```
GetDataCell(VariancePercent(S#Actual,S#BudgetV1)):Name(Var %)
```

### BetterWorsePercent Example Using GetDataCell

This provides a Variance Percent taking Account Type into consideration

```
GetDataCell(BWPercent(S#Actual, S#BudgetV1)):Name("BetterWorse %")
```

### Business Rule Example using GetDataCell

```
GetDataCell(BR#[BusinessRuleName, FunctionName]):Name(Custom Function)
```

For more GetDataCell examples, refer to the Samples Tab in the Member Filter Builder.

### Parameter/Parameter Display

Use one of these buttons in order to enter a custom Parameter reference that comes from either a Form or Dashboard. The Parameter Display Substitution Variable is only used when working with a Delimited List Parameter where the Display Item will be displayed instead of the Member name.

### Business Rules

Business Rules can be passed in a Cube View Member Filter in order to do complex calculations on the Members referenced in the Cube View. Setup the Finance Business Rule to indicate the name of the function and any name-value pairs to reference in the Member Filter. A different action can be performed based on the definition of the name-value pair. In the example below, the Business Rule is going to read the current time period from the rule and get the value for the prior year based on the Member script.

### Function in Business Rule

The Function name in this string is PYMonthForCol. This needs to be referenced in the Member Filter.

```
If functionName.Equals("PYMonthForCol", StringComparison.InvariantCultureIgnoreCase) Then
```

### Name-Value Pair

The Name-Value Pair in this string is Field1. This needs to be referenced in the Member Filter and defined. Based on the Name-Value Pair, the rule can run different actions.

```
Dim ms1 As String = args.DataCellArgs.NameValuePairs("Field1")
```

```
Dim priorYearTimeName As String = api.Time.GetNameFromId(api.Time.GetPriorYearPeriodId())
Dim memberScript As New System.Text.StringBuilder
memberScript.Append(ms1)
memberScript.Append(":T#")
memberScript.Append(priorYearTimeName)
```

Get the Data Cell for the Prior Year

## Cubes

---

```
    Return api.Data.GetDataCell(memberScript.ToString)
End If
End Select
```

### Cube View Member Filter Syntax

In the GetDataCell string define the Business Rule's name, the function's name, and define the name-value pair from the Business Rule.

BR#[BusinessRuleName, FunctionName=yourFunctionName, Name1=Value1, AnotherName=[AnotherValue]]

#### Example

```
GetDataCell("BR#[BRName = XFR_CVDataCellHelperNew, FunctionName = PYMonthForCol,
Field1 =A#60999]"):Name("Sales Last Year")
```

**NOTE:** See the Samples tab in the Member Filter Builder for more examples on this function.

### Custom Member List

Build a custom list in a Business Rule and pass that Business Rule in a Cube View. This can return a list of Member's or Member information when running the rule in a Cube View.

MemberInfo allows users to specify additional information such as a Parent Entity ID and an Indent Level. An unlimited number of name-value pairs can be passed to the Member List Business Rule. The example below will put the member list in alphabetical order.

In the Business Rule, define the List Name and the name-value pairs:

```
Select Case api.FunctionType
Case Is = FinanceFunctionType.MemberList
    Dim listName As String = args.MemberListArgs.MemberListName
    Dim entityList As String = args.MemberListArgs.NameValuePairs("EntityList")
    If listName.Equals("EntityParentList", StringComparison.InvariantCultureIgnoreCase) Then
        Dim objMemberListHeader = New MemberListHeader(args.MemberListArgs.MemberListName)
```

### Cube View Member Filter Syntax

In the Cube View Member Filter, define the Business Rule's name, the Member List Name, define the name-value pairs from the Business Rule, and the Members to which it applies.

E#Root.CustomMemberList(BRName = XFR\_MemberListBasicNew, MemberListName = EntityParentList, EntityList = E#Houston.Base)

**NOTE:** See the Samples tab in the Member Filter Builder for more examples on this function.

## Cubes

---

### Custom Display Names in Member Lists

When creating custom Member Lists, custom display names can be applied and displayed in a Cube View header.

Business Rule Example:

```
Select Case api.FunctionType
    Case Is = FinanceFunctionType.MemberList

        Dim listName As String = args.MemberListArgs.MemberListName

        'Get the Passed in parameters
        Dim entityList As String = args.MemberListArgs.NameValuePairs("EntityList")

        If listName.Equals("AlphaSortList", StringComparison.InvariantCultureIgnoreCase)
    Then Dim objMemberListHeader = New MemberListHeader(args.MemberListArgs.MemberListName)

        'Read the members
        Dim objMemberInfos As List(Of MemberInfo) = api.Members.GetMembersUsingFilter

        (args.MemberListArgs.DimPk, entityList, Nothing)

        'Sort the members
        Dim objMembers As List(Of MemberInfo) = Nothing

        If Not objMemberInfos Is Nothing Then
            objMembers = (From memberInfo In objMemberInfos Order By memberInfo.Member.Name
            Ascending Select memberInfo).ToList()
        End If

        'Apply a custom display name to the first item.
        If Not objMembers Is Nothing Then
            If objMembers.Count > 0 Then
                objMembers(0).RowOrColNameForCalcScript = objMembers(0).NameAndDescription &
                "(Custom Name)"
            End If
        End If

        'Return list
        Return New MemberList(objMemberListHeader, objMembers)

    End If

End Select
```

### XFMemberProperty

This function allows users to specify a Dimension property and display the Member Property selection as a row or column header on a Cube View. Use this function with the Name() and GetDataCell() portion of a Member Filter.

The following name-value pair settings can be used for this function:

- DimType  
Dimension name such as Entity, Account, etc.
- Member  
Dimension Member name

**NOTE:** In order to reference the Dimension Members specified in the Member Filter, use the Dimension-specific Member Filter Substitution Variables. For example, if the Member Filter is A#NetSales.Children, use =|MFAccount| to dynamically point to each Child Member of Net Sales while running this function. See Member Filter Substitution Variables for more details. See below for an example of this syntax.

- Property  
The exact Dimension property name
- VaryByCubeType  
Use this if the property varies by a specific Cube
- VaryByScenario  
Use this if the property varies by Scenario such as =Actual, =Budget, etc.
- VaryByTime  
Use this if the property varies by a specific Time Period such as =2016M1, =2016M5, etc.
- TimeDimProfile  
This can be set to CV, WF, any Time Dimension Profile name, or a Cube name using the CB# qualifier

Example Syntax:

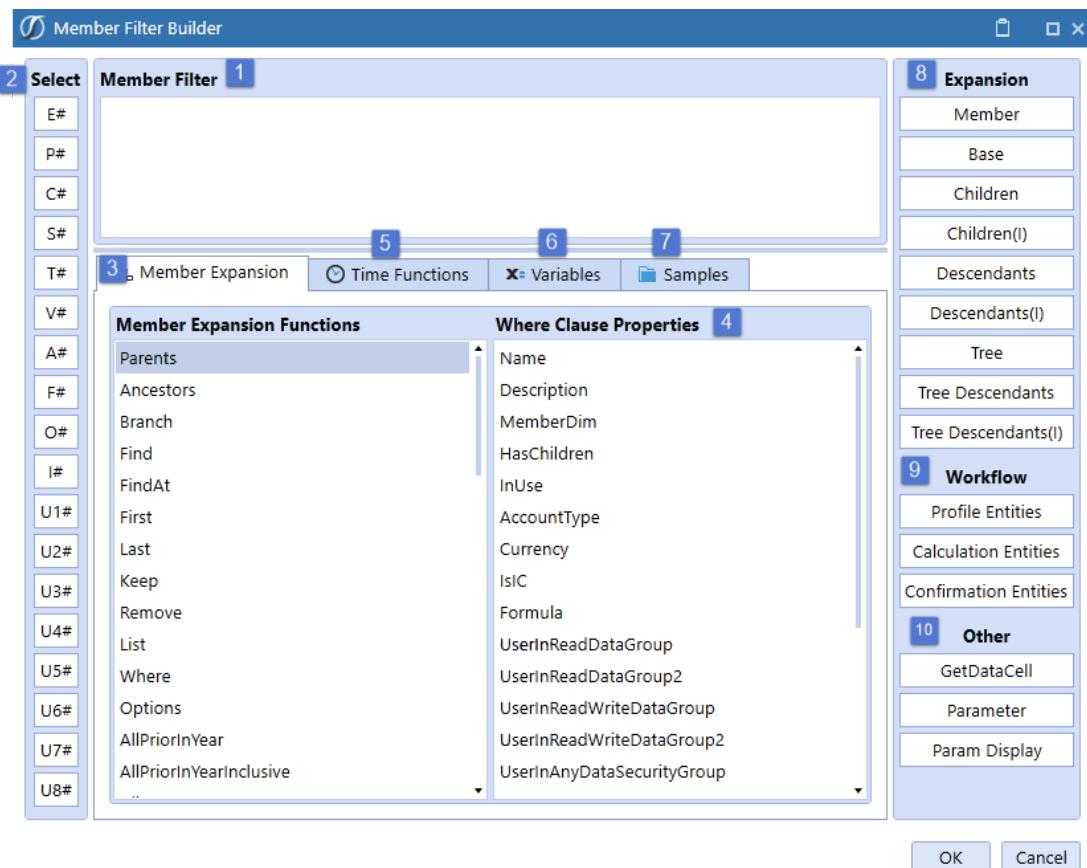
T#2015.Base:Name(XFMemberProperty(DimType=Time, Member=|MFTime|, Property=Description, TimeDimProfile=|CVTimeDimProfile|))

The example above uses the |MFTime| Substitution Variable in order to reference the Members in the T#2015.Base Member Filter. See Member Filter Substitution Variables for more details on these Substitution Variables. It also uses a Substitution Variable for TimDimProfile in order to point to whatever Time Dimension Profile is being used on the Cube View.

**NOTE:** For more examples on using this function, refer to the Samples tab in the Member Filter Builder.

## Member Filter Builder

The dialog makes it simple to build complex Member Filters without having to remember or look up the proper syntax. This dialog is embedded in areas such as Cube Views and the Dimension Library.



### 1. Member Filter

This area is where the Member Filter will be built. Type in this section or use the dialog to help fill it out.

### 2. Member Selection

There is a button for each Dimension here that will launch the appropriate selection dialog.

### 3. Member Expansion Functions

Double-click on a Member Expansion to add it.

### 4. Member Expansion Where Clause

If the Member Expansion Where is selected, use the Where Clause Properties to complete the expression.

Example: UD2#AllProducts.Children.Where(Name Contains Clubs)

### 5. Time Functions

These only apply to the Time Dimension, such as T#POVPrior1. Double-click on a Time Function to add it.

### 6. Substitution Variables

Double-click on a system wide Substitution Variable to add it.

### 7. Samples

Refer to this tab for examples on how to build Member Expansions, Time Functions, Where Clause Expressions, GetDataCell Expressions, and Custom Member List Expressions.

### 8. Expansion

These buttons are commonly used Member Expansions. Click the Expansion to add it.

### 9. Workflow

These buttons are commonly used Workflow Member Expansions used in Cube Views that point to a Report, Form, or Dashboard and are affiliated with a specific Workflow Profile.

### 10. Other

These buttons are commonly used Member Filter Functions which allow the user to create calculated columns and rows or use a custom Parameter to store Member lists.

# Workflow

A primary feature of the OneStream platform is the ability to tailor and optimize all aspects of an application to best fit the requirements of the model design, such as Consolidation, Planning, Forecasting or Operational models. In this section you will learn about the various import methods, workflow analysis, and the blend workflow.

---

## Workflow

The Workflow Engine is the coordinator of all activity in the system. It protects you from having to deal with the complexities of an advanced Multi-Dimension Analytic Model. Profiles and Channels are the basic building blocks of a solid workflow management structure. Data Units represent units of work for loading, clearing, calculating, storing, and locking data within the Multi-Dimensional engine.

## Workflow

Workflow is the overall system manager coordinating all end user activities while guaranteeing the quality and transparency of source data used to feed the Analytic Models contained in OneStream applications.

The Workflow Engine's primary responsibilities are to:

- Protect the end user from Analytic Model complexities.
- Manage and audit data collection.
- Manage and enforce the quality process along with the data certification process.
- Manage and intelligently coordinate the data consolidation process.
- Deliver right-time information and guided analysis.
- Create a standardized end-user experience with the capability of becoming personalized through guided Workspaces.

The primary reason Workflow exists is to care and feed Analytic Cubes. Therefore, before a Workflow hierarchy can be created, at least one Cube marked as Top Level for Workflow must exist in the application.

A Cube is an Analytic Model that consists of eighteen Dimensions and provides system designers the ability to quickly and reliably create multiple Cubes within an application. They also create a Dimension Library that enables reusability and Dimension calculation logic. This capability enables system designers to create optimal Analytic Models tailored to the specific business process exhibited without having to move data in and out of the application.

---

The Workflow Engine's job is to provide a common and seamless user experience no matter how an application designer chooses to implement the underlying Analytic Models (Cubes). This capability enables application designers to create organized applications because they are free from worrying about how to train users on the specific Analytic Models to access during the business process.

When a Cube is defined, it can be marked as a Top-Level Cube for Workflow. This setting tells the Workflow Engine the Cube is eligible for Workflow management. Each Cube can only participate in one Workflow Management Structure, which means if a Cube is referenced by another higher-level Cube, it cannot be set as a Top-Level Cube for Workflow.

After a Cube is tagged as a Top-Level Cube for Workflow, the Workflow Engine will recognize the Cube and allow a Workflow Management Structure to be created for the Cube based on the Suffix Values for varying Workflow by Scenario Type.

Building a Workflow Management Structure for a Cube starts with defining how many variations there will be in the processes used to feed the Cube (suffix for varying Workflow by Scenario Type). Workflow hierarchy variation is aligned with the ten Scenario Types that enable Cube extensibility. By default, no suffix values are created which means the Cube can only have one hierarchy in its Workflow Management Structure.

Setting up a Workflow Management Structure that enables multiple Workflow hierarchies is accomplished by assigning a defining set of suffix values reflecting the different business processes that will be used to populate the target Cube.

## Creating Workflow Suffix Groups

For example, a Cube has been created to collect actual reporting data, financial planning data, and some other miscellaneous data collections (Cube Name = FinancialReporting). The first step to defining a Workflow Management Structure for this Cube is to determine how data will be provided to the Cube and the business processes it was designed to analyze.

The table below details the three business processes for the Cube and the proposed suffix values that will identify the business process in the Workflow Management Structure.

Business Process	Data Source	Suffix
Actual:	System Interfaces	Sys
Plan Data:	Keyed/Excel Uploads by Entity	Ent
Other Data:	Unknown at this point	Oth

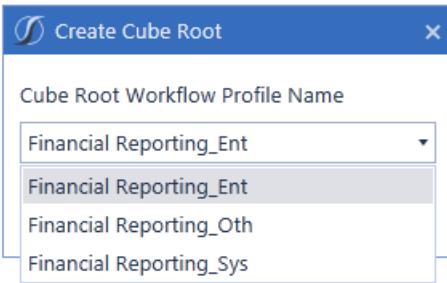
---

Next, the suffix values need to be assigned to the Scenario Type that will be used to capture data for each process. The assignment process will vary from application to application and will be dependent on the requirements of each business process.

Workflow	
Is Top Level Cube For Workflow	True
Suffix for varying Workflow by Scenario Type	
Actual	Sys
Administration	
Budget	Ent
Control	
Flash	Ent
Forecast	
FXModel	Oth
History	Oth
LongTerm	
Model	Oth
Operational	
Plan	Ent
Sustainability	
Target	
Tax	Sys
Variance	Oth

## Creating a Workflow Hierarchy Using a Workflow Suffix Group

Three Workflow hierarchies can be created for the FinancialReporting Cube. This is accomplished by opening the Workflow Profiles screen and clicking the Create a CubeRoot Workflow Profile button from the tool bar. The Workflow Engine will automatically present the three Workflow hierarchy options for the FinancialReporting Cube.



The Workflow Engine can now manage three different collection hierarchies for the FinancialReporting Cube. After a Workflow processing has begun and data is being actively collected for the Cube, the Workflow Management Structure (suffix definitions) cannot be changed.

Workflow can be thought of as an outline (Workflow Profile Hierarchy) of the business process that is being used to model and analyze. This section describes the Components of a Workflow Management Structure and the relationship between Workflow Profiles, Entity Members and Origin Members.

## Workflow Profiles

Workflow Profiles are the foundation of the data loading process, which is where Data Sources and Transformation Rules are assigned, and the anchors for the review and certifications. This section will go through more detail about all the configuration settings available. See Workflow in "Workflow" on page 445 for more details on how the Workflow function works.

**TIP:** It is best to lay out the organization first in Excel to view and review how the structure looks. Once confirmed, create the same structure in OneStream.

**TIP:** There are a variety of combinations for building out the Workflow Profiles. Below are a couple from some best practices.

### Fully Integrated

This is the complete flow from data loading to certification in the same structure.

### Shared Services

Data loading and data certification are integrated; however shared services are separated from the certification.

### Separation of Duties

Data load and certification each have their own structures.

---

### Navigation tips for this section

Each profile can have a separate configuration based on the Scenario. Out of the box, the standard configuration is set to (Default).

## Workflow Profile Toolbar and Right Click Options



### Create a Cube Root Workflow Profile

Use this to begin the hierarchy of a new Workflow Profile.



### Create Child Under Current Workflow Profile

There are three types of profiles that can be created:



### Review

Reviewer or Certification



### Base Input

Data Load



### Parent Input

Parent Adjustments



### Create Sibling of Current Workflow Profile

This has the same option as above.



### Delete Selected Workflow Profile and its Children

Use this to delete the selected Workflow Profile and any children it may have



### Rename Profile

Use this to rename a Workflow Profile or Workflow Profile Input Type



### Cancel All Changes Since Last Save

Use this to cancel any unsaved changes



### Save

Use this to save any changes made to selected Workflow Profile



### **Move Current Workflow Profile as Child of Another Profile**

Use this to move a selected child of a selected Workflow Profile to another Workflow Profile



### **Move Current Workflow Profile as Sibling of Another Profile**

Use this to move a selected sibling of a selected Workflow Profile to another Workflow Profile



### **Move Up**

Use this to move sibling profiles up in the hierarchy



### **Move Down**

Use this to move sibling profiles down in the hierarchy



### **Work with Profiles**

Use this icon when working in the Workflow Template screen to navigate to the Workflow Profile screen.



### **Work with Templates**

Use this icon when working in the Workflow Profile screen to navigate to the Workflow Template screen. See Workflow Templates for more details on this feature.



### **Update Input Children Using Template**

Select a specific Workflow Profile Template and assign it to a Workflow Profile's input child. See Workflow Templates for more details on this feature.



### **Unassign Selected Entity**

Use this to assign a selected Entity from a Workflow Profile



### **Show Items that Reference Selected Item**

Use this to see the other areas where the selected item is being used.



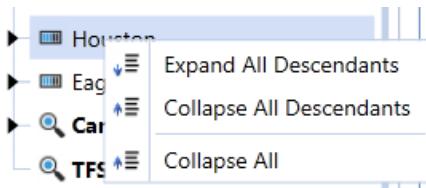
### **Navigate**

This icon appears in various fields and when clicked it navigates to a section that coincides with the Workflow Profile property. For example, if this icon is clicked in the Cube Name setting, the Cube screen will open allowing the user to make any changes needed before assigning a Cube to a Workflow Profile.



Use the Workflow search tool to filter down to the specific Workflow Profile or Workflow Input Type desired. The Cube Root defaults to the last one selected and displays the associated Workflow Profiles. In order to see another Workflow structure hierarchy, select a different Cube Root.

Right-click on any Workflow Profile name in order to expand or collapse all the selected Workflow Profile's descendants.



## Using Workflow Profiles

A Workflow Profile is the basic building block of a Workflow Management Structure. Another way to think about a Workflow Profile is a task list that should be performed by a group of users in relation to a group of Entities. Eight different types of Workflow Profiles are available for use within a Workflow hierarchy. Each Workflow Profile type and its role within a Workflow hierarchy is described below.

### Cube Root Profile Type

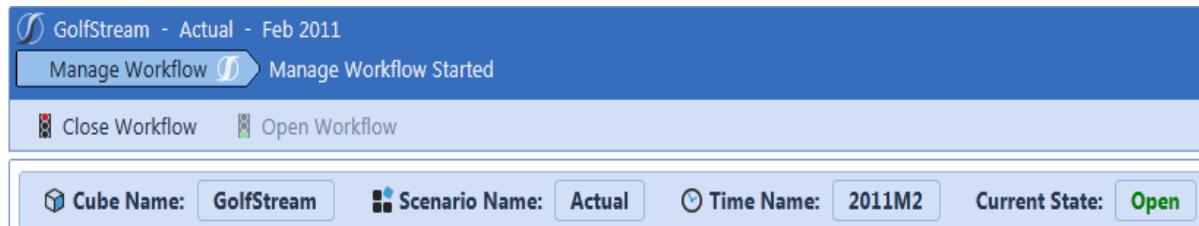
A Cube Root Profile should be thought of as the definition of a Workflow hierarchy for an entire Cube or a suffix group defined by the Cube. (See the preceding section on Cubes and Workflow). As a result, to create a new Cube Root Profile there must be a Cube defined in the Analytic Model marked as Is Top Level Cube For Workflow = True.

#### Default Profile Relationship

When a new Cube Root Profile is created, the Workflow Engine will automatically create a matching Default Input profile that is prefixed with the Cube Root Profile name (e.g. Corporate\_Default). A Default Input profile is required for each Workflow hierarchy and it cannot be deleted. The job of the Default Input is to establish the default relationship between the Workflow structure and the Entity Members within the Cube the Cube Root Profile was created to control.

## Controlling Workflow States

Cube Root Profiles have a predefined Workflow used to control the state of the Workflow Management Structure (Open / Closed). Cube Root Profiles are used as a mechanism to control the state of the entire Workflow hierarchy for a given Scenario and time because they represent the top element of a Workflow hierarchy.



### Open State

The Workflow is available for usage and locking is controlled at the individual Workflow Profile level. In addition, all Workflow hierarchy structure information is read from the current Workflow hierarchy as it reads the Workflow Profiles management screen. This also means the Workflow hierarchy is accessed from memory (cache) rather than being read from the database which provides very fast read performance.

### Closed State

The act of closing a Workflow hierarchy triggers the Workflow Engine to place a high-level lock on the Workflow. This means individual Workflow Profile lock status values do not matter, and the

Workflow level will display a to indicate a closed Workflow. In addition, the Workflow Engine will take a snapshot of the current Workflow hierarchy structure being managed from the Workflow Profiles management screen. It will store it in a historical audit table for the Scenario and time being closed. This also means the Workflow hierarchy is not accessed from memory (Cache) as would be the case with a Workflow in an open state. A closed Workflow must be read from the database rather than memory because it is considered a point in time snapshot stored in a historical table. This is a performance penalty noticed when reading the entire closed Workflow hierarchy for a Scenario and time. Workflow hierarchies should only be closed if major changes are being made to the Workflow hierarchy and the structure of a Cube and historical hierarchy relationships need to be preserved.

## Review Profile Type

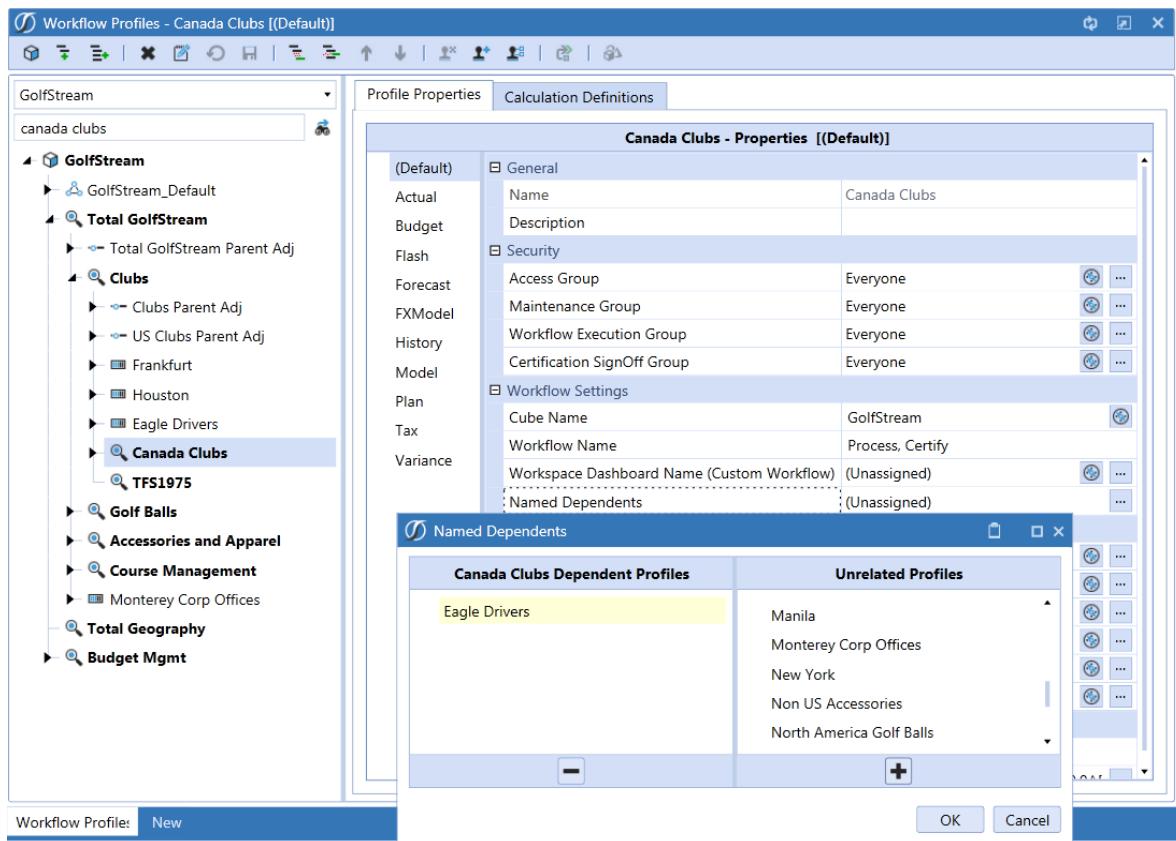
Review Profiles are best thought of as checkpoints in the Workflow hierarchy structure. This type of profile does not have a direct relationship to an Entity Member or Origin Member. However, Review Profiles can have Calculation Definitions assigned to them, so that Cube calculations can be ran in a controlled manner during the Workflow sign-off process.

## Named Dependents

Review Workflow Profiles have a unique ability to establish a dependency on the status of Input Parent Profiles that are not their direct descendant in the Workflow hierarchy structure. This concept is referred to as a Named Dependent relationship and was developed to accommodate situations where a single Input Parent Profile loads data for many legal Entities that have very different responsibility structures from a sign-off perspective. This situation is very common when an organization utilizes a Shared Services infrastructure strategy.

## Creating a Named Dependent

The screenshot below demonstrates an example of the Canada Clubs Review Profile establishing a Workflow dependency on the Eagle Drivers Input Parent Profile, even though the Eagle Drivers Input Parent is not a descendent of Canada Clubs.



---

## **Input Parent Profile Types**

Input Parent Profile types are special profile types because they are the point where a relationship is formed between the Workflow Management Structure and an Entity Member. All Input Parent Profiles types share the common purpose of organizing different types of data input used to feed an Entity. This is accomplished through the use and control of dependent profiles called Input Children.

All Input Parent Profiles must have at least one Input Child of each type (Import, Form and Adjustment). This requirement exists because of the relationship between Input Child Profiles and the Origin Dimension Members. Input Child Profiles can be thought of as a specialized extension of the Input Parent with added intelligence and control features particular to data updating.

### **Default Input**

Default Input Parent Profiles are special because they cannot be created directly. They are automatically created whenever a Cube Root Profile is created.

### **Unassigned Entities**

The primary purpose of the Default Input Profile type is to serve as the initial relationship between the Entities belonging to a Cube Root Profile and the Workflow hierarchy. Entities cannot be explicitly assigned to a Default Input Profile. Any Entity Member under the Cube with which the Default profile is associated and is not explicitly assigned to a Parent Input Profile or Base Input Profile, is implicitly assigned to the Default Profile.

### **Parent Input**

Parent Input Profiles are used to allow adjustments to Parent Entities in the Cube. Adjustments to a Parent Entity are only allowed via Forms or AdjInput Members of the Origin Dimension. Consequently, Import Child Profiles are not allowed to be used with a Parent Input Profile. The Workflow Engine will automatically create an Import Child for each Parent Input Profile, but the Import Child will be forced to be inactive (Profile Active = False).

### **Assigned Entities**

The primary purpose of the Parent Input Profile type is to establish a relationship between Parent Entities that require the ability to accept adjustments and the Workflow hierarchy. Parent Entities do not need to be explicitly assigned to a Parent Input Workflow Profile unless the Parent Entity requires the ability to be adjusted. Most Parent Entities exist as unassigned Entities and therefore are controlled by the Default Input Profile.

### **Base Input**

Base Input Profiles are used to control all methods of data entry for Base Entities in the Cube. This is the most common Workflow Profile type and can be thought of as the workhorse of data update management. Base Input Parents define the Entities that can be updated, the Cube being targeted, and all the Import Child types that will participate in the input scheme.

## Assigned Entities

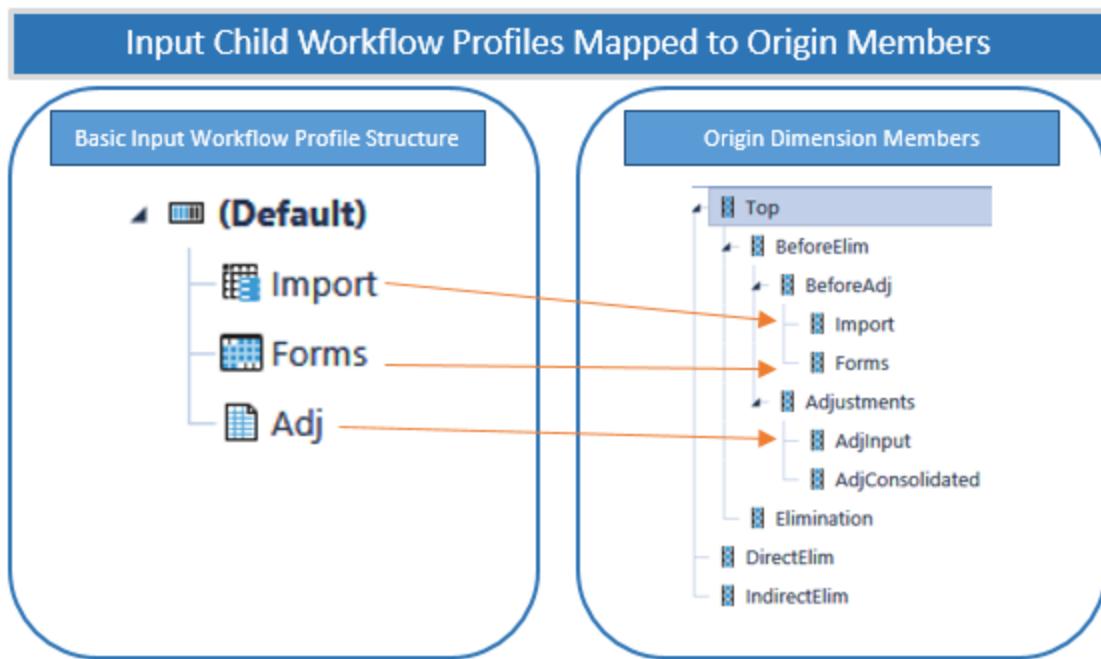
The primary purpose of the Parent Input Profile type is to establish a relationship between Base Entities that need to receive data from an external source and the Workflow hierarchy.

## Input Child Profile Types

Input Child Profile types will always be base Members of a Workflow hierarchy and can only be children of one of the three types of Input Profiles: Default, Parent or Base.

### Origin Dimension Relationship

Input Children are mapped directly to Origin Members in the Analytic Cube to create a control mechanism between the Workflow hierarchy and the Cube. This linkage enables the Workflow Engine to control the importing, form data entry, adjustment (journal) data entry and data locking processes for one or more Entities.



### Controlling Input Channels

The relationship between Origin Members and the Input Child Profiles that map to them enables the Workflow hierarchy to be configured in a way that allows/disallows certain input channels for the Entities assigned to the Input Parent Profile. By setting the Profile Active switch, an entire channel of input can be enabled/disabled.

---

For example, if a Forms Input Child has the Profile Active set to False, and there are no other active Input Child siblings of the type Form, data entry forms and Excel (SetCells Function or Cube Views) cannot be used to set data cell values for the Entities assigned to the Input Parent Profile of the Forms Input Child. The same technique can be used to enable/disable Import and Adjustment input types.

### **Import Child**

An Import Child defines and controls how data is imported into the Cube (See Data Loading for more details). Each Import Child is bound to the Data Source and Transformation Rule Profile which will define its Workflow behavior during the Import Workflow Step.

### **Import Child Origin Mapping**

The Workflow Engine enforces a relationship between Import Child Profile types and the Import Member of the Origin Dimension. This means, when loading data, the Origin Member will be forced to the value Import.

### **Forms Child**

A Form Child defines and controls how data is manually entered in the Cube. Each Form Child is bound to an Input Forms Profile that will define its Workflow behavior during the Input Forms Step.

### **Form Child Origin Mapping**

The Workflow Engine enforces a relationship between Form Child Profile types and the Forms Member of the Origin Dimension. This means when creating data entry forms, the Origin Member must be set to Forms or the data cell will appear as read only.

It is possible to use a data entry form to update the AdjInput Member of the Origin Dimension, but this requires the account being updated to have its Adjustment Type set to Data Entry rather than the default value of Journal.

### **Adjustment Child**

A Journal Child defines and controls how data is entered via journal into the Cube. Each Form Child is bound to a Journal Template Profile that will define its Workflow behavior during the Input Journals Step.

### **AdjInput Child Origin Mapping**

The Workflow Engine enforces a relationship between Adjustment Child Profile types and the AdjInput Member of the Origin Dimension. This means when creating journal entries, the Origin Member will be forced to the value AdjInput.

---

## Workflow Entity Relationship

The relationship between Workflow Profiles and Cube Entities creates a powerful asset that can act as leverage across many areas of the Workflow experience. By performing the one-time setup process of binding Entities to specific Workflow Profiles, the OneStream Workflow Engine can make this relationship information available to the application designer in many areas of the product. This means Workflow control structures and data entry mechanisms can be designed using abstract methodologies that refer to the current Workflow Unit value, which in turn can be resolved to a list of associated Entities.

### **Workflow Entity Relationship Member Filter Examples**

The primary way the Workflow Entity relationship is used is through analytic Member Filters.

#### **E#Root.WFProfileEntities**

When used in a Member Filter, this expression returns all Entities associated with the selected Workflow Unit.

#### **E#Root.WFCalcuationEntities**

When used in a Member Filter, this expression returns all Entities defined as part of the Calculation Definitions for the selected Workflow Unit.

#### **E#Root.WFConfirmationEntities**

When used in a Member Filter, this expression returns all Entities defined as part of the Calculation Definitions when the Confirmed Switch is set to True for the selected Workflow Unit.

## Multiple Input Workflow Profiles per Type

An application can have multiple input Workflow Profiles of the same type (Import, Forms, Journals) within each period. This is useful when multiple source systems are feeding the same Entities that need separate Data Sources in XF. It is also convenient when multiple form groups need to be completed by different groups of people. People only see and work on what they have access to because different access groups are created for each. The example below shows a budget where there is one Import Workflow Profile and eight different Form data entry channels that can be assigned to different groups of people. Each of these input channels can have multiple Forms.

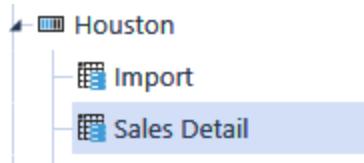


If there are multiple Import Workflow Profiles, it automatically handles clearing and merging data. For example, if there are two Import Workflow Profiles and import has already performed on one of them, when the second import is performed and the user clicks Load, all the target Entities are cleared. The two import data sets are merged, and a replace-style load is made to the financial model.

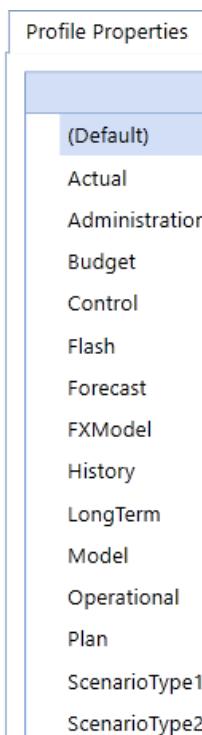
## Load YTD and MTD in the Same Workflow Profile

During the configuration of a Workflow Profile for a Scenario, the Workflow may require the submission of YTD data in one Origin and MTD/Periodic data in another Origin within the same Workflow Profile. If the Default View for the Scenario is configured as YTD, the Origin submitting MTD/Periodic data needs to be configured appropriately for it to behave as MTD/Periodic.

Select the Workflow Profile to configure, expand the Workflow Profile, and then select the Origin to be configured. In this example, Houston.Import is YTD and Houston.Sales Detail is MTD.



Next, choose the Scenario type where this behavior is needed. Select (Default) if this behavior is required for all Scenario Types.



Select Integration Settings to change the following properties:

Integration Settings	
Data Source Name	(Unassigned)
Transformation Profile Name	(Unassigned)
Import Dashboard Profile Name	(Unassigned)
Validate Dashboard Profile Name	(Unassigned)
Is Optional Data Load	False
Can Load Unrelated Entities	False
Flow Type No Data Zero View Override	Periodic
Balance Type No Data Zero View Override	YTD
Force Balance Accounts To YTD View	True

As a result, the Flow Type Accounts will be processed as Periodic rather than YTD upon data submission or when no data is loaded for Flow Accounts. The Balance Accounts are forced to be processed as YTD upon data submission.

## Cube Root Profile

The Cube Root Profile provides the hierarchy build and organizational structure of the Workflow Profiles for the different Cubes used within the application. All top-level Cubes can have one or more Workflow hierarchies defined depending on whether a suffix was added for a given Scenario Type in Cube settings.

## Workflow Profile Types

### Base Input

This is where most of the data updates take place and are broken out by the input channels whereby the Origin Dimension Members are filled out: Import (O#Import), Forms (O#Forms) and Journals (O#AdjInput).

#### Import

Import is typically used to load a GL data file or a OneStream configured Excel Template. The Import Origin can be configured by one of the following:

(Import, Validate, Load), (Import, Validate, Load, Certify), (Import, Validate, Process, & Certify), (Import, Validate, Process, Confirm, Certify) (Central Import), (Workspace), (Workspace, Certify), (Import Stage Only), (Import, Verify Stage Only), (Import, Verify, Certify Stage Only)

---

## **Form**

Form is used to load data either through a Form template or a pre-configured Excel XFSetsCell file. The Form Origin can be configured by one of the following:

(Form Input), (Form Input, Certify), (Form Input, Process, Certify), (Form Input, Process, Confirm, Certify), (Pre-Process, Form Input), (Pre-Process, Form Input, Certify), (Pre-Process, Form Input, Process, Certify), (Pre-Process, Form Input, Process, Confirm, Certify), (Central Form Input), (Workspace), (Workspace, Certify)

## **Journal**

Journal is used to load journal adjustment data through a journal template. The Journal Origin can be configured by one of the following:

(Journal Input), (Journal Input, Certify), (Journal Input, Process, & Certify), (Journal Input, Process, Confirm, Certify), (Central Journal Input), (Workspace), (Workspace, Certify)

## **Parent Adjustment**

If a top side adjustment is needed, do so with a Parent Input Workflow Profile either through a journal or form. Both will update the AdjInput Member in the Origin Dimension.

## **Review**

These Workflow Profiles do not take input, but are meant for reviewing, confirming and certifying results from the lower input Workflow Profiles. Since the Review Workflow Profiles cannot load data, the only tasks available for a Review Workflow Profile are (Process, Certify), (Process, Confirm, Certify)

See Workflow Tasks in "Using OnePlace Workflow" on page 945 of the Reference Guide for more details on each of these task types.

## **Profile Properties**

The first tab is the primary configuration tab and is available for all types of profiles. It is where security and other objects are attached.

**TIP:** This can be configured per Scenario Type.

## **General**

### **Name**

Name of profile.

---

## Description

Brief description of profile.

**NOTE:** If a description is added to a Workflow Profile in the Default Scenario, the description will display in the Workflow Profile POV dialog in OnePlace.

## Security

### Access Group

Controls the user or users that will have access to the Workflow Profile at run time to view results.

### Maintenance Group

Controls the user or users that will have access to maintain and administer the rule group.

### Workflow Execution Group

This group is configured for data loaders and allows users to execute Workflow.

### Certification SignOff Group

This group is configured for certifiers and allows users to sign off on the Workflow. This group can be used to separate duties between a data loader and certifier.

### Journal Process Group (Journals Only)

Access to this group allows users to process a journal.

### Journal Approval Group (Journals Only)

Access to this group allows users to approve a journal.

### Journal Post Group (Journals Only)

Access to this group allows users to post a journal.

You must be a Member of the Access and Workflow Execution Group to perform the Lock action for a Workflow Profile. If you have access to the Lock action, then you also have access to the Lock Descendants action for all Workflow Profiles below it in the hierarchy. This will happen even if you do not have Membership in the same security settings in the descendants' profiles.

**NOTE:** Click and begin typing the name of the Security Group in the blank field. As the first few letters are typed, the Groups are filtered making it easier to find and select the appropriate Group. Once the Group is selected, press CTRL and double-click to enter the correct name into the appropriate field.

---

### **Prevent Self-Post (Journals Only)**

Allows you to prevent users from posting their own journals when users have security rights to post, process, or approve them. The Post button is disabled in journals where the status is Created User, Submitted User, or Approved User. Additionally, the Quick Post button is replaced with Submit, Approve, and Post action buttons.

- If set to True, members of the Journal Post security group cannot post their own journals within this workflow profile. This configuration replaces the Quick Post action with the Submit, Approve, and Post actions if you are a member of all of the following groups: Journal Process Group, Journal Approval Group, and Journal Post Group. Administrators still have access to the Quick Post action.
- If set to True (includes Admins), administrators and members of the Journal Post security group cannot post their own journals in this workflow profile. This configuration replaces the Quick Post action with the Submit, Approve, and Post actions if you are an administrator or a member of all of the following groups: Journal Process Group, Journal Approval Group, and Journal Post Group.
- If set to False, administrators and members of the Journal Post Group are allowed to post their own journals. This is the default.

**TIP:** Administrators are members of the security group Administrators. This is defined at the environment level. They are also members of the security group set in the Application Security Role called AdministerApplication. This security group defines administrators for specific applications. Administrators are automatically granted rights to process, approve, and post journals without being members of Journal Security groups.

### **Prevent Self-Approval (Journals Only)**

Allows you to prevent users from approving their own journals when users have security rights to approve and process journals. The Approve button is disabled in journals where they are the Created User or Submitted User. Additionally, the Quick Post button is replaced with Submit, Approve, and Post action buttons.

Auto Approved and Auto Reversing journals are excluded from the self-approval restriction. A Journal created from an Auto Approved journal template type will be auto-approved when it is submitted. A journal that is automatically created when an Auto Reversed journal is posted will also be auto-approved.

- If set to True, members of the Journal Approval security group cannot approve their own journals within this workflow profile. This configuration replaces the Quick Post action with the Submit, Approve, and Post actions if you are a member of all of the following groups: Journal Process Group, Journal Approval Group, and Journal Post Group. Administrators still have access to the Quick Post action.
- If set to True (includes Admins), administrators and members of the Journal Approval security group cannot approve their own journals within this workflow profile. This configuration replaces the Quick Post action with the Submit, Approve, and Post actions if you are an administrator or a member of all of the following groups: Journal Process Group, Journal Approval Group, and Journal Post Group.
- If set to False, administrators and members of the Journal Approval Group can approve their own journals. This is the default.

**TIP:** Administrators are members of the security group Administrators. This is defined at the environment level. They are also members of the security group set in the Application Security Role called AdministerApplication. This security group defines administrators for specific applications. Administrators are automatically granted rights to process, approve, and post journals without being members of Journal Security groups.

#### **Require Journal Template (Journals Only)**

Restricts users from creating new free-form journals by disabling the Create Journals button. It also prevents user from loading journal files that do not contain a journal template, enforcing new journal creation from existing journal templates only. System and Application administrators can still create free-form journals. This property works in combination with the Journal Process Group property, which defines who can create journals in the workflow profile. Set the value to True to enable this feature.

## **Workflow Settings**

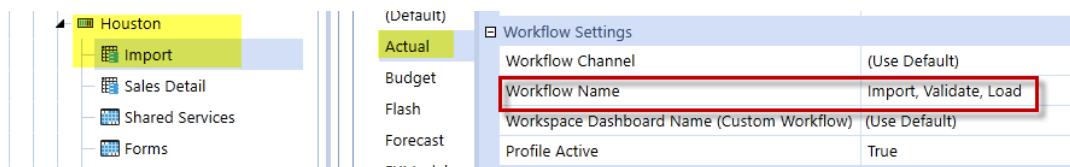
In this section the administrator can assign a Workflow Channel and a Workflow name. It is important to understand the tasks of each input type in order to properly assign a Workflow name.

### **Workflow Channel**

This option ties to the Workflow Channels found under the Application Tab/Workflow/Workflow Channels. This allows for an additional layer of security that defined through the Dimension Library. (e.g., Accounts, or UD1 – UD8). Click on the drop down to display all configured Workflow Channels. By default, this is set to Standard. This is an option on each load channel (such as Import, Forms, and Adjustments).

## Workflow Name

The Workflow name controls the tasks the users need to complete in the Workflow. There are a variety of combinations based on the type of Workflow being designed. These tasks can vary by Scenario and Input Type. For example, the Workflow Name Import, Validate, Load is set for the Houston Workflow Profile Import Input Type for the Actual Scenario.



When working in the Houston Workflow and loading data in the Actual Scenario, complete the following tasks:

A screenshot of the OnePlace interface showing a workflow named 'Houston.Import - Actual - Mar 2011'. The workflow has three main steps: 'Import', 'Validate', and 'Load'. The 'Import' step is highlighted with a red border. Below the workflow, a table displays data for 'Houston Heights' across different time periods. The table has columns for 'Amount', 'HoustonEntity', 'Time', and 'View'. The data rows are: 108,000,000.00 (Houston Heights, 2011M3, YTD), 4,758,681.97 (Houston Heights, 2011M3, YTD), 4,451,355.60 (Houston Heights, 2011M3, YTD), 14,799,505.11 (Houston Heights, 2011M3, YTD), 2,000.00 (Houston Heights, 2011M3, YTD), and 4,000.00 (Houston Heights, 2011M3, YTD). On the left, a sidebar shows the hierarchy: 'OnePlace' > 'Workflow' > 'Houston' > 'Actual' > '2011' > '2011 Periods' (Jan, Feb, Mar) > 'Houston.Import'.

See Workflow Tasks in OnePlace Workflow for more details on these tasks.

## Workspace Dashboard Name (Custom Workflow)

This property only works with a workflow name of Workspace or any selected containing workspace (for example, Workspace, Certify). Use this property to define any dashboard from the Application Dashboards page when an appropriate workflow name is chosen.

## Integration Settings (Import Only Except Where Indicated)

### Data Source Name

The way Fixed Width, Delimited, Data Management Export Sequences, or SQL-based based Data Sources are all managed here. These are originally configured under the Application Tab|Data Collection|Data Sources.

---

### **Transformation Profile Name**

The type of maps used for the Workflow Profile are managed here. These are configured under the Application Tab|Data Collection|Transformation Rules. This option will display the choices of profiles available.

### **Import Dashboard Profile Name**

The type of Dashboards wanted for the Workflow Profile Import phase are managed here.

### **Validate Dashboard Profile Name**

The type of Dashboards wanted for the Workflow Profile Validate phase are managed here.

### **Is Optional Data Load**

If a Workflow needs to load data to some periods, but not others, this option provides a quick way to complete the Workflow Channel if no data is loaded that month. When set to True, this Workflow Channel will receive an additional icon called Complete Workflow. You can then click on this and complete the process if no data is loaded to this channel.

### **Can Load Unrelated Entities (Import and Adjustment)**

Typically set to False unless there is historical data to be loaded.

- If set to True, a Workflow Profile can load data to entities that are not assigned to its Input Parent Workflow Profile.
- If set to False, a Workflow Profile can only load data to assigned entities. A journal can be saved with unassigned entities, but cannot be submitted, approved, or posted.

When creating a new blank journal, the Entity Member Filter property is set to the value E#Root.WfProfileEntities. This ensures that entity properties show only assigned entities in its list. This filter value can be changed, but when creating journals from a template, the Entity Member Filter property uses the value from the template instead.

**TIP:** The next three configurations are for configuring Workflow Profiles that may load as MTD. The entire Workflow for the year must be loaded consistently the same way for Zero No Data.

### **Flow Type No Data Zero View Override**

The Workflow will override the Scenario settings of load data from Stage and force Flow Accounts to the Zero No Data Selected. Settings are YTD and Periodic.

### **Balance Type No Data Zero View Override**

The Workflow will override the Scenario settings of load data from Stage and force Balance Accounts to Zero No Data Deselection are YTD and Periodic.

---

### **Force Balance Accounts to YTD View**

If set to True, Balance Accounts are forced to the YTD View for loading no matter what View Member is assigned to the account in the data load file, if set to False, Balance Accounts are loaded with the View Member assigned to the account in the data load file.

### **Cache Page Size**

Integer must be greater than 0. Setting will default to 20000 records upon save if an invalid value is entered.

### **Cache Pages In-Memory Limit**

Integer must be greater than 0. Setting will default to 200 upon save if an invalid value is entered.

### **Cache Page Rule Breakout Interval**

Integer must be greater than 0. Setting will default to 0 upon save if an invalid value is entered. The value entered reflects the count of Transformation Rules which define a point to pause processing to check if the current Cache Page's data records are fully mapped. If the current Cache Page is determined to be fully mapped, the remaining Transformation rules will not be processed.

## **Form Settings (Forms Only)**

### **Input Forms Profile Name**

The form templates displayed in the forms channel for this Workflow Profile are managed here. These are configured under the Application Tab|Data Collection|Form Templates.

## **Journal Settings (Journals Only)**

### **Journal Template Profile Name**

The journal templates displayed in the adjustments channel for this Workflow Profile are managed here. These are configured under the Application Tab|Data Collection|Journal Templates.

## **Data Quality Settings**

### **Cube View Profile Name**

The Cube Views displayed in the analysis pane are managed here. This section is displayed when clicking on the period level during the Process, Confirm, and Certify steps. These are configured under the Application Tab|Presentation|Cube Views.

### **Process Cube Dashboard Profile Name**

The Dashboard's rules that run during the Process step are managed here.

---

### **Confirmation Profile Name**

The Confirmation Rules that run during the Confirm step are managed here. These are configured under the Application tab|Workflow|Confirmation Rules.

### **Confirmation Dashboard Profile Name**

The Dashboards displayed during the Confirmation step are managed here.

### **Certification Profile Name**

The certification questions that prompt the user during the Certify step are managed here. These are configured under the Application Tab/Workflow/Certification Questions.

### **Certification Dashboard Profile Name**

The Dashboards displayed in the analysis pane are managed here. This section is displayed when clicking on the period level on the Certify step. These are configured under the Application Tab/Presentation/Dashboards.

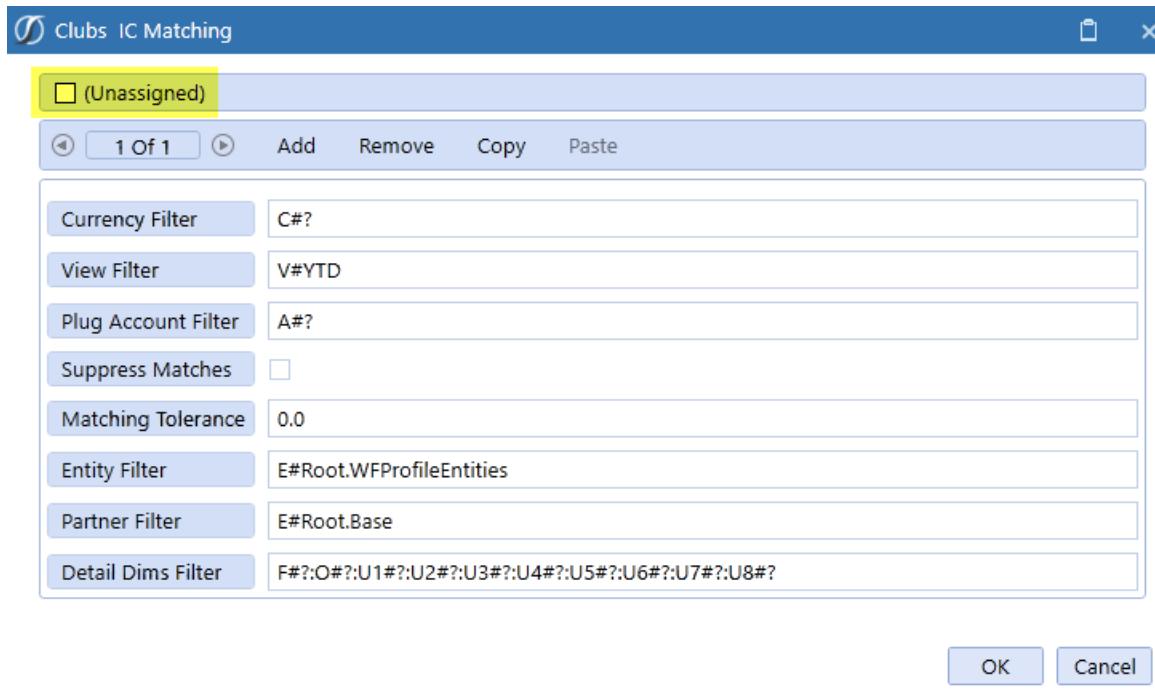
## **Intercompany Matching Settings**

### **Matching Enabled**

When set to True, the Matching Parameters section needs to be configured. This can be configured at all channels of the Workflow Profile. Settings are True or False.

### **Matching Parameters**

Click on the ellipsis button in this field to configure the matching settings.



**TIP:** Make sure the (Unassigned) box is unchecked, otherwise the fields will not be available to update.

### **Currency Filter**

Enter the reporting currency.

### **View Filter**

Enter how the Intercompany data should display in the Workflow (such as V#YTD or V#Periodic).

### **Plug Account Filter**

Enter the Plug Account for this Intercompany match.

### **Suppress Matches**

Check this to apply suppression.

### **Matching Tolerance**

If the matching tolerance must be on the penny, leave this at 0.0, otherwise add a tolerance threshold for the offset amounts.

### **Entity Filter**

The Workflow Entities associated with the Intercompany matching. E#Root.WFProfileEntities automatically points to the Entities assigned to the Workflow.

### **Partner Filter**

Enter a Member Filter specifying the Partner Entities.

---

### **Detail Dims Filter**

Enter the Account-Level Dimension Members (Flow, Origin, UD1...UD8) to ensure users are seeing the correct data in the Workflow.

## **Entity Assignment**

The third tab is only available when clicking on the Cube Root Workflow Profile. This is where the actual Entity gets tied to the Workflow Profile. There are two sections to this tab:

### **Entity Assignment [Cube Name]**

This section repaints the Workflow as it was created in the middle pane. This is where an administrator can click on a Workflow Profile, and then use the Unassigned Entities tab to search for Entities.

### **Unassigned Entities**

This is the search window to find Entities to attach to a profile. This only becomes enabled for the data loading profiles. The search engine uses that contains technology to find Entities. In other words, type the whole word or part of the word and it will search through the complete list looking for that combination of characters. More than one profile may be chosen. Lastly, once an Entity has been attached, it will no longer appear in the search window.



- Click this after the search criteria has been typed



- Click this to attach the item to the profile



- Click this to turn off the search criteria

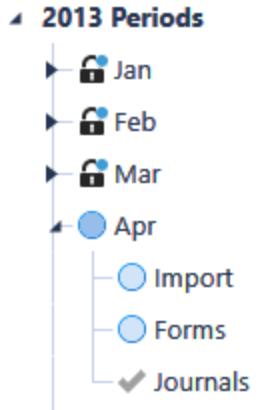
## **Workflow Profile Grid View**

The Workflow Profile Grid View allows an administrator to make several changes to numerous Workflow Profiles at once. Select the Grid View icon and choose data to view. There is also a drag and drop option where the user can select a column label and drag it to the top in order to specifically group the data. This is only available when the Cube Root Profile is selected.

Grid View				
Type: Base Input    Cube: GolfStream    Scenario Type: (Default)    Show Derived Values: No				
Grouped by: Maintenance Group				
Name	Access Group	Maintenance Group	Workflow	Process Cube/Confirmation Cube View Profile Name
<b>Administrators</b>				
Manchester Bags	WF_ManchesterBags_Access	Administrators	Process, Confirm, Certify	Process_All
<b>Everyone</b>				
US Accessories	Everyone	Everyone	Process, Certify	Process_All
Frankfurt Drivers	Everyone	Everyone	Process, Certify	Process_All
Canada	Everyone	Everyone	Process, Certify	Process_All

## Central Input

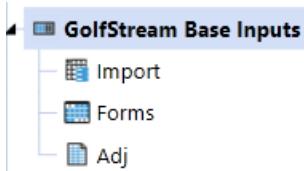
For corporate configurations there is an option called Central Input. This can be used for situations where corporate does the final adjustment after the data has been loaded by the sites. The Workflow Channel can configure with Central Input which will then display a grey check mark for that Workflow Channel. This allows corporate to make updates and adjustments at the top level because the Workflow owns the Entity. All activity is tracked in the audit history.



## Workflow Templates

Workflow Templates are useful when building a series of Base Input Workflow Profiles with similar settings. Design a template as generic or customized as desired and then apply the template to the new Base Input Workflow. On the Workflow Templates screen, there is a (Default) Template

that can be used or click  to create a new one. After the template is created, it will look similar to a Base Input Workflow Profile and include the default input types.



Customize the template to fit the needs of the Workflow design:

- Rename the inputs (for example, rename Adj to Journals)
- Add additional input types
- Disable input types by Scenario (for example, disable Journals for the Budget Scenario if that input type is not used)
- Configure Intercompany Parameters
- Assign common Cube View or Dashboard Profiles

The goal of the template is to make as many common changes and updates as possible which will save time and clicks during the actual Workflow build. After a template is completed, click to navigate back to the Workflow Profile page.

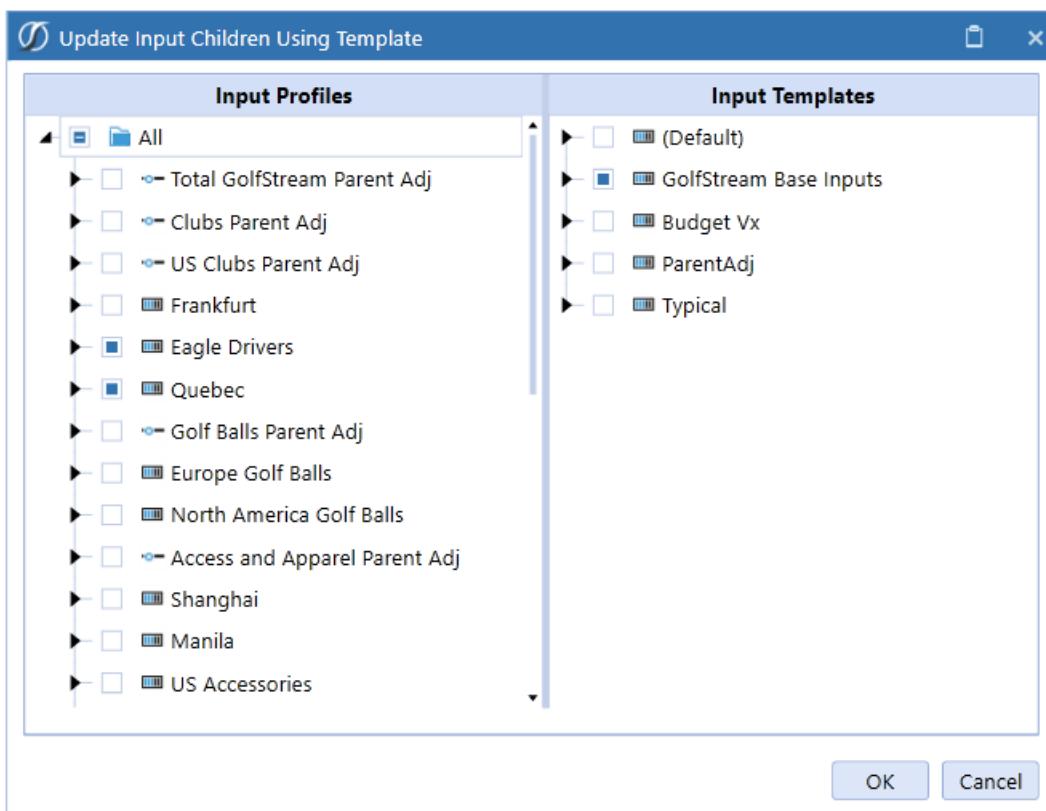
Create a Base Input Workflow and apply the template.

Name	<input type="text"/>
Profile Type	Base Input
Default Cube Name	GolfStream
Template Name	GolfStream Base Inputs

OK Cancel

All the template settings now apply to this Base Input.

**NOTE:** After a template is applied to a Base Input Workflow Profile, any changes made to the existing input types (Import, Forms, Journals) cannot be applied to the Workflow Profile. If a new Input Type is added to the template, this can be applied to an existing Workflow Profile. From the Workflow Profile screen, select  . Select the template with new input types and the Workflow Profiles to which the changes will apply.



## Using Calculation Definitions

All Workflow Profile types, except Cube Root, can create a set of Calculation Definitions. A Calculation Definition can be thought of as a macro or set of instructions performed whenever the Process Cube Step is executed.

Houston - Calculation Definitions										Dependent Entities	
Entity	Parent	Cons	Calc Type	Scenario Type Filter	Confirmed	Order	Filter Value			Dependent Entities	
Houston	(Unassigned)	Local	Consolidate	(All)	<input type="checkbox"/>	20				Houston Heights	
(Assigned Entities)	(Unassigned)	Local	No Calculate	(All)	<input checked="" type="checkbox"/>	30				South Houston	

---

Calculation Definitions are an incredibly valuable tool to the application designer because they take the guess work out of what needs to be calculated and when. During the Workflow hierarchy design process, the Calculation Definitions can be used to run combinations of calculations, translations, and consolidations at Workflow completion points.

When defining Calculation Definitions for a Workflow Profile, the Workflow Entity Relationship can be leveraged. This means predefined variables can be used to run calculations for Entities assigned directly to a Workflow Profile (Input Parent types) or related to the Workflow Profile through its dependency chain.

## Calculation Definition Entity Placeholder Variables

### Review Profile Type

#### Dependent Entities

This defines a calculation for all Entities that assigned to the dependent Workflow Profiles of the Review Profile. This list also includes all Entities assigned to any Named Dependent Profiles.

#### Named Dependent Filter

This allows a filter value to be used to specify the specific Entities that are valid dependents of the Review Profile. This is required because in most cases Named Dependents are shared services and tend to have Entity relationships that span over multiple Review Profiles. This includes all Entities assigned to a Named Dependent that may return Entities not relevant to a Review Profile.

### Input Parent and Input Child Profile Type

#### Assigned Entities

This defines a calculation for all Entities directly assigned to the Workflow Profile.

#### Loaded Entities

This defines a calculation for all Entities imported by the Import Child Workflow Profiles and are dependents of the Input Parent.

#### Journal Input Entities

This defines a calculation for all Entities adjusted with journal entries by the Adjustment Child Workflow Profiles and are dependents of the Input Parent.

#### User Cell Input Entities

This defines a calculation for all Entities affected by data entry performed by the user executing the Workflow. This variable will return a different Entity list for each user. This is typically used in situations where a Workflow is setup without specifically assigned Entities but must update specific data cells across Entities owned by other Input Parent Profiles. This is referred to as a Central Input Workflow design and is usually used by corporate offices to control values in select Entities and accounts.

## **Input Child Profile Type**

Input Child Profiles are considered extensions of their Parent Input Profile and as a result, they do not require an explicit Calculation Definition. If an Import Child does not have an explicit Calculation Definition defined, it will default to those defined by its Input Parent.

### **Confirmed Switch Value**

Each Calculation Definition record has a Confirmed switch associated with it. This switch determines whether the Entities defined by a Calculation Definition should be subjected to the Confirmation Workflow Step. It also gives the application designer control over which Entities are subject to the Confirmation Rule validation process.

### **Filter Value**

Assign a Data Management Sequence to Calculation Definitions by setting the name of the Sequence under the Filter Value and setting the Calc Type to No Calculate. Next, set up a DataQualityEventHandler Extensibility Business Rule to read the Sequence name assigned to the filter and in turn, run the Data Management sequence during the Process Cube task in the Workflow.

Clubs - Calculation Definitions								
Entity	Parent	Cons	Calc Type	Scenario Type Filter	Confirmed	Order	Filter Value	
Clubs	Total GolfStream	Local	Consolidate	(All)	<input type="checkbox"/>	0		
Clubs	Total GolfStream	Local	No Calculate	(All)	<input type="checkbox"/>	10	Nominalize Scenarios	

## **Calculation Definitions**

The second tab is where the administrator assigns the Calculation Definitions for the Workflow Profile. This will determine the type of calculation/consolidation that occurs when a user selects Process Cube during the Workflow. Multiple Entities can be entered and ordered accordingly. See Using Calculation Definitions in "Workflow" on page 445 in the Design Guide for more details.

Houston - Calculation Definitions								
Entity	Parent	Cons	Calc Type	Scenario Type Filter	Confirmed	Order	Filter Value	
Houston	(Unassigned)	Local	Consolidate	(All)	<input type="checkbox"/>	20		
(Assigned Entities)	(Unassigned)	Local	No Calculate	(All)	<input checked="" type="checkbox"/>	30		

**Dependent Entities**  
Houston Heights  
South Houston

Auto-assigning Entities can be done through (Assigned Entities) or (Load Entities). The Confirmed check box controls which Entities are tested by Confirmation Rules.

---

**TIP:** By right clicking on any line item, a user can insert or delete a row, save, or export data.

## Data Units

A data unit is used to load, clear, calculate, store, and lock data in the multi-dimensional engine. With workflow channels, OneStream provides the following data units.

### Level 1: Cube Data Unit

This is the largest unit of work and is commonly referred to as the entity, scenario and time data unit. This aligns with financial analytic system tasks to clear, load, calculate, and lock entity, scenario and time combinations.

Members of the cube data unit are:

- Cube
- Entity
- Parent
- Consolidation
- Scenario
- Time

Cube data unit analytic work items include:

- Clear data
- Load data
- Copy data
- Calculate
- Translate
- Consolidate

### Level 2: Workflow Data Unit

The workflow data unit builds on the cube data unit by including the account dimension. This decreases the unit of work by increasing the data unit's granularity. This indicates a workflow data unit is a sub-set of the cube data unit, allowing fine-grained control over analytic work items such as clearing, loading, and locking data cells.

---

The workflow data unit is the default level used by the workflow engine to control, load, clear, and lock data. Workflow level data loads from the staging data mart to the cube, and is cleared and locked at a granularity level that includes the account dimension by default.

For example, if two import workflow profiles are not siblings of the same input parent, but load to the same entity, scenario and time dimensions, the data loads and clears at the account level. However, if these two workflow profiles load the same accounts, the last workflow profile to load is used. If these workflow profiles load to different accounts, then data loads for both workflow profiles.

Members of the workflow data unit are:

- Cube
- Entity
- Parent
- Time
- Consolidation
- Scenario
- Account

Workflow data unit analytic work items are:

- Clear data
- Load data
- Lock data

## Level 3: Workflow Channel Data Unit

The workflow channel data unit builds on workflow the data unit by including a single user-defined dimension. The user-defined dimension decreases the unit of work by increasing the data unit's granularity. This means a workflow channel data unit is a subset of the workflow data unit, allowing fine-grained control over analytic work items such as clearing, loading, and locking data cells.

The user-defined dimension that extends the data unit is specified at the application level from the **Application Properties** screen. You can only use one user-defined dimension per application, so carefully consider which user-defined to select in relation to the application's dimension dimension.

---

For example, cost center and version user-defined dimensions are commonly used in a workflow channel data unit. These user-defined dimension are frequently included in a workflow channel data unit because they represent data slices that align with data collection and locking requirements of the Budget and Forecast business processes.

## Workflow Channel Unit Budget Example

A typical cost center budget collection process has many users submitting data to a single cube data unit. In addition, all users are submitting data for a cost center to the same workflow data unit. This can inevitably cause data contention within the workflow data unit. Therefore, assigning the user-defined dimension containing cost centers as the user-defined dimension type for Workflow Channels makes each entity/cost center act as a granular and autonomous cell collection in a legal entity. This allows individual cost centers to load, clear, and lock data with no impact to other cost centers.

Workflow channel data unit members are:

- Cube
- Entity
- Parent
- Time
- Consolidation
- Scenario
- Account
- User Defined (x)

Workflow channel data unit analytic work items are:

- Clear data
- Load data
- Lock data

## Data Loading

This section describes how the workflow engine loads data for each relationship between the workflow and data units. The OneStream workflow engine controls data loading from the staging data mart to the analytic model. The workflow engine includes intelligence about what data to load and how to load for each workflow unit. The loaded analytic cube gets this from the binding relationship between its input parent workflow profiles and base entities.

---

In addition, the workflow engine uses the origin dimension's import member exclusively when loading data. This predefined relationship provides a built-in level of data protection between imported data, manual data entry, and journal adjustments. The workflow engine manages how data is placed into the origin dimension's Import, Forms, and AdjInput members. The workflow engine also forces imported data to use the local member of the consolidation dimension.

The workflow engine always starts with a workflow data unit to control clearing, loading, and locking data for its entities. A workflow channel data unit is used if workflow channels are active in the workflow unit's analytic model relationship.

## Data Load Execution Steps (Clear and Replace)

When a workflow unit's data load process executes, the engine does the following:

1. Checks workflow state
  - Implicitly locked (parent workflow certified)
  - Explicitly locked
2. Checks workflow profile data load switches
  - Can load unrelated entities (True / False)
  - Flow type no data zero view override (YTD / Periodic)
  - Balance type no data zero view override (YTD / Periodic)
  - Force balance accounts to YTD view (True / False)

See *Integration Settings* under *Workflow Profiles* in *Workflow* for descriptions of these settings.
3. Analyze prior data loads
  - Evaluate previously loaded data units to list data units to clear during the load.
4. Execute clear data

- 
- User-defined workflow channel configuration not used
  - Clear Workflow data units loaded by the workflow unit. A workflow data unit considers accounts and cube data unit standard members, so data clears at an account level by default.

#### User-Defined Workflow Channel Configuration Execution Steps

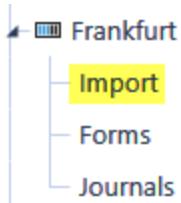
Clear all workflow channel data units loaded by the workflow unit. User-defined members and workflow data unit members are standard members of a workflow channel data unit, so data clears at a user-defined member entity, scenario, time, and account level by default.

1. Execute load data
  - Data loads using parallel processing by entity. Multiple entities process at the same time.

## Workflow Profile Data Loading Behaviors

This section describes three specific data loading behavior patterns. These behaviors range from the basic data load process where one workflow profile loads one data unit, to more than one workflow profile loads one data unit.

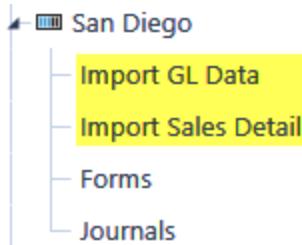
### Behavior 1: Single Workflow Profile - Loading One or More Entities on a Mutually Exclusive Basis:



This workflow profile configuration has only one import child profile under the parent (Frankfurt). The workflow engine follows basic clear and replace data loading steps described in "Data Load Execution Steps (Clear and Replace)" on the previous page.

---

#### **Behavior 2: Multiple Import Child Workflow Profiles**



This workflow profile configuration has more than one import child profile under the parent (San Diego). The workflow engine must perform extra steps to determine how to load data in the child profiles.

In this case the two import child profiles may try to load the same cube or workflow data unit because they have the same input parent workflow profile and are trying to load the same entities. When the import GL data or import sales detail workflow profiles execute the data load step, the following process determines how to correctly load data from both pofiles to the cube.

**Multiple Import Child Data Load Evaluation Steps:**

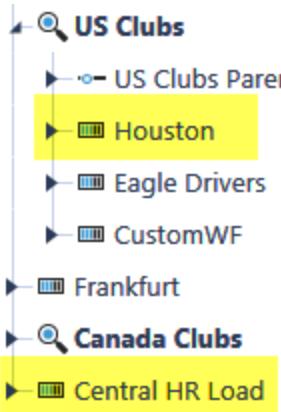
1. Check for overlapped data units between import child siblings (import GL data or import sales detail).
2. Determine the siblings have overlapped data units.

If yes, clear all previously loaded data units for both import child siblings, then reload both using an accumulate load method in the order they appear in the workflow hierarchy. If the two siblings are loading to the same cells, the values are be added together and placed in the cell.

If no, use the basic clear and replace data loading steps described in "Data Load Execution Steps (Clear and Replace)" on page 479.

---

### Behavior 3: Multiple Import Parent Workflow Profiles Loading One or More Common Entities



This workflow profile configuration has a central input parent profile that may load data assigned to another workflow profile. The Central HR Load workflow profile must have the Can Load Unrelated Entities set to **True**, so the workflow engine will let it try to write data for unassigned entities.

In this situation, when either Central HR Load or Houston executes a data load, the basic clear and replace data loading steps described in the previous section are used. However, Central HR Load does not control any entities so it checks and abides by the workflow and locking status of the workflow profiles that own the entities. For example, the workflow engine disallow updates if Houston is certified or locked and Central HR Load tries to load an entity owned by Houston.

#### Managing Sibling Imports

Certain application designs may require a workflow parent to have multiple sibling import channels. These designs typically use parallel processing techniques to load multiple non-overlapping sibling import children. The Load Overlapped Siblings setting on the parent boosts parallel processing performance in these workflow designs by eliminating overlapping checks between sibling channels. It only happens when the sibling channels' data sources do not contain overlapping data unit data records. This switch lets applications optimize data partitioning with parallel processing using the least workflow profiles.

#### Load Overlapped Siblings

- **True:** Default behavior, sibling channels check for overlapping data units.
- **False:** Do not check sibling channels for overlapping data units. If an overlapping condition occurs, the last processed channel overwrites the prior.

The screenshot shows the 'Workflow Profiles - AmericaBaseInput [(Default)]' window. On the left is a tree view of the CorpCube structure, with 'CorpCube\_Default' expanded to show 'America', 'Europe', 'Asia', and 'CorpParentAdjustments'. 'America' is further expanded to show 'AmericaBaseInput' which contains 'Import', 'Import2', 'Forms', and 'Adj'. 'CorpCube' is selected in the search bar. The main area has tabs for 'Profile Properties', 'Calculation Definitions', and 'Entity Assignment'. The 'Profile Properties' tab is active, showing the 'General' section for 'AmericaBaseInput'. The 'Name' field is set to 'AmericaBaseInput'. Under the 'Security' section, 'Access Group' is set to 'Everyone'. In the 'Workflow Settings' section, 'Cube Name' is 'AmericaCube', 'Workflow Name' is 'Process, Confirm, Certify', and 'Workspace Dashboard Name (Custom Workflow)' is 'CorpDashboard'. The 'Base Input Settings' section is highlighted with a red box, containing the 'Load Overlapped Siblings' checkbox which is checked ('True'). Other sections like 'Data Quality Settings' and 'Cube View Profile Name' are also visible.

## Data Locking

OneStream uses a locking strategy different than other analytic systems. All data control tasks are delegated to the Workflow Engine including the Entity data locking control because of the integrated Workflow Controller.

The Workflow Engine creates a bidirectional link between the Workflow Engine, the Staging Engine, and the Analytic Engine. This two-way link creates a much stronger control structure compared to systems with separate Workflow control modules that only interact with an Analytic Model in a unidirectional control structure.

This is an important control feature because if a user of the system attempts to update a data cell directly after all Workflow processing is completed, the Analytic Engine must check with the Workflow Engine to determine if the cell can be updated. In a unidirectional control structure, the data cell can be unlocked and updated regardless of the Workflow control state creating a break in the process audit chain. This situation cannot exist because every input data cell is associated with a Workflow Unit. Any attempt to update a data cell directly (Data Entry Form or Excel, etc.) triggers the Workflow Engine to validate the data cell's Workflow state by resolving its Workflow status through the Entity assignment relationship mechanism.

Locking data in means the data is Locked for Input. When data is locked (Explicitly or Implicitly) the Workflow Engine will not allow any form of data input to affect the Entities assigned to the Workflow Profile of the locked Workflow Unit.

---

## Lock Types

### Explicit Locks

An Explicit Entity Data Lock is created when a Workflow Unit is locked therefore locking its assigned Entity(s) for the Scenario and Time associated with the Workflow Unit.

### Implicit Locks

An Implicit Entity Data Lock is created when a Workflow Unit's Parent Workflow has been certified. Implicit locks are created to ensure after a higher-level Workflow Unit is certified, the underlying Entity data cannot be changed. Implicit locks can be cleared by un-certifying the Parent Workflow Unit.

### Workflow Only Locks

If a Workflow Unit is locked and the Workflow Profile does not have assigned Entities, all Workflow processing is blocked, but there are no Entity locks placed.

### Locking Granularity

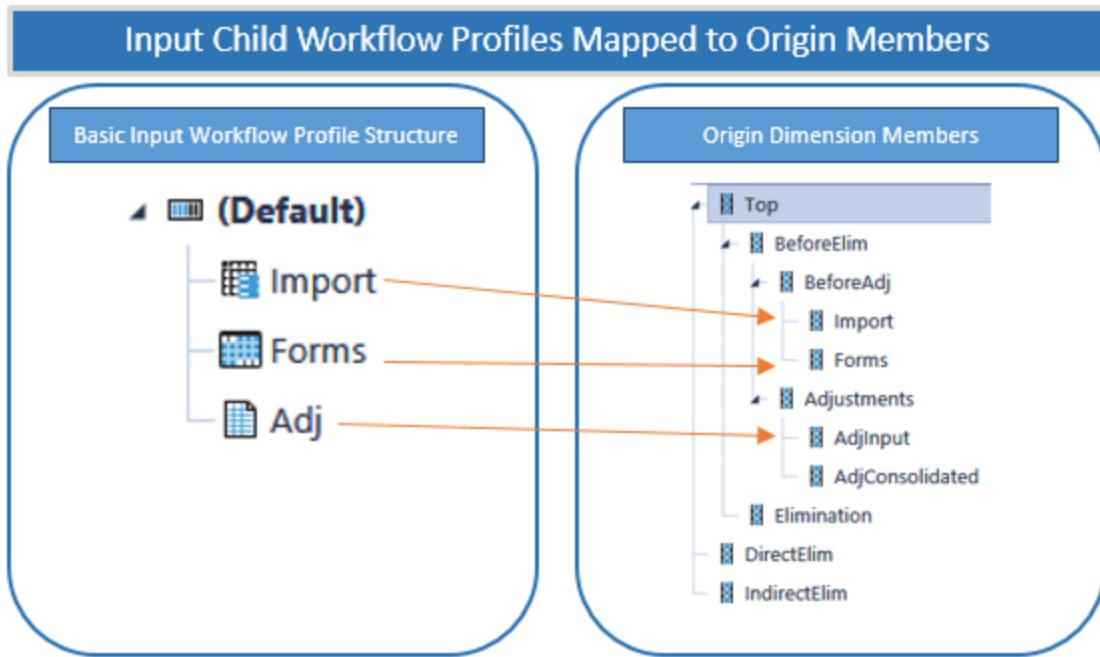
Data locks can be placed at different levels of granularity within the Analytic Model.

#### **Input Parent Workflow Profiles (Level One Data Unit Lock)**

The highest level of locking occurs when an Input Parent Workflow Profile is locked. This is a Level One Data Unit Lock because the Workflow Engine will force all data cells for the Entities assigned to the Input Parent to be locked. This is accomplished by locking the Input Parent Workflow Profile as well as the Input Child Workflow Profiles regardless of Origin Member binding or Workflow Channel assignment.

#### **Input Child Workflow Profile (Origin Lock)**

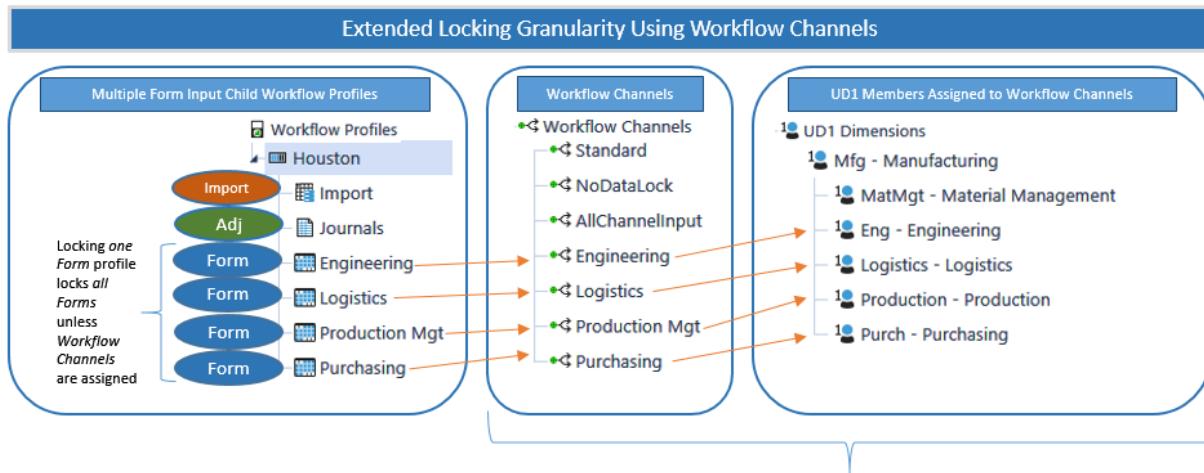
OneStream utilizes a predefined relationship between the three types of Input Child Workflow Profiles and a static Dimension called the Origin Dimension to control and lock the three basic channels of input for an Entity. This relationship provides the ability to control and lock data entry processes and associated data cells by binding the Workflow Profile Input Child Types to Origin Dimension Members.



## **Input Child Workflow Profile (Workflow Channel Lock)**

Through the use of Workflow Channels assigned to Input Child Workflow Profiles and associated with Accounts or User Defined Members, it is possible to provide very detailed locking granularity within a given Origin Member.

The diagram below demonstrates how the Forms Origin Member has been divided into multiple Workflow Channels enabling each Form Input Child Workflow Profile and the UD1 data cells bound to the same Workflow Channel to be locked independently. Workflow Channels can be used with Import Input Children as well Adjustment Input Children.



Assigning a different Workflow Channel to each Form Input Child and then to each corresponding UD1 Member increases the locking granularity to the level of the assigned Workflow Channel.

Example:

Locking the *Purchasing* Form Input Child will lock the intersections for the Entities assigned to Houston, where the Origin Member is *Form* and the UD1Member is *Purch*.

## Workflow Stage Import Methods

Workflow Stage Import has three primary methods of integrating data from the Stage Engine to meet the reporting requirement for each OneStream application model.

- **Standard:** Highly durable and auditable, stored details that target Finance Engine Cubes.
- **Direct:** In-memory, performance-focused, no storage of record details, that target Finance Engine Cubes.
- **Blend:** In-memory, high-performance import designed to blend the multi-dimensional structure with transactional data. No storage of record details or targeting external relational tables needed.

## Standard Import Methods

A Consolidation Model requires that data be durable and auditable, as OneStream is functioning as the “book-of-record” for financial reporting. In this regard, the Stage Engine’s Standard Type Workflow is ideal in that each Import performed has its source and target records stored in Stage tables. This allows every current and historical period loaded to be audited and analyzed at the source and target record level used for loading data to the Finance Engine. The stored tables also allow detailed analysis from the Finance to the Stage using the Drill-Down feature.

---

## Direct Import Methods

Certain Planning, Forecasting or Operational models may not require detailed audit and historical durability of data, with the resulting database overhead, that is required of the Consolidation Model. The Stage Engine's Direct Load Type Workflow is specifically designed to support the needs of data that is more operational in nature. The Direct Load Type's in-memory processing, and lack of storing Source and Target record detail, enhances its performance in processing compared to the Standard Type Workflow. The performance benefits of not storing source and target records is also what makes the Direct Load Type inappropriate for Consolidations, where detailed audit, history and drill-back is required.

## Blend Import Methods

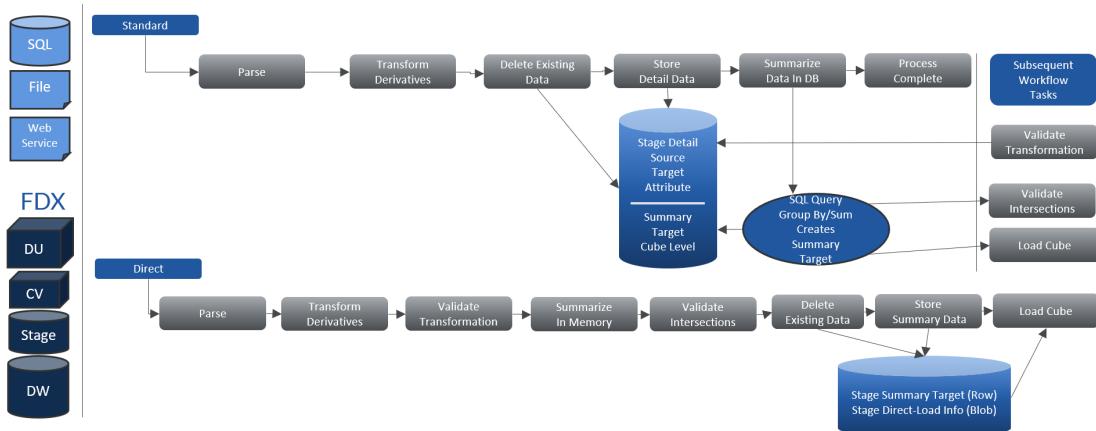
The Blend Type Workflow is an in-memory process and the integration method used by the BI Blend Engine, which writes to external relational database tables and not to a Finance Engine Cube.

The BI Blend Engine is a key element of Analytic Blend models as it is a read-only aggregate storage model. The purpose is to support reporting on large volumes of data that is not appropriate to store in a traditional Cube, such as transaction or operational data.

The Blend Type Import rationalizes the source data into a structure that is uniform and standardized for reporting by leveraging Cube Dimensions, deriving the metadata and aggregation points for the resulting BI Blend relational tables. This enables the transaction content stored in the relational tables to be aligned with the Finance Engine Cube data through common metadata and aggregation points for Analytic Blend Reporting. Refer to the *BI Blend Design and Reference Guide* for more detail.

## Stored versus In-Memory Workflow Imports

The primary difference between the Standard and in-memory processing Import methods is how the source and target record details are stored in the Stage tables.



## Guidelines on Volumes and Limits

Workflow Import performance is optimized by designing a Workflow structure that considers the data volumes of the total number of source records and the resulting total number of summarized target records. The transformed source records are summarized in the **StageSummaryTargetData** table. The volumes can be managed by structuring the data source loads across Workflows, gaining advantages in parallel processing.

Limit	Description
Row Limit Per Workflow	24 million summarized records
Best-Practice Recommendation	1 million summarized records

Example	Description
Single Workflow Results	24 million summarized records
Best-Practice Solution	Parse the file to be used across 24 partitioned Workflows

The benefits of efficient Workflow structure using partitioning when working with large data sources are:

- 
- Performance gains through parallel processing
  - Shorter processing times
  - Faster mapping and error correction
  - More transparent data validation

## Standard Workflow Record Analysis

OneStream Task Activity / Load Cube presents the details of the total number of summary records in the **SelectStageSummaryTargetRows** entry.

## Direct Workflow Record Analysis

The Direct Load Status / Execution Status screen displays the Summary Row Count for each load process to display the total number of summary records generated.

## Blend Record Analysis

The Blend process differs from other Stage Import processes by generating additional data records, or rows, rather than one-to-one or summarizations. This is because the Analytic Blend designs require the generation of aggregation points, which add to the source rows. Detailed analysis of the BI Blend Processing Logs, Live Row Count statistics, and Task Activity BI Blend Load and Transform help guide the requirements for the application and systems. Refer to the *BI Blend Design and Reference Guide* for more detail.

## In-Memory Workflow Imports

### Direct Load Workflow Import

Workflow Import Type is optimized for performance by combining the Parse/Transform, Validate Transformations and Intersections, and Load Cube into a single Workflow process. This consolidation of processes, and resulting performance gains, is achieved by functioning in-memory and bypassing the overhead of writing and storing Source and Target record details to Stage database tables. The in-memory processing limits certain functionality and therefore may not be an appropriate solution for all model requirements. All other Workflow functionality remains as part of the Direct Type, such as all Business Rules, Transformation Rules and Derivative Rules.

---

#### **Direct Load Use Cases**

The Direct Load Workflow is designed for data that has a high frequency of change and does not demand durability for audit and history.

## **Common Uses**

- Data Integrations where the OneStream Metadata and Source System are mirrored, allowing “\* to \*” Transformation rules to pass-through all records, minimizing the need Drill-Down or Transformation analysis.
- Data that is “disposable” in nature. Typically, this may be data that has a high frequency of changes and may only be valid for a short time. Perhaps only valid for one to seven days.
- High-volume data loads, such as nightly batch loading, where optimal performance is desired. Such data is commonly deleted and reloaded frequently.
- Extended Application data moves, where data from a detailed application feeds a summarized target application.
- Where target data in OneStream is not required to be durable “book-of-record” data.

## **Important Limitations**

A key differentiator of the Direct Load is that it does not store source and target records in Stage Database tables. This, by design, will eliminate the audit and historical archiving of Workflow Activity. Other limitations as a consequence of the in-memory Workflow are that the Drill-Down feature will not function to support analysis of records between the Finance and Stage tables.

- Direct Load Type is not appropriate where:
  - Source file import history is required for historical reference.
  - Transformation Rule history is required for historical reference.
  - Drill-Down from Finance Engine is required.
  - Text based View member values are required to be file based.
  - Data loads are required to Append to prior Imports.

- Direct Load Type does not support historical audit of workflow history, such as Import and Transformation Rule history.
- Direct Load Type does not support Re-Transform as Import records are not stored data. Data must be re-loaded.
- Transformation and Validation analysis and map correction is limited to 1000 error records per load.
- Data files cannot load to Time and Scenarios beyond the current Workflow Scenario and Time. The data record's Time and Scenario being loaded must match the Workflow Scenario and Time.

As Direct is an in-memory Workflow with only a single step for the data integration process, Load And Transform. The Direct Load Execution Status screen provides statistics to analyze the Workflow's performance. These key statistics are helpful in determining if the Workflow design is supportive of best-practice designs to optimize application performance.

Execution Status		Completed
User Name	Administrator	
Execution Time	2/2/2021 1:07:33 PM (L), 2/2/2021 6:07:33 PM (UTC)	
Detail Row Count	896	
Summary Row Count	735	
Loaded Row Count	735	

- **Detailed Row Count:** The total number of Data Source and Derivative Rule records.
- **Summary Row Count:** The total number of records summarized in the Transformation process.
- **Loaded Row Count:** The recorded number of records loaded to the Finance Engine target Cube, which should always equal the Summary Row Count.

---

#### **Direct Load Transformations and Validations**

The Direct Load Workflow's in-memory processing results in Transformation and Validation errors that are not stored being stored in a table. The total number of errors that can be processed and presented in Validation is limited to 1000 records. If the total number of errors exceeds 1000, the data must be re-loaded to re-execute the Transformation and Validation process to generate the next batch of errors, at a maximum count of 1000 records per load.

- Total Direct Load Error Storage Limit = 50,000 records
- Error Presentation Limit = 1000 records presented per Import load

Integrations with high complexity and mapping may benefit by having a “development”, Standard Workflow, to finalize the core Transformation Rules. A Standard Workflow supports pageable Validation and Intersection Error analysis, as well as the ability to Retransform source data that the Direct Load does not. The Standard Workflow also provides the Drill-Back from the Finance Engine to Stage that may streamline the data validation process. Once the core Transformation Rules are developed, a “production” Direct Load Workflow can be used, managing only the Validation exceptions.

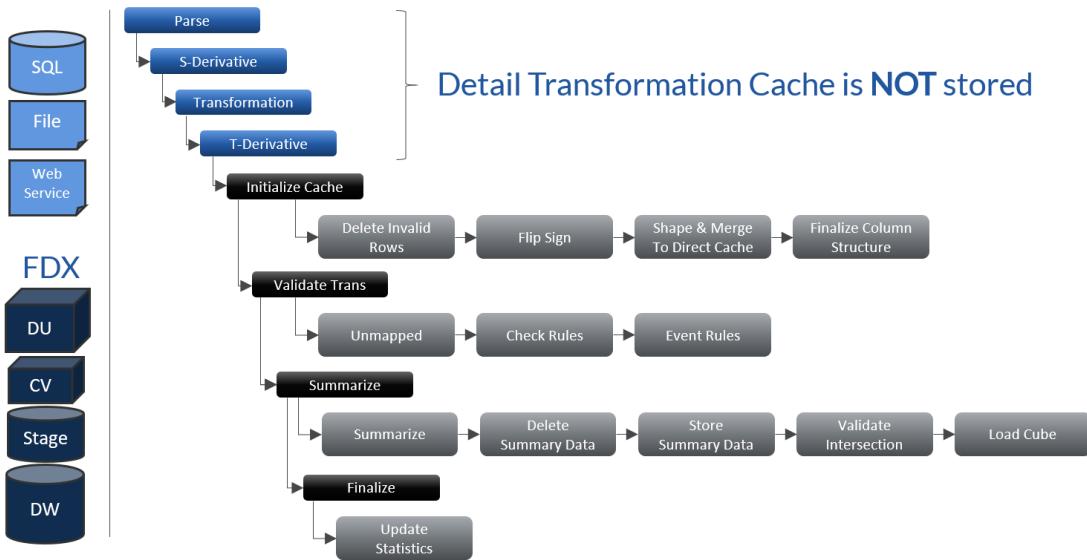
## **Direct Load Workflow Implementation**

The Direct Load Workflow Design and Requirements should consider strategies for configuration and data validation. Designers should consider the impact of the volume of source records and the complexity of mapping because of the Direct Load Workflow's lack of stored source and target details. Additionally, the Direct Load Transformation and Validation of records is limited to 1000 records per data load. Therefore, multiple imports may be required to resolve all mapping errors.

- Tightly coupled source metadata to OneStream metadata allows the use of “ \* to \* ” Mask maps to easily associate source data to cube data.
- Integrations with complex mapping may best make use of a Standard Workflow as “development” to provide transparency and as a platform for debugging.

---

### Direct Load Data Flow



### Direct Load Workflow Configuration

The Direct Load Type Workflow is an Import channel Workflow configured using various combinations of the “Direct” Workflow Name.

The screenshot shows the SAP Fiori interface for managing import profiles. The top navigation bar has tabs for 'Profile Properties' and 'Calculation Definitions'. Below this, a large table displays properties for the 'DevelopmentEU.ParisImport' profile. The left sidebar lists categories like 'Administration', 'Budget', 'Control', etc. The main table has two columns: 'Property' and 'Value'. Several fields are highlighted with red borders, specifically 'Workflow Name' (set to 'Direct'), 'StorageType = Row' (under 'Direct-Load Parameters'), and 'Substitution Text Settings'.

## Row or Blob Summarized Data Storage

The Direct Load has two options for managing results of the Transformed records that are ultimately loaded to the Finance Engine, the default being Row.

- **Row:** Transformed summary records are stored in the StageSummaryTargetData table.
- **Blob:** Summarized records are not stored directly in the StageSummaryTargetData, but as a serialized Byte array stored in the StageDirectLoadInformation.

### Sample Blob Resource Analysis

Rows	Size	Serialization & Write Time	RAM Utilization
12 Million	300-400 MB	2 Min 45 Sec	6 GB
24 Million	650-800 MB	5 Min	12 GB
36 Million	1 GB+	7 Min	21 GB

## Direct Load Error and Troubleshooting

Direct Load Import Source File Data Keys must match the Workflow Data Keys. Unlike the Standard Workflow Import, the Direct Load Import cannot load data beyond the current Workflow. As an example, an Import to M1 cannot contain records for M2 in a Monthly Workflow.

---

## Blend Workflow Import

The Blend Workflow Type utilizes the BI Blend Settings, which vary by Scenario Type. These settings allow the BI Blend Administrator to define, and optimize, the generation BI Blend tables to meet the reporting requirements.

The BI Blend Settings contain core properties used to design and structure the relational tables created by the BI Blend Engine.

BI Blend Settings Property Groups	Description
Data Controls	Defines the core data source and output structure and design of the relational tables.
Aggregation Controls	Settings to leverage Cube Dimension metadata to filter and define the relational tables.
Performance Controls	Server management and optimization settings.

## BI Blend Use Cases

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. By 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. Refer to the *BI Blend Design and Reference Guide* for more detail.

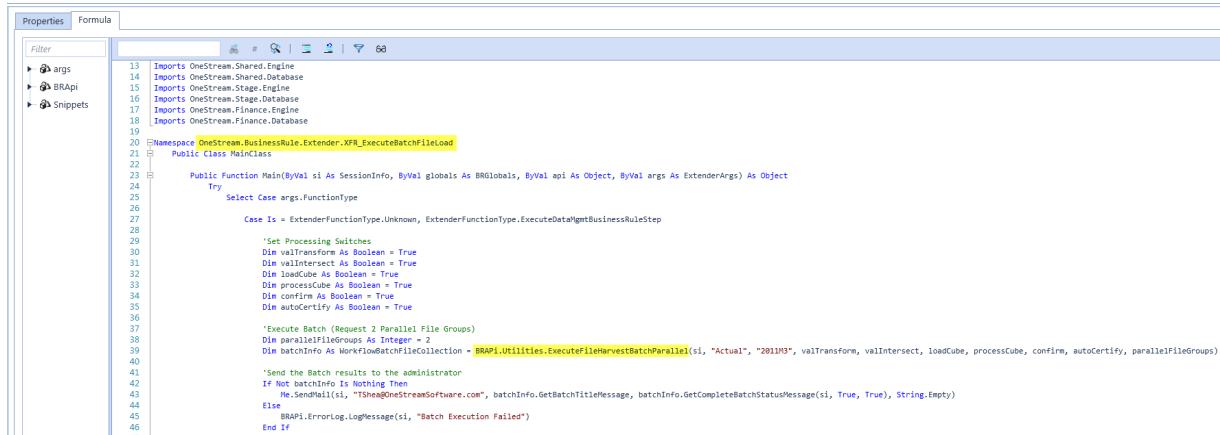
## Batch File Loading

You can import and process files all the way through the Workflow certification process. In addition, as the Workflow batch process is executed, the batch processing engine raises events used to monitor the processing and notify administrators and users of the status of the batch.

## Setting Up and Using Batch File Loading

### 1. Create Batch Processing Extender Business Rule

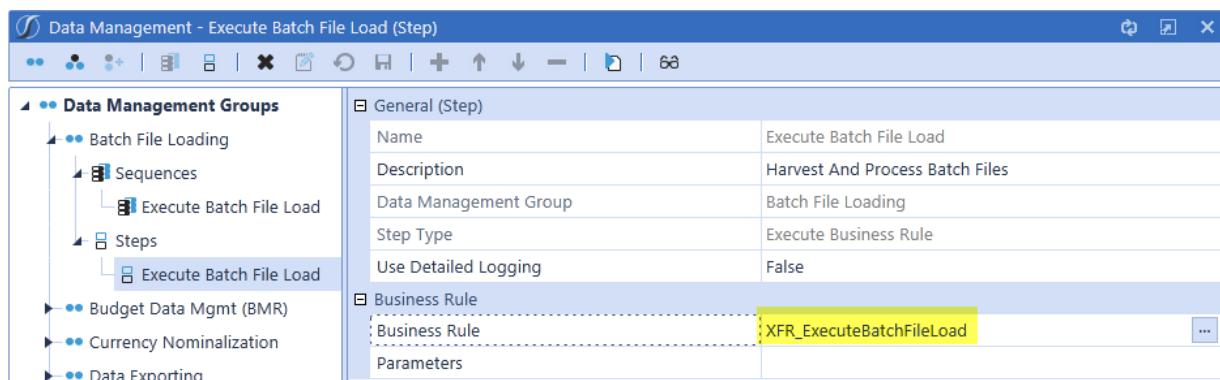
Batch file processing is executed by creating an Extender Business Rule that calls the OneStream API function BRAPi.Utilities.ExecuteFileHarvestBatch. This function also accepts switches that control the level of Workflow processing execution.



```
Properties Formula
Filter
Imports OneStream.Shared.Engine
Imports OneStream.Shared.Database
Imports OneStream.Stage.Engine
Imports OneStream.Stage.Database
Imports OneStream.Finance.Engine
Imports OneStream.Finance.Database
Namespace OneStream.BusinessRule.Extender
    Public Class MainClass
        Public Function Main(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Object, ByVal args As ExtenderArgs) As Object
            Try
                Select Case args.FunctionType
                    Case Is = ExtenderFunctionType.Unknown, ExtenderFunctionType.ExecuteDatagmtBusinessRuleStep
                        'Set Processing Switches
                        Dim valTransform As Boolean = True
                        Dim valIntersect As Boolean = True
                        Dim loadCube As Boolean = True
                        Dim processCube As Boolean = True
                        Dim confirm As Boolean = True
                        Dim autoCertify As Boolean = True
                        'Execute Batch (Request 2 Parallel File Groups)
                        Dim parallelFileGroups As Integer = 2
                        Dim batchInfo As WorkflowBatchFileCollection = BRAPi.Utilities.ExecuteFileHarvestBatchParallel(si, "Actual", "201IM3", valTransform, valIntersect, loadCube, processCube, confirm, autoCertify, parallelFileGroups)
                        'Send the Batch results to the administrator
                        If Not batchInfo Is Nothing Then
                            Me.SendMail(si, "TsheeOneStreamSoftware.com", batchInfo.GetBatchTitleMessage, batchInfo.GetCompleteBatchStatusMessage(si, True, True), String.Empty)
                        Else
                            BRAPi.ErrorLog.LogError(si, "Batch Execution Failed")
                        End If
                End Select
            Catch ex As Exception
                BRAPi.ErrorLog.LogError(si, ex.Message)
            End Try
        End Function
    End Class
End Namespace
```

### 2. Create Data Management Sequence

Batch file processing is executed by creating a Business Rule Data Management Step that calls an Extender Business Rule such as the rule example in step one.



---

### 3. Formatting File Names

Batch processing requires each file being processed to use a specific file name format to tell the batch engine how to process the file (Format: File Id-ProfileName-ScenarioName-TimeName-LoadMethod.txt). See Batch File Name Format Specification later in this section for details on creating file names that comply with OneStream's required format.

### 4. Copy Files to Batch\Harvest Folder

OneStream automatically creates a Batch\Harvest folder in each application's file share folder structure. This can be found in:

System Tab|File Explorer|File Share|Applications|Application Name|Batch|Harvest

### 5. Execute the Batch

After the files have been copied to the Harvest folder, execute the Data Management Sequence created in step one and the files will be processed.

Manual Execution

Navigate to Application Tab| Data Management, select the Batch Processing Data

Management Sequence, and click  to run the sequence.

### 6. Evaluate Batch Processing Results

Task Activity Logging

Each batch process creates a detailed Task Activity entry that provides overall status results for the batch as well as detailed information about each processed file and Workflow step.

Batch Function Return Value

In addition to writing information to the Task Activity Log, the batch processing function returns a detailed results object to the Extender Business Rule. This object provides information and can be programmatically evaluated and used to create custom reporting and notification.

### 7. Scheduling Batch Processing

Batch file processing can run using the Windows Task Scheduler or any other scheduling tool an organization may use. This is accomplished by creating a PowerShell Script to execute a batch processing Data Management Sequence when called from the specific scheduling tool.

See Data Management Automation through PowerShell in "Implementing Security" on page 252 for more information on executing OneStream Data Management Sequences from PowerShell scripts.

---

## Batch File Name Format Specification

The information below provides a detailed list for each segment of the required batch file format.

### Field Layout

File ID-ProfileName-ScenarioName-TimeName-LoadMethod.txt

aTrialBalance-Houston;Import-Actual-2011M1-R.txt

#### Field Definitions and Values

### File ID

Any text value used for file identification and controlling sort order.

### Profile Name

A valid Import Child Workflow Profile name. Use a ; to delimit Parent and Child Profile names.

Houston.Import becomes Houston;Import

### Scenario Name

This is a valid Scenario name passed to Data Sources using the Dimension data type Current DataKey Scenario. C can be passed as a substitution variable to reference the Scenario name passed in the function call: HarvestAndProcessFiles. G can be passed as a substitution variable to reference the Global Scenario name set for the application.

Using Current Scenario: A-Houston;Import-C-2011M1-R.txt

Using Global Scenario: A-Houston;Import-G-2011M1-R.txt

### Time Name

This is a valid Time name passed to Data Sources using the Dimension data type Current DataKey Time. C can be passed as a substitution variable to reference the Time name passed in the function call: HarvestAndProcessFiles. G can be passed as a substitution variable to reference the Global Time name set for the application.

Using Current Time: A-Houston;Import-Actual-C-R.txt

---

Using Global Time: A-Houston;Import-Actual-G-R.txt

## Load Method

This is a value used to control how the file is loaded.

R = Replace, A = Append

## Workflow Channels

Workflow Channels are a free form list of Members representing a logical grouping or binding point between an Input Child Workflow Profile and a specific set of accounts or the Members of a designated User Defined Dimension.

In addition, Workflow Channels are a mechanism used to increase the granularity of the standard Data Unit. They provide application designers with the ability to clear, load, and lock data at the intersection of accounts and the Members of a User Defined Dimension. See [Data Units](#) for more details.

There are three predefined default Workflow Channels when building an application: **Standard**, **NoDataLock**, and **AllChannelInput**. New metadata Members and new Workflow Profile Input Profiles are configured with default Workflow Channels. They have no effect on the granularity of application Data Units or the Workflow processes associated with clearing, loading, and locking Data Units.

## Pre-build Workflow Channels

### Standard

This is a basic Member without any special purpose other than to act as the default Workflow Channel for Account Members and Workflow Profile Input Children.

### NoDataLock

This is a special Member that only applies to a metadata Member (Account or UDx) that should not participate in a Workflow Channel grouping scheme. Therefore, this Workflow Channel should not be assigned to Workflow Profiles but only to appropriate Metadata Members. This is the default value for any UDx Member.

Assigning this Member removes it from the Workflow Channel process and allows it to function with any Workflow Profile regardless of the Workflow Channel assigned to the Workflow Profile.

---

### AllChannelInput

This is a special Member that only applies to a Workflow Profile Input Child (Import, Forms or Adj) and indicates the Workflow Profile can control data input processes for any Workflow Channel.

AllChannelInput should not be assigned to any Metadata members, but rather used to signal that this Workflow is not participating in the Workflow Channel grouping scheme. By default, this Workflow Channel is not assigned in the OneStream Application.

## Using Workflow Channels

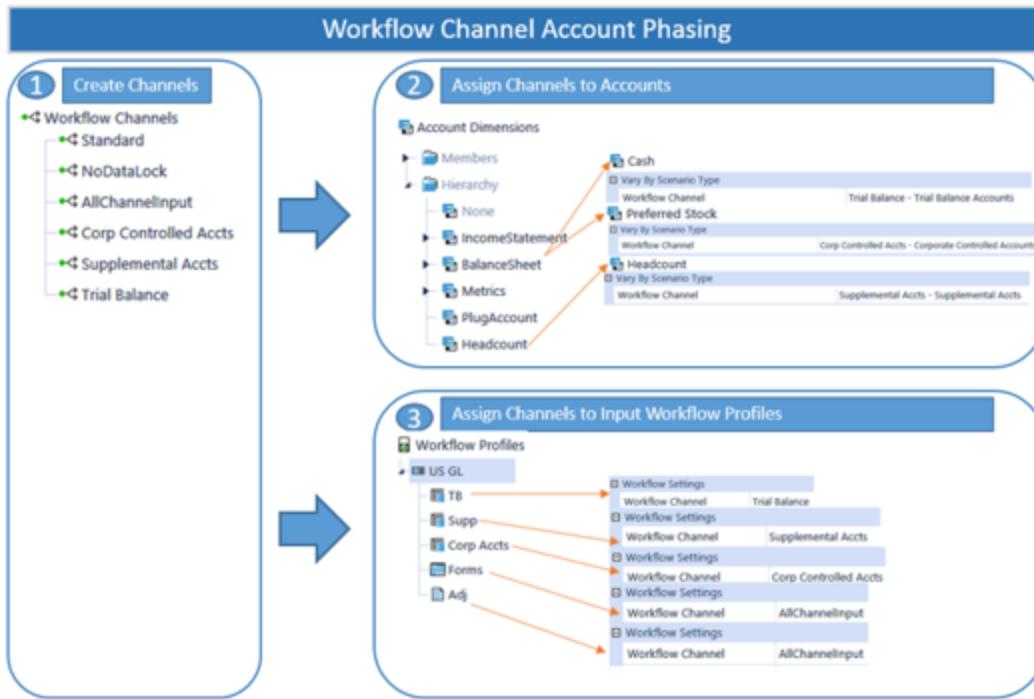
Workflow Channels allow the process to be locked down to a more granular level than the standard Workflow Profiles. This is an additional setting that can be configured to one additional Dimension. For example, this can lock down by product.

To set this, go to the **Application Tab > Tools > Application Properties**. See [Workflow](#) for more details.

Workflow Channels
UD Dimension Type For Workflow Channels (Not Used)

## Workflow Channel Account Phasing

Using a combination of Workflow Channels and Accounts enables independent Workflow control to be applied to groups of accounts. The diagram below details the steps it takes to set up a metadata and Workflow structure that isolates process management for groups of Accounts and binds specific Workflow Profiles to control the care and feeding of these groups (data clearing, data loading, and data locking).



## Setting Up Account Phasing

1. Create Workflow Channels to represent the groups of Accounts that should be linked together from a Workflow control perspective. In this example, three Workflow Channels have been created to provide separate control points for Basic Trial Balance Accounts, Corporate Controlled Accounts, and Statistical Accounts.
2. Tag each Account Member with the proper group to which it belongs. The Workflow Channel settings can vary by Scenario Type for both Metadata Members and Workflow Profile Members.
3. Tag each Workflow Profile Input Profile with the Workflow Channel it should control. This step hard wires the Workflow Profile to control data clearing, loading, and locking behaviors of the Metadata Members associated with the assigned Workflow Channel.

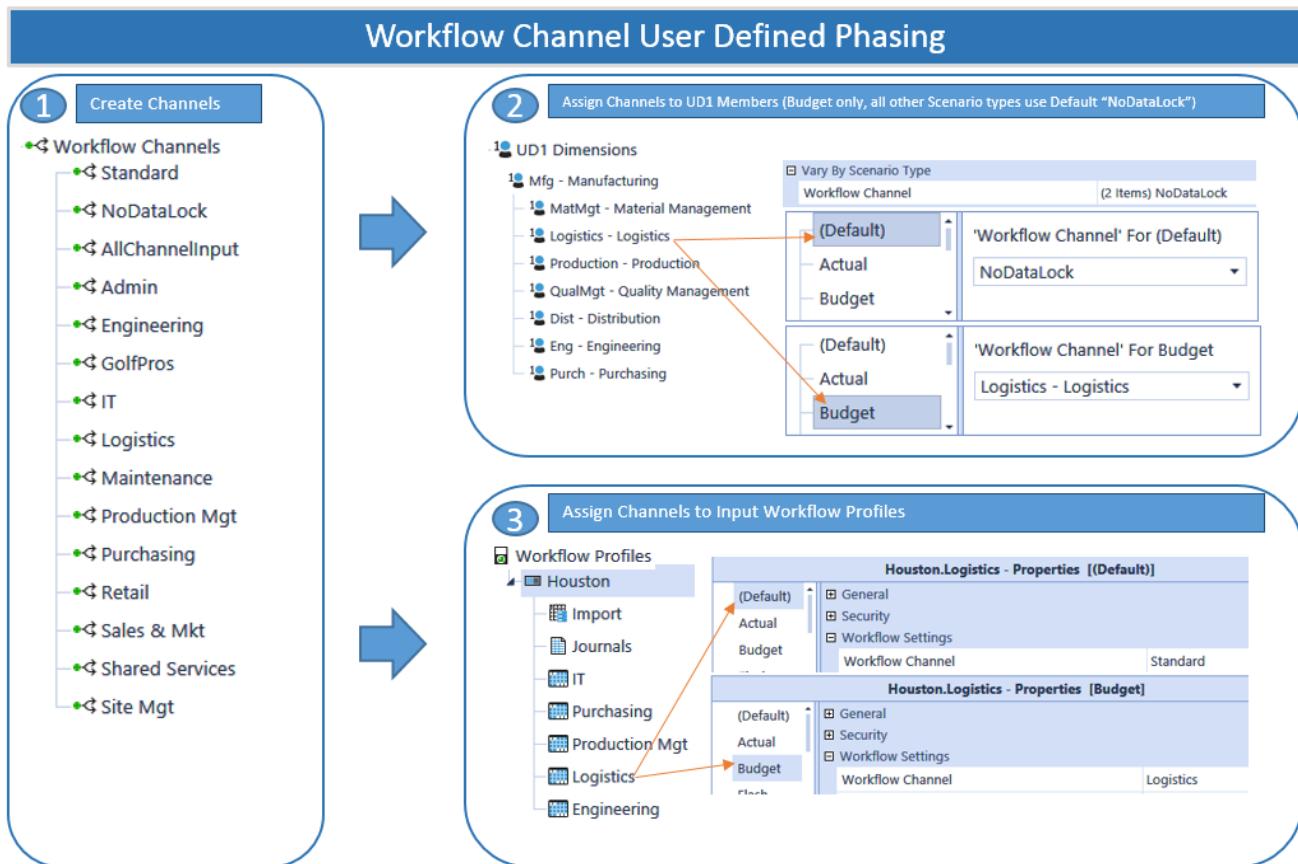
## Workflow Channel User Defined Phasing

Using a combination of Workflow Channels and a specific User Defined Dimension enables independent Workflow Control to be applied to groups of User Defined Members. Before Workflow Channels can be used in conjunction with a User Defined Dimension, a single User Defined Dimension type (UD1-UD8) must be selected as the designated User Defined type to control Workflow Channel binding. This is done in Application Tab|Application Properties|User Defined Dimension Type for Workflow Channel. This selection is made at the application level and will apply to all Scenario Types and all Cubes within the application.

The screenshot shows the 'Application Properties' dialog box with the 'General' tab selected. On the left is a tree view of categories: (Default), Actual, Administration, Budget, Control, Flash, Forecast, FXModel, History, LongTerm, Model, Operational, Plan, Sustainability, Target, Tax, Variance, and ScenarioType1. The 'UD Dimension Type For Workflow Channels' under 'Global Point Of View' is set to 'UD1'. The 'Logo File (png, height ~50 pixels)' field is highlighted with a dashed border. Other visible settings include 'Actual' for Global Scenario, '2022M1' for Global Time, 'GolfStream' for Company Name, 'N2' for Number Format, and 'CAD, EUR, GBP, USD' for Currencies.

Category	Setting	Value
(Default)	Global Scenario	Actual
	Global Time	2022M1
Administration	Company Name	GolfStream
	Logo File (png, height ~50 pixels)	<input type="file"/>
Budget	UD Dimension Type For Workflow Channels	UD1
	Formatting	
Control	Number Format	N2
	Currencies	CAD, EUR, GBP, USD
Flash	Currency Filter	<input type="button" value="..."/>
	Transformation (Default)	
Forecast	Enforce Global POV	False
	Allow Loads Before Workflow View Year	True
FXModel	Allow Loads After Workflow View Year	True
	Certification (Default)	
History	Lock After Certify	False
	Transformation (Default)	
LongTerm	Enforce Global POV	False
	Allow Loads Before Workflow View Year	True
Model	Allow Loads After Workflow View Year	True
	Certification (Default)	
Operational	Lock After Certify	False
	Transformation (Default)	
Plan	Enforce Global POV	False
	Allow Loads Before Workflow View Year	True
Sustainability	Allow Loads After Workflow View Year	True
	Certification (Default)	
Target	Lock After Certify	False
	Transformation (Default)	
Tax	Enforce Global POV	False
	Allow Loads Before Workflow View Year	True
Variance	Allow Loads After Workflow View Year	True
	Certification (Default)	
ScenarioType1	Lock After Certify	False

The diagram below details the steps to set up a metadata and Workflow structure that isolates process management for groups of User Defined Members and binds specific Workflow Profiles to control the care and feeding of these groups (data clearing, data loading, and data locking).



## Workflow Channel Combined Account and User Defined Phasing

**IMPORTANT:** When attempting to combine Workflow phasing using both Accounts and a User Defined Dimension, it requires the use of the NoDataLock Workflow Channel on Metadata Members and the use of the AllChannelInput Workflow Channel on Workflow Profiles.

Based on the examples above, if both Accounts and a User Defined Dimension are making use of Workflow Channel tagging, a situation can occur where the Workflow Channel assigned to a Workflow Profile is incompatible with either the Workflow Channel assigned to the Account Dimension or the one assigned to the User Defined Dimension.

---

This situation can be solved in one of two ways:

1. Assign the AllChannelInput Member to the Workflow Profile's Workflow Channel. This will allow the Workflow Profile to function in a more generic manner by limiting its usage to metadata Members tagged with a specific Workflow Channel. The only negative consequence of this approach is the Workflow locking for Workflow Profiles using this setting reverts to the Origin Member level which is less granular than the Workflow Channel level.
2. Sacrifice the Workflow Channel assignment of either the Account or the User Defined Dimension by assigning the NoDataLock Member. Assigning this Member will basically take it out of the Workflow Channel process and allow it to function with any Workflow Profile no matter what Workflow Channel is assigned to the Workflow Profile.

## Using Workflow Channels Across Two Base Input Workflow Profiles

The purpose of a Workflow Channel is to bind an Input Workflow Child Profile to a set of cells in a Cube by Account and a specified User Defined Dimension Member. This relationship controls Cube Data Load/Replace granularity and cell locking granularity. If Workflow Channels were used across two Import Base Input Workflow Profiles, one importing a Trial Balance and one importing Supplemental data, and the Trial Balance Workflow Profile was locked, OneStream goes through an extensive process to check data overlap. In this case, there are two distinct Data Units defined by the Trial Balance and Supplemental Workflow Channels.

### Step 1: Stage Load

The Trial Balance Workflow Channel data or the Supplemental Workflow Channel data is loaded and validated as usual. At the intersection validation step, will make sure that all intersections being processed belong to the correct Workflow Channel (Trial Balance or Supplemental). If not, the users will receive validation errors.

### Step 2: Cube Load

When the Load Cube button is clicked for either Workflows, the following algorithm runs:

It checks to see if there are any other sibling Import Workflow Profiles. If there are, it will then check for overlapping Data Units within the proposed Stage load.

### Workflow Channels in Use (One Channel per Workflow Import Child)

If the Workflow Channels are separate, only the data for the selected Workflow Profile will be written to the Cube. The data is only going into its own set of cells, so it does not need to consider the data in the other Workflow regardless if it is locked or not. Trial Balance and Supplemental can never overlap, the Workflow Channel guarantees it.

---

## **Workflow Channels Not in Use (or Same Channel Applied to Multiple Import Children)**

In this case, it is possible that the two import siblings are attempting to load to the same intersections. Consequently, the Workflow Engine will evaluate the sibling import data content to determine if they overlap and are trying to write to the same data unit.

### **No Overlap**

Load the Workflow Profile being processed because it is not overwriting sibling data.

### **Yes Overlap**

Clear all data for all assigned Entities for the Import Origin Member. Next, reload the first Import Child Workflow using Replace. Then, reload the second, third, etc., Import Children in order, so the ultimate value in overlapped data units is the cumulative value from all Import Siblings.

## **Confirmation Rules**

Confirmation Rules are used as a control to check the validity of the processed data. The rules can be setup to act as an error to the process or show a warning message. If the rule was setup as an error and it failed within the process, the user would not be able to proceed within the Workflow. The rules will process individually for each Entity associated with the Workflow Profile.

## **Confirmation Rules Properties**

### **General**

#### **Name**

The name of the Confirmation Rule group.

#### **Description**

A field for a more detailed description of the rule group.

### **Security**

#### **Access Group**

This controls the user or users that have access to the rule group within the Workflow.

#### **Maintenance Group**

This controls the user or users that have access to maintain and administer the rule group.

---

**NOTE:** Click  in order to navigate to the Security screen. This is useful when changes need to be made to a Security User or Group before assigning it to a Confirmation Rule. Click  and begin typing the name of the Security Group in the blank field. As the first few letters are typed, the Groups are filtered making it easier to find and select the desired Group. After the Group is selected, click CTRL and Double Click. This will enter the correct name into the appropriate field.

## Settings

### Scenario Type Name

The rule group can be made available to a specific Scenario Type or all Scenario Types.

### Order

The order in which the rules will process and display within the group when the Workflow is processed.

### Rule Name

A name given to the rule. This name will be seen in the Workflow, so it is best to give it a descriptive, purposeful name.

### Frequency

This option will dictate how often the rule is required to run in the Workflow Profile.

#### All Time Periods

This runs the rules every period.

#### Monthly

This runs the rules every month. If this is for a weekly application, they will run the last week of each month.

#### Quarterly

This runs the rules every quarter, or four times a year.

#### Half Yearly

This runs the rules two times a year; once in June and December.

#### Yearly

This runs the rules once a year in December.

### Member Filter

This turns on the Frequency Member Filter. Filters can then be defined in that section.

---

### **Frequency Member Filter**

This only becomes available when Member Filter is chosen in the Forms Frequency options above, otherwise this will be grey. The purpose of this option is to allow the ability to filter by time.

### **Rule Text**

A description of the rule. This will also be seen in the Workflow. The Rule Text should be a textual description of the Rule Formula associated with this rule.

### **Action**

A drop down list containing the options Warning (Pass) or Error (Fail). If the data being evaluated does not pass the rule, these options dictate how to handle the problem. If a rule is associated with the Warning action, a warning message will display to alert the user, but the process will not stop, and you will be allowed to Certify if there were no errors in other rules. If a rule is associated with the Error action, an error message will display, and the rule will have failed. You will not be able to proceed further until all failures have been addressed and/or resolved. If No Action is associated with the rule, the value for the given rule will just be displayed during confirmation. This data can be used by the user for informational purposes.

### **Failure Message**

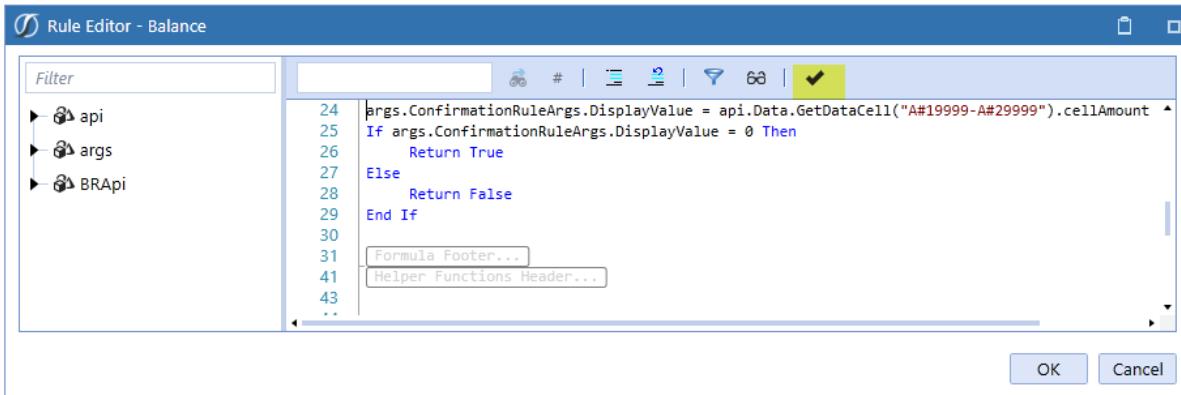
This message will be displayed to the user if an error occurs. This field will only be used if the rule is associated with the Error Action. The message should stipulate the error again and possible options of what the user should look for or do to resolve it.

### **Warning Message**

This message will be displayed to the user if the line item was associated with the Warning Error Action. The warning will alert the user to the issue.

**NOTE:** Messages longer than 2000 characters will be truncated with an ellipsis (...) placed in the last three characters.

**TIP:** In order to edit the Rule Formula, right click on the line item that needs to be edited and select Edit Rule Formulas. This will open the Rule Formula Editor and the rule logic can be created or maintained.



Click the highlighted check icon, to ensure the formula was compiled successfully.

The right-click feature also allows the user to insert or delete a row, save, and export data.

## Confirmation Rule Profiles

Once the Confirmation Rule Groups are created, they are organized into Confirmation Rule

Profiles. The Profiles are then assigned to Workflow Profiles. Choose the  icon to create a new Profile. To assign a Rule Group to a Profile, choose the  icon. This allows the user to select which Groups will be in the Profile.

Under the Rule Profile Settings, choose the Cube Name and Scenario Type where this Profile can be viewed and used when designing a Workflow Profile. Assigning a Rule Profile to a Workflow Profile is done in the Application Tab| Workflow Profiles| Data Quality Settings section.

## Certification Questions

Certification Question maintenance is the area where the repository of questions is maintained. The questions are answered by the assigned users and act as the certification to the data load process. Once the Certification Question Groups are created, they are organized into Certification Question Profiles. The Profiles are then assigned to Workflow Profiles. See "Confirmation Rule Profiles " above.

---

# Certification Questions Properties

## General

### Name

The name of the certification question group.

### Description

The field for a more detailed description of the group.

## Security

### Access Group

This controls the user or users that have access to the rule group within the Workflow.

### Maintenance Group

This controls the user or users that have access to maintain and administer the rule group.

**NOTE:** Click  in order to navigate to the Security screen. This is useful when changes need to be made to a Security User or Group before assigning it to a

Certification Question. Click  and begin typing the name of the Security Group in the blank field. As the first few letters are typed, the Groups are filtered making it easier to find and select the desired Group. Once the Group is selected, click CTRL and Double Click. This will enter the correct name into the appropriate field.

## Settings

### Scenario Type Name

The rule group can be made available to a specific Scenario Type or to all Scenario Types.

### Order

The order in which the questions will appear to the user.

### Name

A descriptive name given to the question.

### Category

A drop down list of question types or categories. Select the best option for the question type.

---

### **Risk Level**

Assign a risk level to the question. This will dictate the importance of the question as it pertains to being answered correctly.

### **Frequency**

This option will dictate how often the question is required to be answered and when it will appear to the user.

#### **All Time Periods**

This displays the questions every period

#### **Monthly**

This displays the questions every month. If this is for a weekly application, they will display the last week of each month.

#### **Quarterly**

This displays the questions for every quarter, or four times a year.

#### **Half Yearly**

This displays the questions two times a year; once in June and December.

#### **Yearly**

This displays the questions once a year in December.

### **Member Filter**

This turns on the Frequency Member Filter. Filters can then be defined in that section.

### **Frequency Member Filter**

This only becomes available when Member Filter is chosen in the Forms Frequency options above, otherwise this will be gray. The purpose of this option is to allow the ability to filter by time.

### **Question Text**

This is the question the user will answer. The question should be phrased to illicit a Yes or No response. The user is given a field to explain his/her answer in free text.

### **Response Optional**

Check this box in order to make the question optional.

### **Deactivated**

This will deem the question not active and it will not appear to the user any longer. All historical responses will be preserved if the question is not deleted.

### **Deactivated Date**

Select the date the question was or is to be deactivated.

**TIP:** By right clicking on any line item, a user can insert or delete a row, save, or export data.

# Data Collection

Data sources are built to act as a blueprint for the types of imports that need to be done and to define how the data should be parsed and imported. These include fixed files and delimited files, which use connectors to pull the data directly from the source. Data sources also have dimensions associated with them along with specific properties. In this section you will learn about these data sources and how to leverage them to build your data collection.

## Data Sources

Data sources are blueprints of the types of imports required and define how to parse and import data. Data sources can be a fixed width file, delimited file, connectors file, or a Data Management Export sequence that pull data from a source system. Once built, you can assign a data source to one workflow profile or to many workflow profiles sharing a common file format.

Associate a source file with a data source by clicking  in the upper toolbar. The file opens in the top area of the screen and you can select fields and functions to build Data Source dimension definitions.

## Data Source Properties

General properties are standard across all data types.

### Name

The name of the data source.

### Description

The field for a detailed description of the data source.

## Security

The security properties are standard across all data types.

### Access Group

Members of the assigned group have the authority to access the data source.

### Maintenance Group

Members of the assigned group have the authority to maintain the data source.

**NOTE:** Click  and begin typing the name of the Security group in the blank field. As the first few letters are typed, the groups are filtered making it easier to find and select the desired group. Once the group is selected, click CTRL double-click. This will enter the correct name into the appropriate field.

## Settings

Settings are standard across all data types.

### Cube Name

The cube associated with this data source which will dictate the available dimensions that can be used.

### Scenario Type

This allows the profile to be assigned to a specific scenario type or all scenario types. If the data source is assigned to a specific Scenario Type, it will only be available when assigned to the Workflow Profile.

## Data Structure Settings

Data structure settings are standard across all data types.

### Type

This defines the source file structure. The Type can be Fixed, Delimited, Connector, or Data Management Export Sequence. Details on these types can be found below.

#### Data Structure Type

##### Tabular

This will have a line or lines specific to a single intersection with one amount.

##### Matrix

This will have multiple amounts on a given line using rows and columns to determine the intersection that corresponds to each amount.

##### Allow Dynamic Excel Loads

If set to True, users will be able to load Excel templates as well as the data file for which the data source has been built. See the section on Using Excel as a data source for information on the proper formatting of these files.

## Fixed Files

Fixed files are in a column format with data in predefined columns.

## Connector Settings

**Connector Name**

A drop down list of available Connector Type business rules. The rules will be built containing the code to connect to and pull data from a given source system or database.

## Delimited Files

Delimited files have fields that are separated by a common character.

## Delimited File Settings

**Delimiter**

The character being used to separate the distinct fields in the source file is defined here.

**Quote Character**

In delimited files, the fields are often put in quotes in case the delimiter is also a valid character in one of the field members. This option specifies what quote character is being used in the source file.

## Connector Settings

**Connector Name**

A drop down list of available Connector Type business rules. The rules will be built containing the code to connect to and pull data from a given source system or database.

## Connectors

Connectors pull data directly from a source system or database. See "Collecting Data" on page 132 for more information.

## Connector Settings

**Connector Name**

A drop-down list of available Connector Type business rules. The rules will be built containing the code to connect to and pull data from a given source system or database.

---

### Connector Uses Files

Set to True to process files that cannot be parsed using a data source. For example, if an Excel or text file has complex formatting, this setting allows you to code the way the file should be imported rather than defining it with a data source. See "Business Rules" in "Application Tools" on page 727.

## Data Management Export Sequences

Instead of a file or connector, use a data management export sequence in a workflow import. This can be comprised of multiple data management steps. See "Data Management" in "Application Tools" on page 727 for more information.

## Data Management Settings

### Data Export Sequence Name

The name of the data export sequence being used for this data source.

## Source Dimensions

Each data source has assigned Source dimensions. To add Source dimensions to a data source,

click  Add in the toolbar and select a dimension. These dimensions correspond with the dimensions in the cube specified in data source properties.

## Source Dimension Properties

You can specify the following properties for a selected Source dimension.

## Settings

### Data Type

Not all data types will be available for every Source dimension.

### DataKey Text

This will read the value from the file as defined in the position settings.

### Stored DataKey Text

This will override the position settings and force the Time value to be a constant value for every line.

### Global DataKey Time

This will use the Time value from the current Global POV time being processed.

### Current DataKey Time

This will use the Time Value from the current Workflow POV.

### Current DataKey Scenario

This will use the Scenario value from the current Workflow POV.

### Matrix DataKey Text

This is used as a Data Type when the data source is setup as a Matrix Load with multiple periods and used to identify the defined Start Position and Length of a single period.

### Text

This will read the value from the file as defined in the position settings.

### Stored Text

This will override the position settings and force the value to be a constant value for every line.

### Matrix Text

This will have multiple amounts on a given line using rows and columns. This will determine the intersection that corresponds with each amount when more than one column contains the same dimension.

### Label

This will read the value from the file as defined in the position settings.

### Stored Label

This will override the position settings and force the value to be a constant for every line.

### Numeric

This defines the numeric amount field for the data source. This field will be read and stored as a number, not as text or string.

## Position Settings

Position settings are the definition of where the Source dimension will be found in the source file. For both fixed width and delimited files, there are tools in the toolbar and an attached file that assist in populating these values. Highlight the specific area to assign it to a dimension (for a delimited file, it only needs to be a portion of the column). The highlight will appear in red. When

the defined area is selected, click  . This will commit the selection to the dimension and the corresponding values will be populated in either the Start Position and Length fields for Fixed Width or the Column Number field for Delimited. To clear this selection without committing it to the

dimension, click .

### Start Position (Fixed Files Only)

This is the numerical representation of the starting point for a line item.

### Length (Fixed Files Only)

This defines how many characters will be taken from the start position.

A Fixed data source with a start position of 20 and a length of five will start with the 20<sup>th</sup> character and include the next five characters.

### Column (Delimited Files Only)

A delimited data source will use a column number. If the column number is four, that dimension will be represented by the 4<sup>th</sup> column in the source file.

## Connector Settings

Source field names will be provided by the Connector business rule assigned to the connector data source. These field names will either be explicitly listed out in the business rule, or dynamically returned from a SQL query. Source field names for Data Management Sequences are provided and always contain the same list.

## Logical Expression and Override Settings

### Logical Operator

Allows the ability to assign a .NET scripting functionality to a dimension of a data source.

### Complex Expression

This selection is used when .NET scripting is needed for the dimension, but not needed elsewhere. The script used in the complex expression will only be available within that dimension.

### Business Rule

This selection is used when .NET scripting is needed for the dimension and the script is available in the Business Rule Library.

### Logical Expression

This is the name of the business rule assigned to the dimension when Business Rule is selected for Logical Operator.

### Static Value

This is an override setting which allows a hard-coded value to be assigned to a dimension rather than being read from a file or data source.

## Text Fill Settings

### Leading Fill Value

Characters entered in this field will precede whatever value is brought in from the file upon import.  
Lead Fill Mask = xxx, data value = 00, results value = x00

### Trailing Fill Value

Characters entered in this field will be placed after any value brought in from the file upon import.  
Trail Fill Mask = xxx, Data Value = 00, Results Value = 00x

## Substitution Settings

### Substitution Old Value (Find)

If the value entered in this field is encountered in the dimension, it will be replaced with what is entered in the Substitution New Value.

Single Value = value1

Multiple Values = value1^value2

### Substitution New Value (Replace)

This will replace the value in Substitution Old Value if it occurs in the dimension.

Single Value = value1

Multiple Values = value1^value2

Empty String | Null|

Single Space |Space|

## Matrix Settings

This setting is only available when Matrix is set as the Data Structure Type.

### Matrix Header Values Line #

This setting indicates which row to look to for the dimension being matrixed. For example, if months are listed across columns on line four, and time is the intended Member to be matrixed, a 4 would be entered in this field.

## Data Collection

The screenshot shows a software interface for budgeting. At the top, it displays '12MonthBudget (Time)' and 'GOLFSTREAM MANUFACTURING BUDGET REPORT FOR'. The main area contains a table with columns for 'RegionProduct' and dates from '13 - Jan' to '13 - Apr'. The data includes various sales revenue entries for different bag types like Leather Bags, Travel Bags, Stand Bags, and Cart Bags. Below the table is a settings panel titled 'Matrix DataKey Text' with sections for 'Settings', 'Position Settings', 'Logical Expression And Override Settings', 'Text Fill Settings', 'Substitution Settings', and 'Matrix Settings'.

## Numeric Settings

These settings are only available in the Amount Source dimension which will help with the formatting and properties of the amount values.

### Thousand Indicator

Enter the character used to separate thousands in the value. For example, for the value 1,000 the Thousand Indicator is ",". This can also be done by highlighting the character in the file and

clicking .

### Decimal Indicator

Enter the character used to separate decimals in the amount value. This can also be done by

highlighting the character in the file and clicking .

### Currency Indicator

Enter the currency symbol for the respective currency. This can also be done by highlighting the

character in the file and clicking .

### Positive Sign Indicator

If the amount values in the file contain text characters to dictate a positive value, enter the characters here. This can also be done by highlighting the character(s) in the file and clicking 

### Negative Sign Indicator

If the amount values in the file contain text characters to dictate a negative value, enter the

characters here. This can also be done by highlighting the character(s) in the file and clicking 

### Debit / Credit Mid-Point Position

If debits and credits are in the same amount field, but are offset within that column, a midpoint can be entered here. Values to the left of the midpoint are considered a debit while values to the right are a credit.

### Factor Value

The amount being imported is factored by the value entered in this field.

### Rounding

The available options for Rounding are Not Rounded and the values 1 – 10. Not Rounded will not round the values. If a value between 1 and 10 is selected, the value will be rounded to the corresponding digit.

### Zero Suppression

If the import process should not include zero values, set this to True. To import 0 values, set this to False.

### Text Criteria for Valid Numbers

Fill in the criteria for numbers that are valid.

## Bypass Settings

Bypass allows an administrator to look for a specific value in a column or an entire line. If a value is found, that line will not be processed. In order to setup the Bypass dimension, highlight the

value to skip. Click  to skip the value only if it is found in the exact position. Click  to skip the value if it appears anywhere on a line.

## Bypass Type

### Contains at Position

This switch will tell the data source to skip an entire line of a file if the Bypass Value is found at the specified location in the Position Settings section.

### Contains Within Line

This switch will tell the data source to skip an entire line of a file if the Bypass Value is found anywhere on the line.

### Bypass Value

The value defined will indicate an entire line should be skipped when found in a specific location, or anywhere on the line.

**TIP:** Create a bypass in a fixed data source for blank spaces by specifying the position settings and entering double square brackets around the number of blank spaces in the Bypass Value. This can be used if an import will encounter an area in the data source containing blank spaces in the location specified.

General	
Name	Bp
Settings	
Data Type	Bypass
Position Settings	
Start Position	1
Length	7
Bypass Settings	
Bypass Type	Contains At Position
Bypass Value	[[ ]]

## Stored Text Settings

### Text Criteria to Bypass in Storage Buffer

This field provides the opportunity to enter a value or string of values indicating a bypass of the record being read even if you specify a stored value.

Single Value = value1

Multiple Values = value1^value2

### Stored Value Line #

The line number to be used repeatedly to obtain value for each record's importing regardless of whether it is on the line.

## Using Excel as a Data Source

An Excel file can be imported without having to configure the data source completely by setting Allow Dynamic Excel Loads to True and configuring an Excel template. See "Loading Data via Excel Templates or CSV" in "Collecting Data" on page 132 for more information.

## Transformation Rules

Transformation rules help map data from source systems to the financial model. The different member scripts and how you can use them with transformation rule types are described below. These are listed in the order in which the rules run during the Validate step in a workflow.

## Transformation Rules Toolbar

### Create Group

Use this to create a Transformation Rule Group under each dimension.

### Create Profile

See Transformation Rule Profiles.

### Manage Profile Members

See Transformation Rule Profiles.

### Create and Populate Rule Profile

Selecting this will create a Transformation Rule Group and a Transformation Rule Profile with the same name. The Rule Profile will already be populated with each Dimension Rule Group and update as the Group is updated.

### Delete Rule Profile and Groups

This can only be done by selecting a specific Transformation Rule Profile. It will delete the Transformation Rule Profile and all non-shared Groups within the Profile.

## Data Collection

---



### Delete Selected Item

Use this to delete one item such as one Rule Group or Profile.



### Rename Selected Item

This allows a user to rename Transformation Rules, Groups or Profiles.



### Cancel All Changes Since Last Save

This will undo any unsaved changes made.



### Export Selected Group to a TRX File

Use this to export a Transformation Rule Groups to use in another application or group. This exports a comma delimited file which are valid for only certain Transformation Rule designs. Logical Operators and complex expressions are not supported. Application Tools, Load/Extract of application \*.XML files should be used to manage Transformation Rules.



### Import a TRX file into the Selected Group

Use this to import Transformation Rule Groups from another application or group. This comma delimited file format may not be valid for all Transformation Rule designs. Properties such as Logical Operators and complex expressions are not supported. The \*.TRX import is intended as a utility to import Transformation Rules from compatible legacy systems. Application Tools, Load/Extract of application \*.XML files should be used to manage Transformation Rules.

## General Settings

### General

#### Name

The name of the Transformation Rule group.

#### Description

A short description of the rule or where it is used.

### Security

#### Access

Members of this group will have access to the Transformation Rule group.

### Maintenance

Members of this group have the authority to maintain the Transformation Rule group as well as load/extract from the Transformation Rules page.

**NOTE:** Click  and begin typing the name of the Security group in the blank field. As the first few letters are typed, the groups are filtered making it easier to find and select the desired group. After the Group is selected, press CTRL and double-click to add the correct name into the appropriate field.

## Settings

### Cube Dimension Name

The specific dimension to which the rule group is assigned.

## Mapping Types

Mapping Types allow the data to be mapped in different ways with the possibility of using conditional rules, wild cards, ranges and others.

### One-to-One

One-to-One mapping allows one Source dimension member to be mapped or transformed to one target dimension member explicitly. No member scripts are used.

#### Source Value

This is the value for the related Cube dimension in a defined data field.

#### Description (optional)

This is a description of the mapping rule.

#### Target Value

The Dimension Library member to which the Source Value is being mapped or transformed.

#### Order (Optional)

The one-to-one mapping rules will be processed in the order of the (defaulted) alpha numerical sort. The Order field allows a value to be assigned to a record which will allow a custom sort order of the mapping table.

## Composite

Composite mapping is a way to do a map conditionally. Dimensional tags may be used to include another dimension in the evaluation.

A#[199?-???:\*]:E#[Texas]

In the example, any similar Formatted Account starting with 199 and an entity of Texas would be mapped to a specific target entity.

## Range

A range mapping gives the upper limit and lower limit of a range. Any member that falls within this range will be mapped to the corresponding target member.

If an administrator wants the range of source Accounts from 11202 to 11209 to map to Account 12000, then enter 11202~11209 under Rule Expression with ~ as the separator.

## List

List mapping allows the user to create a delimited list of members that all map to the same target.

If an administrator wants the list of Accounts 41137, 41139 and 41145 to map to Account 61000, enter 41137;41139;41145 under Rule Expression with ; as the separator.

## Mask

Wildcards are used in mask mappings. The wildcard characters for these mappings are \* and ?. The \* character is used to represent any number of characters. 27\* would capture 270, 2709, or 27XX-009. The ? character acts as a placeholder for a single character. 27?? Would capture 2709, but would not capture 27999 or 2700-101.

Double-sided wildcards can be used as well in Mask transformation rules. For example, \*000\* would capture any account number with character(s) before and after the 000 sequence.

The following properties are standard for Composite, Range, List and Mask Mapping Types.

### Rule Name

This is a unique name assigned to each mapping rule that will also determine the default sort and processing order.

### Description (Optional)

The description of the mapping rule.

### Rule Expression

This is where the specific source field processing rule is placed. For example, 27\* would capture 270, 2709, or 27XX-009.

### Target Value

The Dimension Library member to which the Source Value is being mapped or transformed. The use of wild cards to define Target Values is not recommended. The following exceptions apply to Target Value wild card usage:

- The ? character is not supported.
- The \* character is not supported when used as a prefix (left side) to a target value.
- The \* character used as a suffix (right side) will yield the Source value.

### Logical Operator

This provides the ability to extend a normal mapping rule with VB.NET scripting functionality.

Expression Type:

#### **None (Default)**

No script is assigned or employed for this related transformation rule.

#### **Business Rule**

This selection is used when .NET scripting is needed for the dimension and the script is available within the Business Rule Library.

#### **Complex Expression**

This selection is used when .NET scripting is needed for the dimension but will not need to be used elsewhere. The script used in the complex expression will only be available within that dimension.

#### **Order (Optional)**

The Order field allows a value to be assigned to a record which will allow a custom sort order of the mapping table.

## Derivative

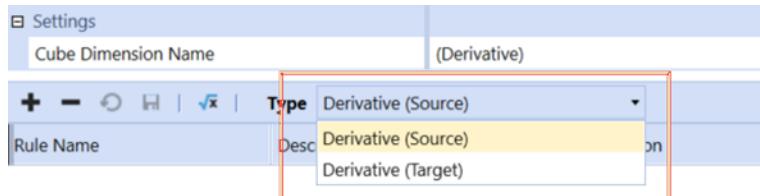
Derivative rules apply logic to stage data and generate additional records in the Stage environment. The two types of derivative rules are Source and Target.

## Source Derivative Rules

Source derivative rules are created to apply logic defined by the inbound source data in the Stage environment. This will create new members as a data record, which are stored in the stage area, to be mapped to a cube or a temporary member.

## Target Derivative Rules

Target derivative rules apply logic defined by the post-transformed stage environment data. This will create new members as a data record, which are stored in the stage area. Since these are post-transformation, any records stored as final will not be processed by transformation rules.



### Rule Name

A unique name given to a particular derivative rule.

### Description

A detailed description used in the Label field of the data imported.

## BI Blend Derivative Rules

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

- BlendUnit All
- BlendUnit Base

## Data Collection

---

- BlendUnit Parent

For more information, see the *BI Blend Design and Reference Guide*.

## Derivative Rule Expressions

Below are some examples of the syntax used to write derivative rules. The variation between the use of Source or Target derivative rules is by the definition of the rule. Source derivative rules reference the inbound record members. Target type derivative rules reference the post-transformed records. In the rule expression samples, assume these rules will run in this order presented in the example.

Import		Validate		
Account	Entity	Source Derivative	Transformation Rules	Target Derivative
10200123	TX47	A#[102*]=AdminExp	AdminExp	
10299999	GB11		ItExp	A#[*Exp]=TotExp
14000000	TX47			GenAdmExp
14300000	GB11			

Rule Expression	Expression Type	Derivative Type	Notes
A#[11*]=Cash	None	Final	Accounts that start with 11 aggregate to a new Account called Cash and stored in Stage.
A#[12*]=AR	None	Interim	

## Data Collection

---

Rule Expression	Expression Type	Derivative Type	Notes
			Accounts that start with 12 aggregate to a new Account called AR, but not stored.
A#[1300-000;Cash]=CashNoCalc	None	Interim (Exclude Calc)	The Derivative Account Cash is excluded because the calc is excluded. The CashNoCalc interim Account is created as an aggregate of Account 1300-000, but not stored.
A#[1310-000;Cash]=CashIncludeCalc	None	Interim	

## Data Collection

---

Rule Expression	Expression Type	Derivative Type	Notes
			The two Accounts (1310-000 and Cash) aggregate to equal the new Derivative Account called CashIncludeCalc because the calc is included. Note the use of the semicolon (;) as a list separator.

The following rules create additional rows in the Stage area when importing data based on logic.

Rule Expression	Expression Type	Derivative Type	Notes
A#[1000~1999]<<New_:E#[ [Tex*]=TX	None	Applies to All Types	

## Data Collection

---

Rule Expression	Expression Type	Derivative Type	Notes
			Creates a new row in the Stage for any Account that falls between 1000 and 1999 in the source data, but will add a suffix to it. Account 1010 will create a new row for Account New_1010. The end of the rule syntax shows each Entity name starting with "Tex" will be created as the Entity called TX in these new Stage rows.
<pre>A#[2000~2999]&gt;&gt;_.Liability:U2# [*]=  None:U3[*]=None:U4#[*]=None</pre>	None	Applies to All Types	

## Data Collection

---

Rule Expression	Expression Type	Derivative Type	Notes
			Creates a new row in the Stage for every Account between 2000 and 2999 with a prefix. Account 2300 will come into a new row as 2300_Liability. The rest of the rule means all UD2, UD3 and UD4 Dimension Members will be set as the None Member.
A#[3000~3999]@3:E#[Tex*]@1,1	None	Applies to All Types	

Rule Expression	Expression Type	Derivative Type	Notes
			Takes the first three digits of each Account between 3000 and 3999 to create new rows in Stage. Each Entity starting with Tex will be shown as "T" since the @1,1 syntax starts at the first position of the string and looks one character to the right.

## Logical Operator

Logical Operator provides the ability to extend a normal mapping rule with VB.Net scripting functionality.

### Derivative Expression Types

These are used to perform additional calculations on the transformed Member's data.

#### None

This is the default and no changes will be made.

#### Business Rule

## **Data Collection**

---

A business rule runs on the resulting Derivative member data. This business rule must have Derivative as its type.

### **Complex Expression**

Write a script here instead of a shared business rule and it will run against the resulting Derivative member's data.

#### **Multiply**

This will multiply the resulting Derivative member's value by another specified value.

#### **Divide**

This will divide the resulting Derivative member's value by another specified value.

#### **Add**

This will add the resulting Derivative member's value by another specified value.

#### **Subtract**

This will subtract a specified value from the resulting Derivative Member's value.

#### **Create If > x**

If the resulting Derivative Member's value is greater than a specified value, it will be created.

#### **Create If < x**

If the resulting Derivative Member's value is less than a specified value, it will be created.

## **Derivative Type**

Derivative types determine if the resulting Derivative Member will be created, not created, or if the member will be calculated.

### **Interim**

---

## **Data Collection**

---

This will not be stored in the Stage area and cannot be mapped to a target member. It can be used within other subsequently run derivative rules.

### **Interim (Exclude Calc)**

This is similar to Interim but will be excluded in other derivative calculations.

### **Final**

This will be stored in the Stage area and available to be mapped to a target member.

### **Final (Exclude Calc)**

This is similar to Final but will be excluded in other derivative calculations.

### **Check Rule**

This is a custom validation rule that uses the same syntax as member filters and can be applied to the source data during the Validation task of the workflow.

### **Check Rule (Exclude Calc)**

This is similar to Check Rule but will be excluded in other derivative calculations.

## **Order**

The Order field allows a value to be assigned to a record which will allow a custom sort order of the mapping table.

## **Lookup**

The Lookup transformation rule has a versatile configuration and can be used as a table for formulas, business rules or as a simple look up.

## Transformation Rule Profiles

Once transformation rule groups are created, they are organized into transformation rule profiles. The profiles are assigned to workflow profiles.

Click  to create a new profile.

Click  to assign a rule group to a profile so users can select groups for a profile.

Under Rule Profile Settings, choose the cube and scenario type where the profile can be viewed and used when designing a workflow profile. Assign a rule profile to a workflow profile by clicking **Application > Workflow Profiles > Import > Integration Settings**.

## Form Templates

Form templates can be setup to allow manual data entry. The entries can be done in a cube view from an Excel file or from the Spreadsheet feature (OneStream Windows App only). Each form template group has an assigned cube view, dashboard or Excel file. Forms can also be loaded using an Excel or CSV template. See "Loading Form Data" in "Collecting Data" on page 132 for more information.

## Form Template Group Properties

This section includes general group properties.

### General (Form Template Group)

#### Name

The name of the form template group.

#### Description

A short description of the template group such as how or where it is used.

## Security

#### Access Group

Members of this group have access to the Form Template group.

#### Maintenance Group

Members of this group have the authority to maintain the Form Template group.

**NOTE:** Click  to navigate to the Security page so you can modify users or groups before assigning them. Click  and begin typing the name of the Security group in the blank field. As the first few letters are typed, groups are filtered, making it easier to find the right group. Select the group, click CTRL and double-click.

## Form Template Properties

Form details can also be loaded using an Excel or CSV template. See Loading Form Data in "Collecting Data" on page 132 for more information.

### General

#### Name

Name of the form template.

#### Description

This allows for a more descriptive definition of the form template.

### Form Type

#### Cube View

Select this to utilize a cube view for the form's data entry method.

#### Dashboard

Select this to utilize a dashboard for the form's data entry method.

#### Spreadsheet (OneStream Windows App Only)

Select this for the Form to be visible using the Spreadsheet feature in OneStream Windows App. When selected, it is the same functionality as attaching an Excel file, but the spreadsheet only exists within OneStream.

#### Cube View/Dashboard

Select the cube view or dashboard that will be associated with this form template. Click  and begin typing the name of the cube view or dashboard in the blank field. Names are filtered making it easier to select items. If the name is unknown, click Cube View or Dashboard Group and scroll through the list. Select a cube view or dashboard, click CTRL and then double-click.

## Data Collection

---

### Spreadsheet (OneStream Windows App Only)

This spreadsheet should have been built with an embedded cube view or retrieve functions in order to interact with form data. See "Navigating the Excel Add-In" on page 1034 for more information on this functionality that is available within the Excel Add-in and the Spreadsheet feature in OneStream Windows App. Upload a pre-created Excel file for data entry when this

option is selected. Click  to upload an Excel file. Once selected, the file name in the Excel File space will appear.

- Click  to delete an uploaded Excel file.
- Click  to download a copy of the uploaded Excel file.
- Click  to open the uploaded Excel file.

### Excel File (Optional)

An Excel file may be used for data entry. Click  to upload an Excel file. Once selected, the file name in the Excel File space will appear. You can also:

- Click  to delete an uploaded Excel file.
- Click  to download a copy of the uploaded Excel file.
- Click  to open the uploaded Excel file.

## Workflow

Form Requirement Level

### Not Used

This setting is used if the form is no longer in use and shows the form as Deprecated in the Workflow.

### Optional

This setting will allow the user to enter data via the form if desired.

## Data Collection

---

### Required

This setting will make the form mandatory for any process to which it is assigned

### Form Frequency

#### All Time Periods

This allows the form to display every period

#### Monthly

This allows the form to display every month. If this is for a weekly application, they will display the last week of each month.

#### Quarterly

This allows the form to display every quarter, or four times a year.

#### Half Yearly

This allows the form to display two times a year; once in June and December.

#### Yearly

This allows the form to display once a year in December.

### Member Filter

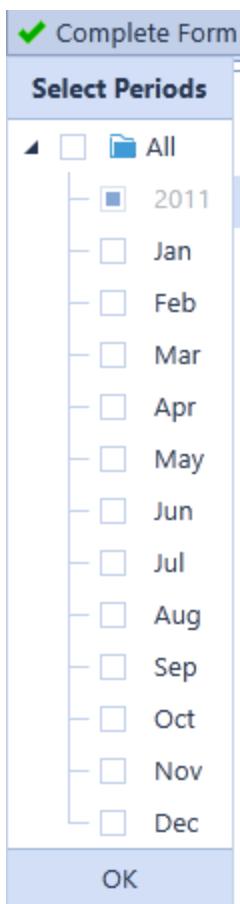
This turns on the Frequency Member Filter. Filters can then be defined in that section.

### Frequency Member Filter

This only becomes available when Member Filter is chosen in the Forms Frequency options above, otherwise this is gray for all the others. The purpose of this option is to allow the ability to filter by time.

### Time Filter for Complete Form

The Time filter will dictate how often the form is required. The example below is what will be shown when selecting Complete Form using the filter T#WFYear.Base.



## Literal Parameter Values

If the dashboard selected for the form, contains a parameter for the cube view or Spreadsheet name, enter the name/value pair which specifies which cube view the form will use in the Workflow.

Example: ParameterName = CubeViewName

See "Applying Literal Value Parameters to Form Templates" in "Collecting Data" on page 132 for more information about the Literal Parameter Value feature.

## Parameters (Cube View Forms Only)

### Parameter Type (for Parameters 1 through 6)

Parameter types allow for different values or variables to be passed to the cube view.

## Data Collection

---

### **Literal Value**

The value you specify will be hard coded.

### **Input Value**

This allows the user to enter or change the value.

### **Delimited List**

This provides a distinct list of values populated in the Parameter Type.

### **Member List**

This will produce a flat list of Members to show to the user.

### **Member Dialog**

Similar to the Member list, this allows the user to select a member, but through a pop-up Member Selection dialog which also has search capabilities. This is more appropriate for a dimension such as Accounts or Entities where the user can choose a base or Parent member by traversing a hierarchy.

### **Variable Name from Cube View**

Place a parameter in this field to replace another parameter used in a cube view. For example, a prompt such as `!MyEntityName!` will suppress the Parameter dialog from appearing when opening the form.

### **User Prompt**

This prompts the user based on the question entered here.

### **Default Type**

Set the default value so it is not blank.

### **Dimension Type**

Choose the dimension being prompted

### **Member Filter**

Allows the ability to put in a defined filter. (for example, `E#Root.WFProfileEntities`)

## **Form Template Profiles**

After the form template groups are created, they are organized into form template profiles. The profiles are then assigned to workflow profiles.

- Click  to create a new profile.
- Click  to assign a group to a profile. This allows the user to select which groups will be in the profile.
- Assign a profile to a workflow profile by clicking Application > Workflow Profiles | Form Settings, or Journal Settings for journal template profiles.

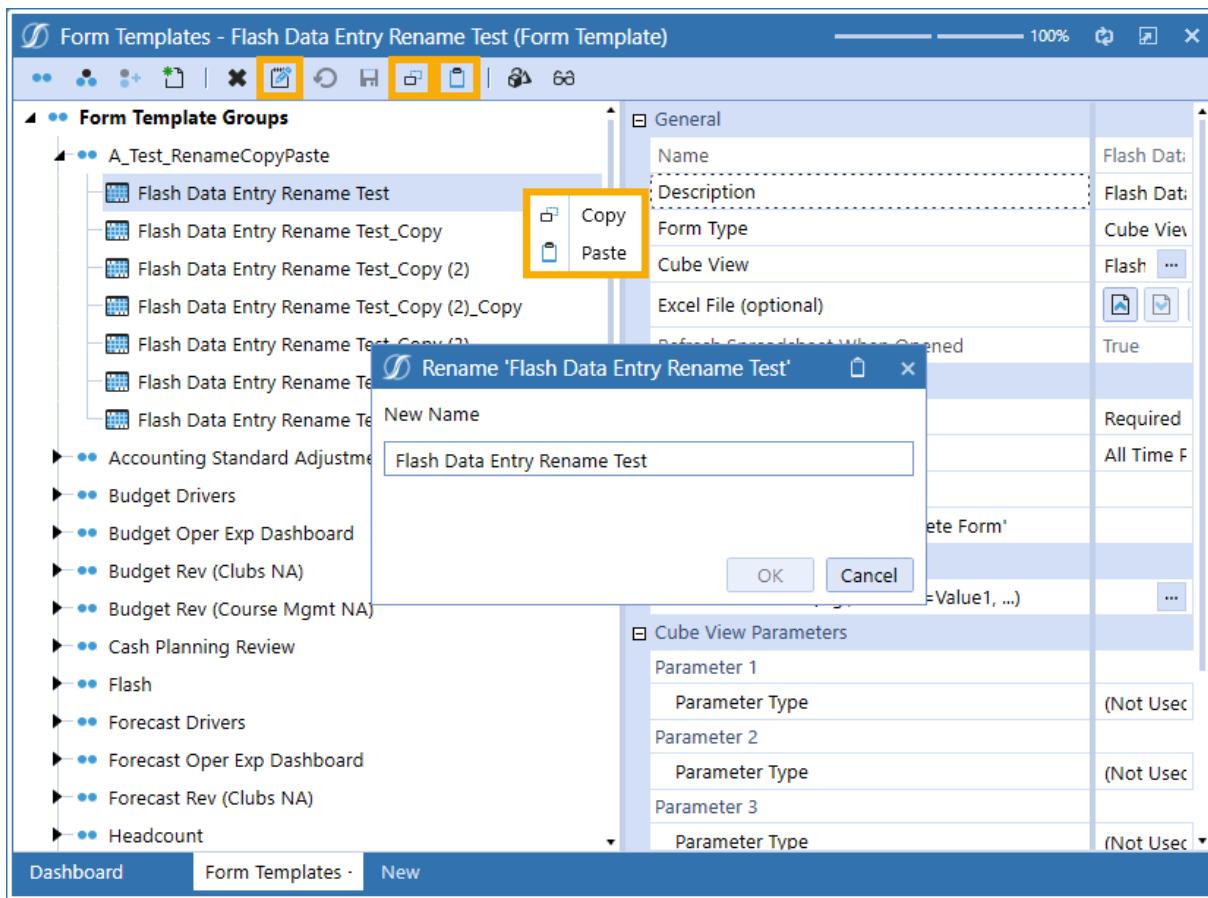
**NOTE:** Click  to access the Security screen to modify users and groups before assigning them to a form template profile. To assign groups, click  and begin typing the group name. Groups are filtered, so you can select the group, press CTRL and then double-click. This specifies the correct name into the appropriate field.

## Managing Form Templates

Form Templates can be renamed, copied, and pasted from the toolbar. Form templates can be copy and pasted using a right-click.

## Data Collection

---



Rename a form template by clicking the Rename icon from the toolbar. All associated links for the Workflow Profile and Dashboard Groups will remain valid as the form template is based on a unique identifier, not the name.

Copy a form template by clicking the Copy icon from the toolbar or right-clicking the form template name and selecting Copy.

Paste a form template by clicking the Paste icon from the toolbar or right-clicking the form template name and selecting Paste.

# Journal Templates

Create pre-set Journal Templates in Journal Template Groups. After the Journal Template Groups are created, they are organized into Journal Template Profiles and then assigned to Workflow Profiles. For more details on Profiles, see Form Template Profiles.

Journal details can also be loaded via an Excel or CSV template. See "Loading Journal Data" in "Collecting Data" on page 132 for more information.

## Journal Template Group Properties

This section lists the general group properties.

### General (Journal Template Group)

#### Name

The name of the Journal Template group.

#### Description

A short description of the Template group such as how or where it is used.

## Security

#### Access Group

Members of this group have access to the Journal Template group

#### Maintenance Group

Members of this group have the authority to maintain the Journal Template group

**NOTE:** Click  to use the Security page to modify users and groups before

assigning them. Click  and begin typing the name of the group. As the first few letters are typed, groups are filtered making it easier to find and select the desired group. Select the group, press CTRL and then double-click.

## Journal Template Properties

This section includes various journal template properties.

# General

### Name

Name of the journal template.

### Description

A detailed description of the journal template.

# Journal Template

### Journal Template Type

A journal template can either be standard or auto approved. If auto approved, a user can create a journal from the template with limited editing ability meaning permission to change name, description, and so forth. After pressing Save, the line items are validated, and the journal skips the approval process and becomes approved.

If the journal needs to be edited, it is handled like any other journal from that point forward. Someone in the Approve Journals group must reject it for the original user to edit it. Then, the regular Submit, Approve steps will occur. This approach is needed because the person who created the auto approved template only pre-approved certain numbers to be put in the system.

The user does not need to be in the ApproveJournals security group to do this. They only need to be in the ProcessJournals security group which is normally the group for end users to save or submit journals.

### Journal Template Group

This displays the journal template group that was created under the new template.

### Journal Requirement Level

This setting works together with Frequency and determines when a journal is required. The options available are Not Used (anymore), Optional, or Required.

# Journal Frequency

### All Time Periods

This allows the journal to display every period.

### Monthly

This allows the journal to display every month. If this is for a weekly application, it will display the last week of each month.

### Quarterly

This allows the journal to display every quarter, or four times a year.

## Data Collection

---

### Half Yearly

This allows the journal to display two times a year; once in June and December.

### Yearly

This allows the journal to display once a year in December.

### Member Filter

This turns on the Frequency Member filter and then the filters can be defined in that section. For example, if this journal only needs to be completed in September, use the filter T#WFYearM9.

### Frequency Member Filter

This becomes available when Member Filter is chosen in the Forms Frequency options above, otherwise this is gray for all the others. The purpose of this option is to allow the ability to filter by time.

The following settings control what the user can modify when creating a journal. Settings are True or False.

Can Change Journal Settings

Can Change Journal POV

Can change Journal Line Items

Can change Journal Amounts

## Journal

### Journal Type

The options are Standard, Auto Reversing, or Allocation. When posting an Auto Reversing journal, the auto reversal journal is automatically created in the next time period and set to the Approved state. The Auto Reversal journal has all the debits and credits reversed.

When un-posting an Auto Reversing journal, check to make sure the auto reversal is not posted first. If not, delete the Auto Reversal journal from the next time period and un-post.

An Allocation journal can be set up to perform simple or more intricate allocations, such as creating the weighting of the allocation, previewing the actual allocation entries, and un-posting them if they need to be run again.

## Journal Balance Type

### Balanced

The entire set of journal lines must balance between the debits and credits.

### Balance by Entity

Debits and credits in a multi-Entity journal must balance for each entity.

### **Unbalanced**

Balance check will not be performed. This is normally used for one-sided journals.

### **Is Single Entity**

If True, the entity name is entered in the Journal POV and all journal lines relate to this one entity.

If False, the Cube, Entity and Parent columns must be filled out for every line in the journal instance.

### **Entity Member Filter**

This Member Filter will help limit the list of entities presented to the user in the Journal POV and journal lines.

### **Point of View**

In order to limit the amount of setup for every journal line, the items that remain constant (e.g. Flow = None) can be set in the Journal POV instead of in every line item.

**TIP:** A Journal template can be repeated on a regular basis if values are placed in the journal lines and journal's settings require repeating upon a certain frequency.

# Long Running Requests

Tasks that timeout can be resolved by OneStream Support. To contact OneStream Support, register at <http://support.onestreamsoftware.com>.

For additional information, see [OneStream Web API Endpoints](#).

# **Presenting Data With Books, Cube Views and Dashboards**

You can present data using report books, cube views, and dashboards. Report Books allow you to combine a variety of reports and files in a way that fits your reporting needs. Cube views allow you to query cube data and present it in a variety of ways. Dashboards present data by combining a variety of sources such as components, data adapters, and other files. In this section you will learn how to present data using these methods.

## **Presenting Data Using Report Books**

Report Books let you combine a variety of reports and files into a single document. They are commonly used to create financial statements and management report packages. Anyone who needs a snapshot view of their company financials can run report books, but this is done most often by managers and individuals who report to management.

There are pre-built reports that can be customized to fit your reporting needs, such as:

- Cube views (which can be exported to an Excel file)
- Dashboard reports
- Charts
- Extensible documents

After creating a report book, it can be viewed and used in different contexts. For example, they can be added to other books, added to dashboards using either the Book Viewer or File Viewer dashboard component, or emailed via the OneStream Parcel Service Marketplace solution. You can also generate and view report books by running a data management sequence.

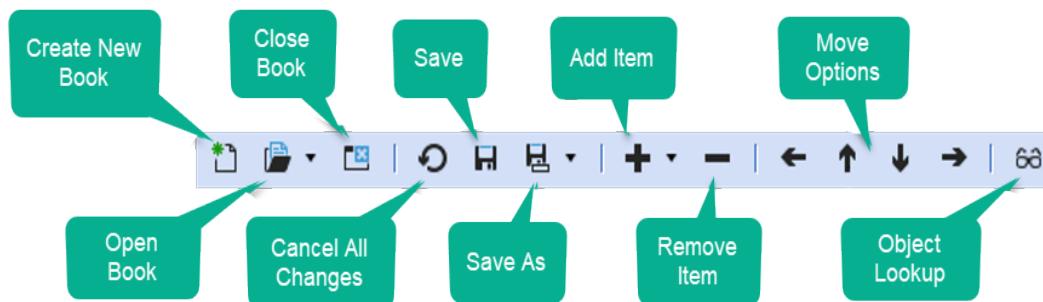
Books can be stored in locations such as a fileshare, your computer desktop, or a dashboard file. When saving a report book it is important to follow pre-defined naming conventions depending on the format of the book, such as `bookname.xfDoc.pdfBook` (PDF books), `bookname.xfDoc.zipBook` (zipped books), and `bookname.xfDoc.xlsBook` (Excel books).

**NOTE:** Embedding fonts in a PDF Report book could increase the size of the PDF file. Use the PDF Embedded Fonts to Remove property in the Application Server Configuration File to specify which fonts should not be embedded. This will reduce the size of PDF files and control the resolution during report book PDF generation. The default font-stack setting is: Arial; Calibri; Segoe UI; Tahoma; Times New Roman; Verdana.

## Book Designer Toolbar

When designing a book, you will work in the two tabs of the book designer to build and preview your book. The book designer is where you use the book toolbar icon to create a book in a hierarchical format. The OneStream Book Designer includes tools and methods that help you customize data when creating reports.

The following image displays the Book Designer toolbar.



The names of the tools describe their functions, but there are options to choose from when you click **Save As** and **Add Item**.

**Save As** lets you specify the kind of book you are creating and the location.

- **Save as File in XF File System:** Saves the book in the File Explorer. It can be saved in a public folder or a local folder on your system.
- **Save as File on Local Folder:** Saves the book in a local folder on your system.
- **Save as Application Dashboard File:** Saves the book in the File section of an application dashboard maintenance unit.
- **Save as System Dashboard File:** Saves the book in the File section of a system dashboard maintenance unit.

You can create three types of books: Excel books, PDF books, and Zip File books. Excel books produce an Excel Workbook with each piece of content on a new tab. PDF books compile all content in a single PDF file. Zip books produce a zip file where each piece of content exists individually. You can create each type of book by saving it with the appropriate file suffix.

You can specify the book type when you save the file using any of the following extensions:

- Excel books: ReportBookName.xfDoc.**xlBook**
- PDF books: ReportBookName.xfDoc.**pdfBook**
- Zip File books: ReportBookName.xfDoc.**zipBook**

**Add Item** lets you specify the kind of item to add to the book. You can add:

- Files
- Excel Export Items
- Reports
- Loops
- Condition statements (If, Else If, and Else)
- Item to Change Parameters

## Book Properties

When you are creating a new book, regardless of type, there are two book properties to consider:

- Determine Parameters from Content
- Required Input Parameters

Parameters let you determine what data should be displayed in the resulting report book. They act like filters in that only certain data is selected and displayed in the report book. Each of these items is explained in further detail below.

### Determine Parameters from Content

This item can be set to True or False. If set to True, it automatically determines the required input parameters from the book's content items. For large report books, setting this to False and manually specifying the required parameters will result in a performance gain.

## Required Input Parameters

This field requires a comma-separated list of parameter names. You can leave this field blank if Determine Parameters from Content is set to True.

## Using an Item to Change Parameters

Change parameters are used to turn off the prompt that the parameter would normally generate to define the desired result of the book. The Loop definition populates the items instead.

For example, imagine a cube view that would normally prompt you for the time member to be added to the report book. In this case, you might not want the user to make a selection, and instead, you want the report book to run for the workflow time. Rather than copying or altering the cube view, you would add an Item to Change Parameters.

## File Type

Part of the process of creating new report books is selecting a file source type. You select a file source type whenever a new file is added to a book.

You can select the following source types for your files:

- **URL:** Specify a file from an internal or external web page. You must use the full URL.
- **Application Dashboard File:** Select a file stored in the Application dashboard file system.
- **System Dashboard File:** Select a file stored in the System Dashboard Maintenance Unit file section.
- **Application Database File:** Select a file stored in the Application Database Share.
- **System Database File:** Select a file stored in the System Database Share.
- **File Share File:** Select a file from the File Share. Many file types can be added such as Word, PDF, CSV, and XLSX.

In addition to specifying the file source type, you must also set the following:

- **URL or Full File Name:** Select the URL or name of the file you want to use. Type the full URL or click the ellipsis to browse to a file.

- **Output Name:** Type a name for the file when creating a Zip book. If the .pdf extension is included for a Word or Excel file, the file will be converted into a PDF. This property is optional and does not apply to PDF books.

## Excel Export Item

When creating an XLbook, you will use Excel Export Items. You can specify a cube view to export to an Excel file. Each cube view will display on a separate worksheet in the book. These are useful when added to XLBooks rather than PDFBooks.

**NOTE:** Report and File items are not supported by Excel books and are ignored if added.

To create an Excel Export Item, set the following properties:

- **Cube View Name:** Click the ellipsis and select a cube view to add to the Excel Book.
- **Output Name:** This field is optional. Type a name for the cube view to display on the Excel worksheet. The character limit is 31.

## Report

When you add a report to a report book, you must specify a report type. This usually depends on how you have saved the book.

You can select from the following report types:

- **Cube View:** The report is based on a cube view.
- **Dashboard Chart:** The report is based on an Application Dashboard Chart.
- **Dashboard Report:** The report is based on an Application Dashboard Report.
- **System Dashboard Chart:** The report is based on a System Dashboard Chart.
- **System Dashboard Report:** The report is based on a System Dashboard Report.

**NOTE:** The chart and report items listed above are application dashboard chart components, rather than dashboard components.

In addition to selecting a report type, you must also set the following:

- **Cube View or Component Name:** Name of the Cube View, Dashboard Report or Dashboard Chart that you are using in the book. Click the ellipsis and browse to the source for a report.
- **Output Name:** Type a name for the report when it is a Zip book. This field is optional and does not apply to PDF books.
- **Include Report Margins/Report Header/Page Header/Report Footer/Page Footer:** Select True to keep the original margins, report or page headers, and report or page footers. Select False to remove the original formatting.

## Loop

A Loop is a sequence of instructions that will continually run a process as many times as is defined in the Loop Definition. For example, a book can be set up to loop through all the base entities under a particular hierarchy and generate an instance of the same Cube View Report for each entity.

### Loop Type

There are three types of Loops to use when building your book:

**Comma Separated List:** Select this option to enter values separated by a comma to be referenced later in the book hierarchy.

**Dashboard Parameter:** Select this option to use a pre-configured Parameter found in the Application Dashboards Page.

For example, a ParamSalesRegions parameter returns a list of all Sales Regions within the application resulting in the report book's Loop Variables using the same list.

**Member Filter:** Select this option to utilize the member filter builder to build a Member Filter based on dimension members.

This Loop will run each of the report book items for the specified members.

### Loop Definition

When using the loop type **Comma Separated List**, enter a comma-separated list of text values to loop over. Use brackets [ ] when names have spaces.

For example:

(Houston, Clubs, [Houston Heights])

When using the loop type **Dashboard Parameter Loop Type Used**, enter the name of a Dashboard parameter to create a list based on an existing parameter.

For example, a ParamSalesRegions parameter returns a list of all Sales Regions within the application resulting in the Report Book's Loop Variables using the same list.

When using the loop type **Member Filter Loop Type Used**, enter a Member Filter to supply a list of members to use in the loop.

For example:

E#Frankfurt, E#Houston, E#Montreal

This loops over each entity and performs the process three times.

E#[NA Clubs].Base

This loops over each base entity under NA Clubs and performs the process for however many base entities there are.

Dimension (only available when Member Filter is the Loop Type)

The name of the specific dimension used by the Member Filter. Click the ellipsis and select the correct dimension.

## Loop Variables

Create loop variables by selecting a variable from the drop down menu.

### |Loop1-4Variable|

Allows all Report Book items located in the loop's hierarchy to reference the Loop Definition's values by name.

Up to four Loop Variables can be referenced and all must be enclosed in |pipes|.

For example

|Loop1Variable|.

Use |Loop2Variable| through |Loop4Variable| to create nested loops within a loop.

### |LoopDisplay1-4Variable|

Allows all Report Book items located in the loop's hierarchy to reference the Loop Definition's values by description.

For example

|Loop1Display|

|Use |Loop2Display| through |Loop4Display| to create nested loops within a loop.

### |Loop1-4Index|

Assigns a number to the values in the Loop Definition beginning with number one, which can be referenced in the Report Book items in the Loop's hierarchy.

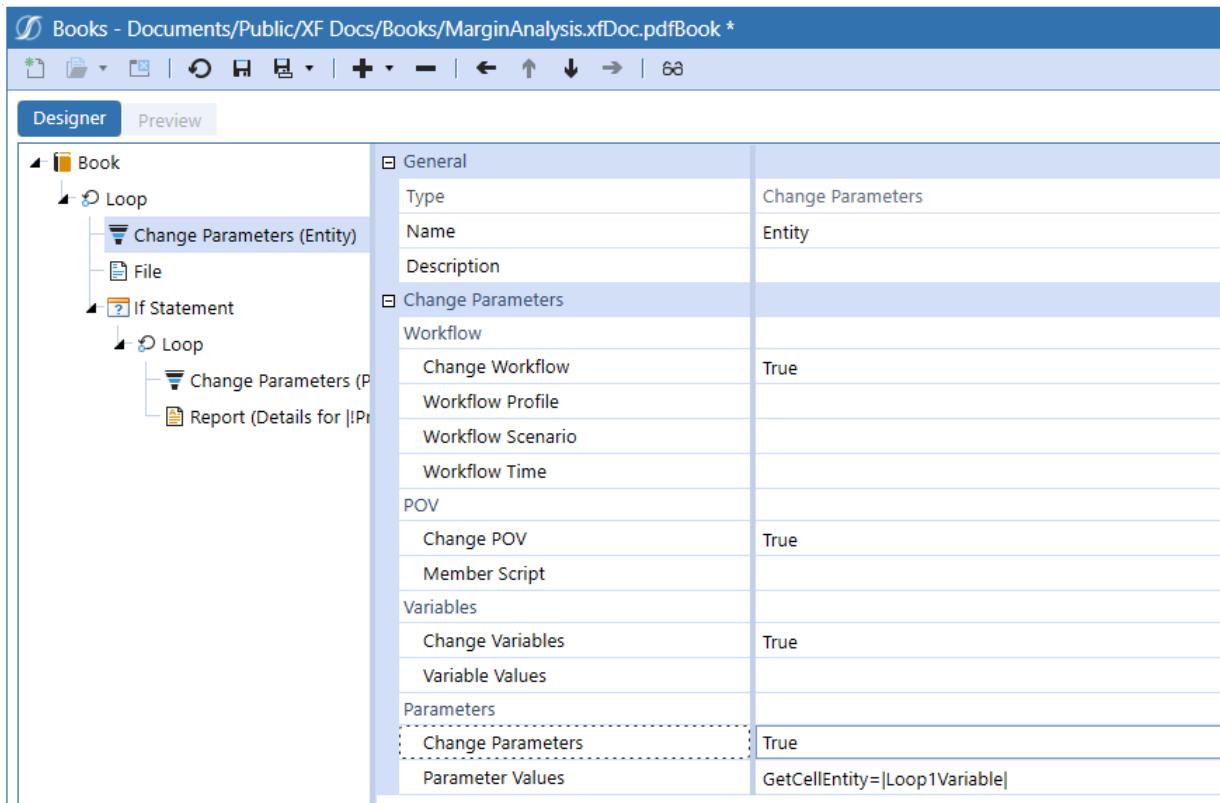
Use |Loop2Index| through |Loop4Index| to create nested loops within a loop.

## Change Parameters

Change parameters are used to enhance the data in report books. Setting these parameters customizes the report book output without altering the input from the source.

Loops require Change parameters to run, but Change parameters are not exclusive to loops. When a Change Parameter type is encountered in a loop, the Loop variable is updated to use the next Loop variable.

For example, a loop may be used to loop through a list of entities and run a report for each. The Change parameters book Item should be added here to pass along the appropriate Loop variable (such as |Loop2Variable|) which applies to each entity included in the loop.



## Workflow

**Change Workflow:** Set to True to change settings for cube views or reports driven by Workflow POV information.

**Workflow Profile:** Specify the Workflow Profile name to replace the original Workflow Profile referenced in the cube view or report.

**Workflow Scenario:** Specify the Workflow Profile Scenario Type to replace the original Workflow Profile Scenario Type referenced in the cube view or report.

**Workflow Time:** Specify the Workflow Profile Time to replace the original Workflow Profile Time referenced in the cube view or report.

## POV

**POV:** Set to True to change the book's POV without having to change the actual cube view or report.

**NOTE:** To use this feature, the POV tab in the cube view should not have any members selected.

**Member Script:** Click the ellipsis button  to launch the Member Filter Builder and enter a Member Script to change the POV. The example below changes the POV for both the entity and account.

E#[[Loop1Variable]]:A#Sales

## Variables

**Change Variables:** This serves as a placeholder that can store up to ten variables. This is valuable when If statements are used.

**Variable Values:** Enter a comma-separated list of name-value pairs to change the values of the predefined variables named Variable 1-10.

Example: Variable1=Red, Variable2=Large, Variable3=[(Loop1Variable)]

## Parameters

**Change Parameters:** Use to specify a value for a dashboard parameter found in a cube view.

**Parameter Values:** Click  to select a parameter. Enter a comma-separated list of name-value pairs to override the custom parameter's values.

Example: MyParam=Red, MyOtherParam=[|Loop1Variable|]

## If Statement

If Statements can be used to provide conditional logic to your report books. They are commonly used in tandem with loops to create a more fine-tuned output but they can also stand on their own. Additional logic can also be added through the use of Else If and Else statements. If Statements determine how book items within the hierarchy are processed.

**Statement:** Enter a conditional statement using parameters to determine whether the book items will be processed.

### Example of Using Variables in If Statements

(|Loop1Variable| = [Frankfurt])

This is an example of an If Statement that would have existing with a loop. If “Frankfurt” is found within the Loop Definition for the Loop1Variables, the book items in this hierarchy will process for Frankfurt.

### Example of Using Parameters in If Statements

(|!UserName!| = Administrator)

If the person running the book has a user name of Administrator, the book items in this hierarchy will be included in the book.

### Example of Combining Statements

(|!UserName!| = Administrator) Or (|!UserName!| = JSmith)

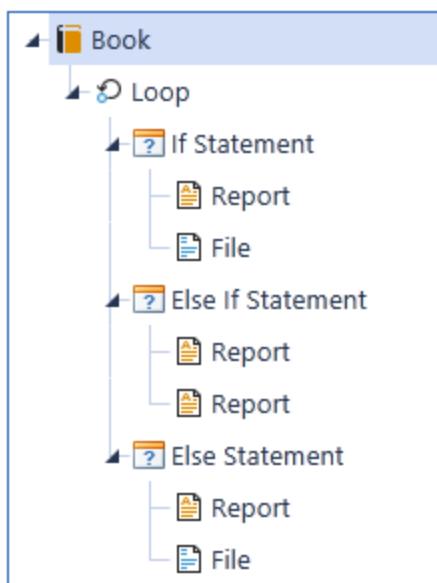
If the person running the book has a user name of Administrator or JSmith, the book items in this hierarchy will be included in the book.

## Else/Else If Statement

Else and Else If Statements, like If Statements, may or may not be used within loops to provide additional logic to the report book output. Else and Else If statements determine how book items within the hierarchy are processed. An If statement is needed to use an Else or Else If statement.

**Statement:** Enter a conditional statement using parameters to determine whether the child items will be processed. See examples in "["If Statement" on the previous page.](#)

Example:



Loop1Variable: E#[Frankfurt], E#[Europe Clubs], E#[Clubs]

**If Statement:** ( $|Loop1Variable| = [Frankfurt]$ )

If Loop1Variable is Frankfurt, run the report and file under the If statement.

**Else If Statement:** ( $|Loop1Variable| = [Europe Clubs]$ )

Else If (or otherwise if) the Loop1Variable is Europe Clubs, run the report under the Else If statement.

**Else:** Else (in all other cases) run the report under the Else statement.

## Create a Report Book

This procedure will guide you through the process of creating a basic report book.

**NOTE:** When creating books, you can drag and drop book items. This makes it easier to move items up and down the heirarchy.

1. Open OneStream.
2. On the **Application** tab, under **Presentation**, click **Books**. The Book Designer opens.
3. Click **Create New Book** on the toolbar.
4. Set the Determine Parameters from Content field to either **True** or **False**. See "Book Properties" on page 549.
5. If you set the field to False in Step 4, type the appropriate parameters into the **Required Input Parameters** field.  
If you set the field to True, leave the Required Input Parameters field blank.
6. Click **Add Item** on the toolbar and select the items you want to add to the book:
  - **Add File**: See "File Type" on page 550.
  - **Excel Export Item**: See "Excel Export Item" on page 551
  - **Report**: See "Report" on page 551
  - **Loop**: See "Loop" on page 552
  - **Conditional Statement**: See "[If Statement](#)" on page 556 and "Else/Else If Statement" on the previous page.
  - **Add Item to Change Parameters**: See Parameters.
7. If you need to remove items from the Book Designer, click **Remove Item** on the toolbar.
8. Click **Save As** and select a location.
9. Click the **Preview** tab to see the new book. A book similar to the following image is displayed.

## Presenting Data With Books, Cube Views and Dashboards

---

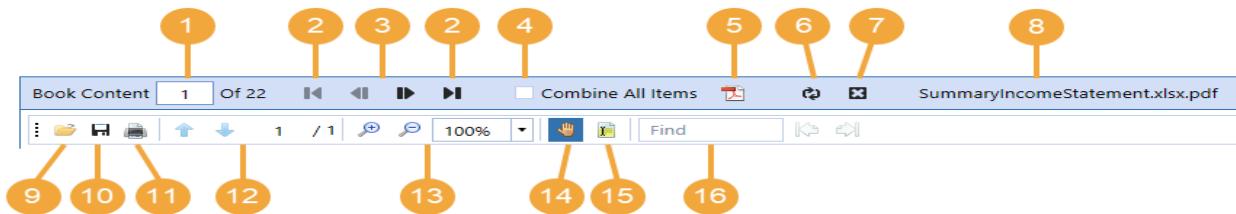
The screenshot shows the 'Book Content' tab selected in the top navigation bar. The page number '1 Of 3' is displayed. The main content area contains a 'Cash Flow Statement' for 'GolfStream™ Get Back to the Green'. The statement is dated 'Dynamic 2022' and lists various cash flow items with their amounts. The table includes sections for operations, investing, and financing activities, all showing zero values.

Cash Flow Statement	
Dynamic 2022	
Net Income	0.00
Depreciation and Amortization	0.00
Gain (Loss) on Sale of Assets	0.00
Changes in Payables	0.00
Changes in Accrued Expenses	0.00
Changes in Accrued Taxes	0.00
Changes in Accounts Receivable	0.00
Changes in Prepaid Expenses	0.00
Changes in Inventory	0.00
<b>Net Cash Provided by Operations</b>	<b>0.00</b>
Proceeds from Sale of PPE	0.00
Sale of Investments in Debt or Equity Instruments	0.00
Collection on Loan Principals	0.00
Purchase of Additional PPE	0.00
Increase in Investments in Debt or Equity Instruments	0.00
Issuance of Loans	0.00
<b>Net Cash Provided by Investing</b>	<b>0.00</b>
Issuance of Common Stock	0.00
Issuance of Preferred Stock	0.00
Issuance of LT Debt	0.00
Cash Dividends Paid	0.00

10. Click **Close Book** on the toolbar when you are done. You can open another book by clicking **Open Book** and selecting the kind of book.

## Book Preview Tool Bar

The Book preview tool bar allows you to interact and navigate in the book being previewed.



1. **Page Navigation:** This displays what page is currently being previewed in the report book. To navigate to a specific page, enter the page number and press **Enter**.
2. **First/Last:** Use these buttons to navigate to the first or last page of the report book.
3. **Previous/Next:** Use these buttons to navigate forward or backward one page.

4. **Combine All Items:** This combines the report book's pages and treats them as one content item. Use this feature in conjunction with saving or printing an entire report book. If selected, you can save the entire report book as one PDF file, print the entire book, and navigate page by page using the blue navigation arrows. If this box is cleared, only the current page will save or print, and black navigation icons will be enabled.
5. **Download Combined PDF File:** This combines the entire report book into a PDF file using the standard Adobe rendering process and less memory when attempting to display large combined PDF files. This does not require enabling Combine All Items. This is the suggested method when downloading any report book to PDF.
6. **Refresh:** Use this button to refresh the report book and select new Parameter values.
7. **Close:** Use this to close the report book in the Preview screen.
8. **File name:** Shows the name of the file currently being previewed.
9. **Open:** Use this to open a report book from a computer desktop or folder.
10. **Save:** Use this to save the current report book page. To save the entire report book as one PDF file, ensure that the Combine All Items checkbox is selected.
11. **Print:** Use this to print the current report book page. To print the entire report book, ensure that the Combine All Items checkbox is selected.
12. **Previous Page/Next Page:** Use these icons to navigate pages of a single report book content item or to navigate a report book's pages when the Combine All Items checkbox is selected. If a particular content item, such as a Cube View Report, is more than one page, use these buttons to navigate that report.
13. **Zoom:** Use these buttons to zoom in and out of a report book.
14. **Pan:** Use this button to scroll through a report book by clicking anywhere on the screen and moving the mouse.
15. **Text Selection:** Use this to select portions of the report book. This feature allows you to copy and paste. Highlight a portion of the report book, press **Ctrl+C** then **Ctrl+V**.
16. **Find Filter:** Use this to find specific keywords in a report book. Type the word in the Find Filter and press **Enter**. Use the navigation arrows to Find Previous and Find Next.

# Build Reports Through Cube Views

Cube views submit and present cube data. You can format cube views to display final reports or forms for data entry, both of which can be exported to a PDF or Excel spreadsheet. You create and maintain cube views on the Application tab in the Presentation section, but can access and run cube views from:

- Application Cube Views page
- Workflow Task (forms)
- Workflow Analysis section of a Workflow
- Cube Views section in the OnePlace tab
- Spreadsheet/Excel Add-In

You can incorporate cube views in report books, dashboards, form templates, spreadsheets, and extensible documents. That is why they are often called the "building blocks of reporting". You can also link cube views to support a more focused, granular analysis. See:

- [Build a Simple Cube View](#)
- [Member Filter Builder](#)
- [Cube View Performance](#)
- [Link Cube Views](#)
- [Analyze Data Using Passed Point of View Selections](#)

## Build a Simple Cube View

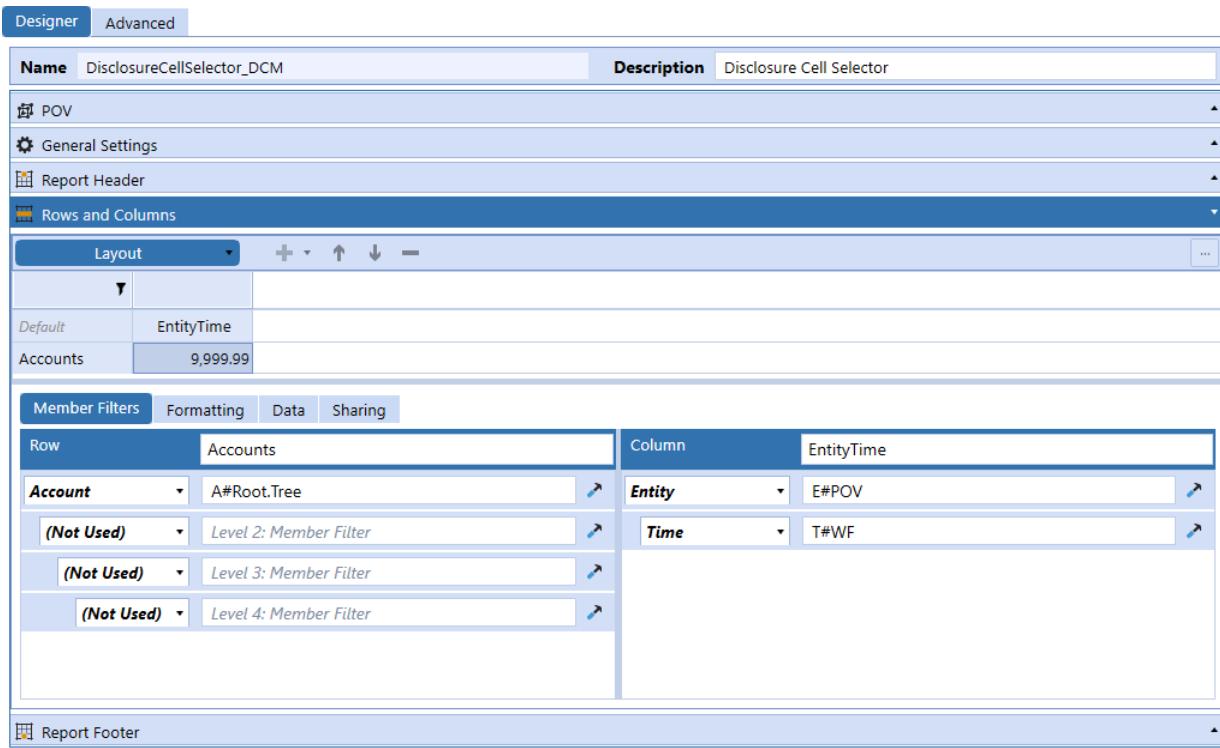
A cube view can query and submit data into a cube. They are flexible in how they present data, but this section focuses on how to build a simple cube view.

## Cube View Properties

Use the following features to manage cube views. In OneStream, go to **Application > Presentation > Cube Views** to access the Cube Views page. Then, under **Cube View Groups**, expand a cube view group, and select a cube view to access these properties.

## Presenting Data With Books, Cube Views and Dashboards

---



- **Designer and Advanced:** Each tab has the same properties, but they are organized differently. You can use either tab.
- **POV:** Make selections for the cube and all dimensions used in the cube view. You do not need to make selections for any dimensions used in rows or columns. Dimensions can also be selected in the Cube POV.
- **General Settings:** Apply row and column templates, control common settings, configure headings and formatting, and set navigation links using bound parameters.
- **Report Header:** Control what displays in the header when a cube view is run as a data explorer report (PDF style).
- **Rows and Columns:** Determine the content for the rows and columns.
- **Member Filters:** Select the dimension members, parameters, variables, expansions, expressions, and business rules.
- **Formatting:** Set the header and cell formatting for the selected row or column or for the default cell for the entire cube view.

- **Report Footer:** Control what displays in the footer when a cube view is run as a data explorer report (PDF style).

## Cube View Toolbar

Use the following toolbar items to manage cube views.

-  **Create Group:** Organize cube views.
-  **Create Profile:** Organize cube view groups.
-  **Manage Profile Members:** Assign cube view groups to cube view profiles.
-  **Create Cube View:** Create a new cube view in a cube view group.
-  **Delete Selected Item:** Delete a selected item, such as a cube view or cube view group.
-  **Rename Selected Item:** Rename a selected item.
-  **Cancel All Changes Since Last Save:** Cancel unsaved changes.
-  **Save:** Save changes.
-  **Copy Selected Cube View:** Copy the selected cube view to use it as a template for another cube view.
-  **Search:** Search for cube views.
-  **Open Data Explorer:** Run a cube view and see it in a data explorer report.
-  **Show Objects That Reference The Selected Item:** View other areas where the selected item is used. For example, see where a cube view is being used in an application to know the impact of a change. This icon is available on other pages as well.
-  **Object Lookup:** Find and select an object to copy.

## Manage Rows and Columns

Access these additional items on the Advanced tab and the Rows and Columns slider to manage rows and columns.



**Add Row or Column:** Add rows or columns to a cube view.



**Move Up:** Move rows or columns up.



**Move Down:** Move rows or columns down.



**Remove Selected Row or Column:** Remove a row or column.

## Create Cube View Groups and Profiles

Groups organize the artifacts that users are building. Cube views are organized into cube view groups. A cube view group needs to be created before a cube view, and it can include one or more cube views. Profiles further structure the artifacts and enable them to be used in other places in the application and accessed by users. Cube view groups are organized into cube view profiles that are assigned to different areas of the application, such as a workflow profile. Cube view profiles can include one or more cube view groups.

1. On the **Cube Views** page, click **Create Group**.
2. Type a name for the new group.
3. Click **Save**.
4. Click **Create Profile**.
5. Type a name for the new profile.

**TIP:** Use the **Visibility** drop-down menu to specify where cube views can be viewed in the application. The default is Always, which displays the cube views in all listed locations. If you select a specific location, the cube views will display in that location. If you select Never, the cube views will only display in the Cube Views page in the Application tab.

6. Click **Save**.

## Cube View Group Properties

Cube view group properties apply to general and security settings.

### General (Cube View Group)

#### Name

The name of the cube view group.

#### Description

A short description of how the group is used, or what it contains.

#### Workspace

The currently selected workspace.

#### Maintenance Unit

The currently selected maintenance unit.

## Security

#### Access

Members of this group can access the cube views in the cube view group.

#### Maintenance

Members of this group have the authority to maintain the cube views in the cube view group.

**NOTE:** Click  and begin typing the name of the security group in the blank field. As the first few letters are typed, the groups are filtered making it easier to find and select the desired group. Once the group is selected, click CTRL and double-click. This will enter the correct name into the appropriate field.

## Assign Cube View Groups to Cube View Profiles

Assign cube view groups to cube view profiles to control how the cube views in the group are accessed, using the visibility cube view profile property.

1. On the **Cube Views** page, under **Cube View Profiles**, select a profile.
2. Click **Manage Profile Members**.

3. Under **Available Groups**, select the group and then click the arrow to move it to the right.
4. Click **OK**.

**NOTE:** You can assign a cube view group to multiple cube view profiles.

## Create a New Cube View

1. On the **Cube Views** page, under **Cube View Groups**, select a group.
2. Click **Create Cube View**.
3. Type a name and description for the cube view.
4. Click **Save** and specify [General Settings](#).

## Copy a Cube View

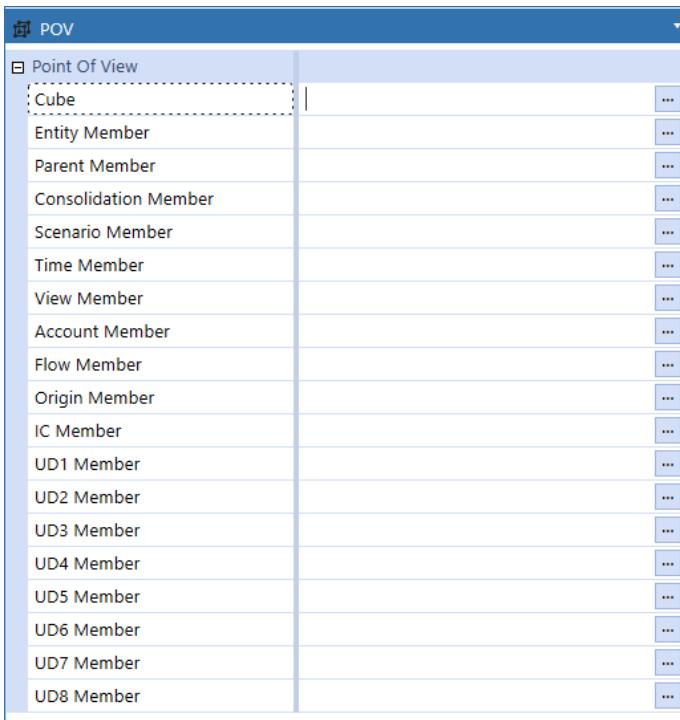
For efficiency, you can copy a cube view to use its formatting settings rather than creating a new cube view. A copy of a cube view is not linked to another cube view.

1. On the **Cube Views** page, under **Cube View Groups**, select a cube view.
2. Click **Copy Selected Cube View**.
3. Select the cube view group from the drop-down menu.
4. Enter a unique name, not used by another cube view, and click **OK**.

## Set the Cube View POV

Every cube view needs a POV setting to retrieve data. To query data in a cube view, set your Cube View POV, rows, and columns. The Cube View POV will set the cube and all of the dimensions used to query the various cells of the cube views.

1. On the **Cube Views** page, under **Cube View Groups**, select a cube view.
2. Select the **POV** slider. Enter the information individually by typing in the fields or clicking the ellipsis for each member, or move the entire Cube POV.



To move the entire Cube POV, on the **POV Pane**:

- a. Right-click **Cube POV**.
- b. Click **Copy Cube POV**.
- c. Right-click the light blue **Point Of View** header.
- d. Click **Paste POV**.

**TIP:** To move an entire cube POV, you can also drag and drop the Cube POV from the POV pane into the Cube View POV.

**NOTE:** The information in the POV slider takes priority over the information in the POV pane.

## Create Cube View Rows and Columns

1. To set the cube view rows and columns, select the **Rows and Columns** slider. The preview grid for the new cube view displays default row and column names: **Row1** and **Col1**. You can edit these names to make them more descriptive.

Layout	
Default	Col1
Row1	9,999.99

2. Click **+** to add rows or columns, and click **-** to remove rows or columns.
3. Select a row, column, or intersection in the preview grid, and then set the display options by using the settings tabs. The member filters, formatting, data suppression options, and overrides can be applied at the row and column level.

Member Filters	Formatting	Data	Sharing	Column Overrides
Row	Row1			
(Not Used)	Level 1: Member Filter			
(Not Used)	Level 2: Member Filter			
(Not Used)	Level 3: Member Filter			
(Not Used)	Level 4: Member Filter			

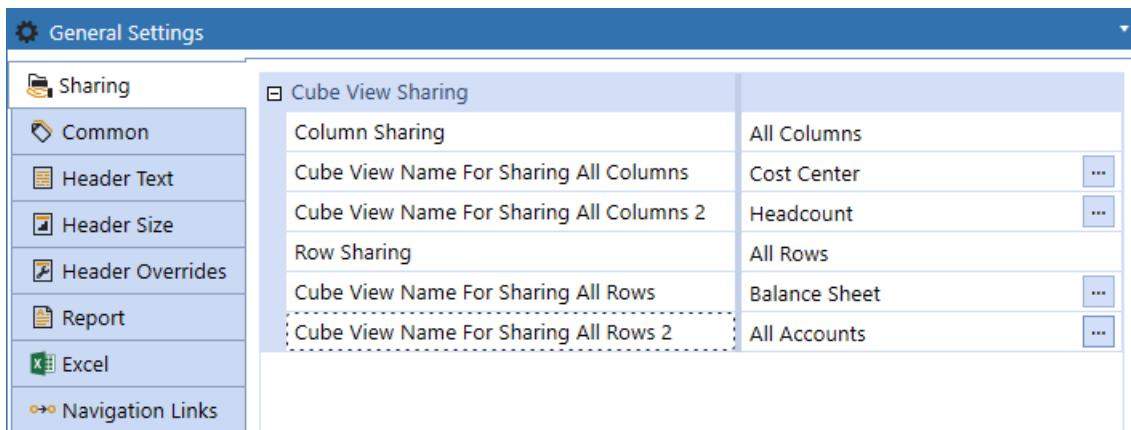
4. Click **Save**.

**NOTE:** See [Member Filter Builder](#).

## Create a Cube View Using Shared Rows and Columns

An administrator can share all rows and columns or selected rows and columns from another cube view. Sharing is more efficient when building and managing cube views.

1. On the **Cube Views** page, under **Cube View Groups**, select a cube view.
  2. Go to **General Settings > Sharing**.
  3. In the rows for the **Row Sharing** and **Column Sharing**, click in the fields to the right to access the drop-down menus. Select the option to share all or specified rows or columns.
    - **All Rows/Columns:** Enter the source cube view name of the rows and columns to be shared.
    - **Specified Rows/Columns:** Reference a single row or column from the source cube view when designing rows and columns.
- TIP:** You can share rows and columns from two cube views. If you share rows or columns with another cube view, updates are automatically linked. This can make building a cube view more efficient.
4. If you selected the option to share all rows or columns, click the ellipsis and then select the cube view. Click **OK** to confirm.



5. Click **Save**.

## Use a Cube View Template

Creating and using cube views as templates can increase efficiency and consistency for cube views that have similar formatting and properties for the rows and columns. If you plan to use templates, create a cube view group specifically for this purpose. For example, create a cube view group with the name template (Templates\_Columns or Templates\_Rows) and save cube views to use as templates in this cube view group. Then, other users can find and use the templates.

1. On the **Cube Views** page, under **Cube View Groups**, select a cube view to use as a template.
2. Click **Copy Selected Cube View**.
3. Select the cube view group from the drop-down menu and type a name for the new cube view.
4. Click **OK**.

**NOTE:** Each cube view must have a unique name.

## Cube View Shortcuts

Use Cube View shortcuts to launch other cube views. This is helpful when an administrator wants to use the same cube view but does not want to use the same parameters.

For example, a user can launch a cube view for an Income Statement and be prompted with a ParamView parameter. The ParamView has two values of YTD or Periodic, meaning an Income Statement can be launched to show the data with a Year to Date view or a Periodic view (such as Month to Date).

In this case, a shortcut can be used to save two versions of the same report without prompting the user or having to maintain two cube views. In each case, the shortcut cube view Name would be the same (such as Income Statement), but the literal parameter value would be different. The YTD version would be ParamView = [YTD] and the Periodic version would be ParamView = [Periodic]. Each would open the Income Statement cube view to the proper view settings without prompting the user.

## Member Filter Builder

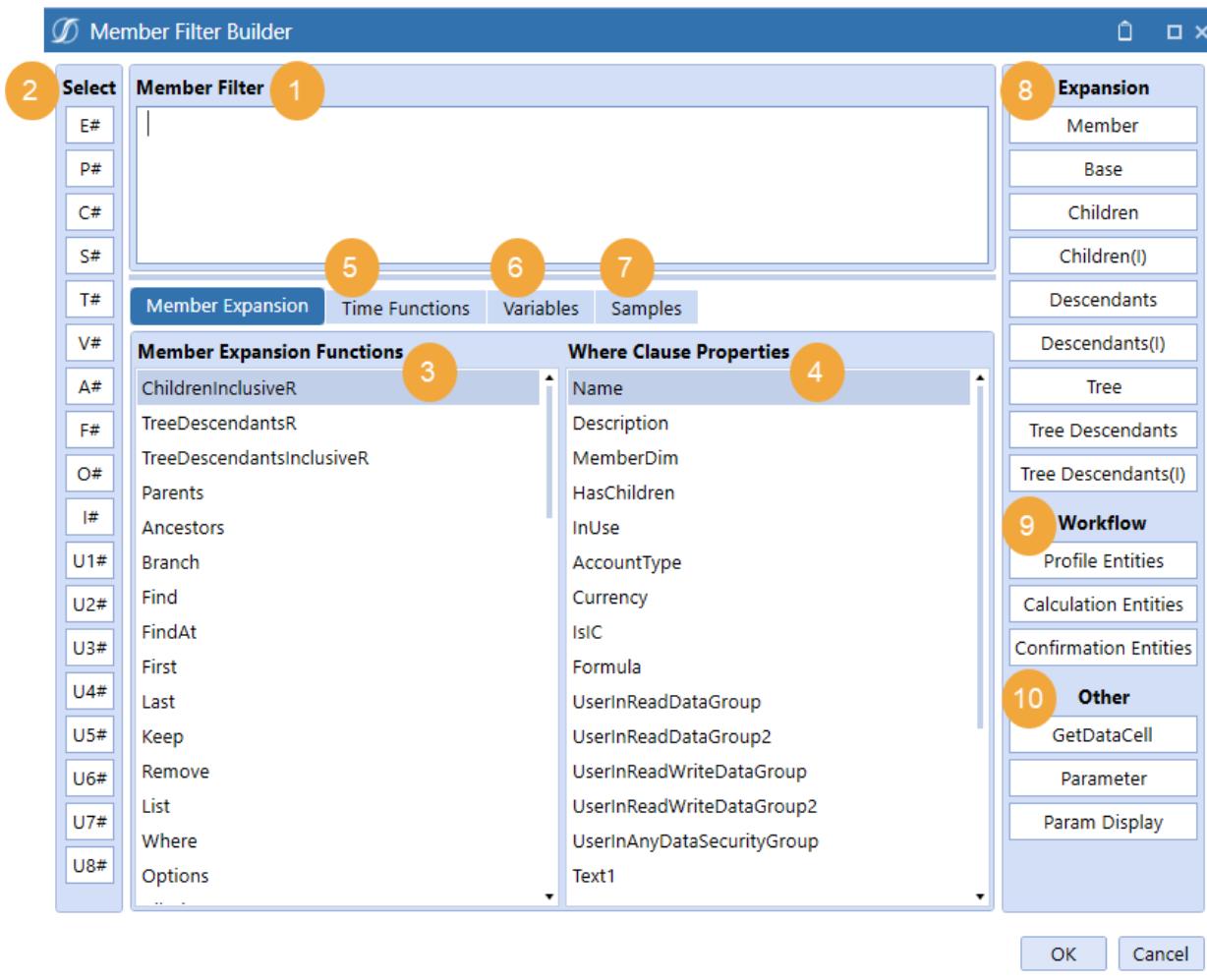
The Member Filter Builder dialog box makes it simple to build complex member filters without having to remember or look up the proper syntax.

On the **Cube Views** page, select a cube view. Select **Rows and Columns** and then select a row or column header. Click the **Member Filters** tab and then **Member Filter Builder** to launch the dialog box.

## Navigation

Hover over the items in the Member Filter Builder dialog box for additional information.

## Presenting Data With Books, Cube Views and Dashboards



- 1. Member Filter:** Location where the member filter is built. Type in this section or use the other items in the dialog box to enter information into the field.
- 2. Dimension Tokens:** There is a button for each dimension that launches the appropriate selection dialog box.
- 3. Member Expansion Functions:** Double-click a member expansion to add it to the member filter.
- 4. Member Expansion Where Clause:** If you select Member Expansion Where, use the where clause properties to complete the expression.  
Example: UD2#AllProducts.Children.Where(Name Contains Clubs)

5. **Time Functions:** Only applies to the Time dimension, such as T#POVPrior1. Double-click a time function to add it.
6. **Substitution Variables:** Double-click a system-wide substitution variable to add it.
7. **Samples:** Refer to this tab for example syntax for complex queries and calculations, including how to build member expansions, where clause expressions, time expressions, calculations (GetDataCell and column/row expressions), custom member list expressions, and XFBR, XFCCell, and XFGetMemberProperty.
8. **Expansion:** Commonly used member expansions added to the end of filters. Double-click the expansion to add it.
9. **Workflow:** Commonly used workflow member expansions used in cube views that point to a report, form, or dashboard and are affiliated with a specific workflow profile.
10. **Other:** Commonly used member filter functions that enable you to create calculated rows and columns or use a custom parameter to store member lists.

## Member Expansion Tab

To reduce cube view maintenance, use simple member expansions and connect your cube views to the metadata when complex expansions are needed. Avoid referencing individual members.

- **Reverse order expansions:** These selections include ChildrenInclusiveR, TreeDescendantsR, and TreeDescendantsInclusiveR.
- **Where clauses:** Enable further qualification of the results and can pull members based on properties such as "name contains" or text properties.
- **Remove functions:** Removes some of the members from the results.
- **Parents:** Returns the direct parent of the member. There may be multiple parents if there are alternate hierarchies.
- **Ancestors:** Returns all members up the chain from the original member.
- **Options:** Advanced option if you are a report builder familiar with the extensibility design in your application. This expansion is commonly used to focus on one portion of your extensible dimensionality.

This is not an exhaustive list of all the member expansions. Each expansion can also be found in the Samples tab. Hover over each to see the purpose and a sample of the expected syntax. Double-click the member expansion to place the sample syntax in the Member Filter field.

Use the member filter to filter data by creating a list of restricted members. Member filters can contain multiple member scripts. A member script queries a defined set of dimensional members. Members can be specified for any or all dimensions and the primary dimension can also specify a member expansion formula (for example, Descendants).

For example, a simple member script that returns the year 2022: T#2022.

For example, a member script with a member expansion that returns all the income statement accounts: A#[Income Statement].Descendants.

If one or a few dimensions in the member script have a dimension token, the remaining dimensions are pulled from the Cube View POV, the Global POV, or Workflow, Time, and Scenario. Separate each dimension that has a token in the member script with a colon as in the following example:

Cb#GolfStream:E#Houston:P#Texas:C#Local:S#Budget:T#2022M3:V#YTD:A#60000:F#None:  
O#Forms:IC#None: U1#Sales: U2#HybridXL: U3#Northeast: U4#SportsCo: U5#None: U6#None:  
U7#None: U8#None

For example, a member filter that contains three different member scripts that returns the Actual, Budget, and Forecast scenarios: S#Actual, S#Budget, S#Forecast.

## Time Functions Tab

Time is a fixed dimension and is based on the time dimension type associated with the application. OneStream software has custom time dimensions (for example, weekly and monthly), which affect time functions and expansions. Time functions pivot the member, and time expansions extend the member.

- Time function example to display the prior period of the time member in a Cube POV: T#POVPrior1
- Time expansion example to display all months in the year 2022: T#2022.Months

Add a time function from the Time Functions tab:

1. In the **Member Filter Builder** dialog box, click the **Time Functions** tab.
2. Select the type of time function to display the examples available: POV, WF, Global, or General. You can also select All.

3. Double-click a function to populate it in the **Member Filter** field.

**NOTE:** Click the **Samples** tab to view more time functions.

The following time function example demonstrates a style of syntax that provides flexibility. You can choose from a variety of time functions, but this one enables you to pivot easily on the year and period separately:

T#YearPrior1(|PovTime|)PeriodNext1 (|PovTime|)

- Prior year from the POV time: **T#YearPrior1(|PovTime|)**
- Next period from the POV time: **PeriodNext1 (|PovTime|)**

## Variables Tab

Substitution variables are short scripts that use pipe characters to include a predefined substitution variable. For example, |WFProfile| would refresh to display the current workflow profile name. They come with every installation of OneStream and cannot be edited, so you do not need to create or maintain them.

Substitution variables can be used throughout the application. For cube views, you can use them for:

- Headers and footers
- Rows and columns
- Cube view page captions

Substitution variables are always referenced with pipes (for example, |POVTime|). The Member Filter Builder dialog box includes substitution variables in the following categories:

- POV (Cube POV)
- WF (Workflow POV)
- Global (Global POV)
- CV (Cube View POV)

- MF (Member Filter)
- General (items not related to a POV)

When using a substitution variable to return a member name, the prefix indicates where the value of the variable is pulled from. Choose if it should refresh based on the Cube View POV (CV), Workflow POV (WF), Cube POV (POV), Member Filter (MF), or Global POV (Global).

Choose if you want to display the member description (usually in headers and footers) by adding Desc as the suffix. Time has an added short description option that is set through the time profiles. For example, |WFTimeDesc| returns the description Feb 2022, and |WFTimeShortDesc| returns the short description Feb.

Common substitution variables:

- Username that ran the report: |UserName|
- Cube view name: |CVName|
- Members text properties: |Text1|
- Today's date: |DateDDMMYYYY|

On the Variables tab, you can double-click a substitution variable to add it to a member filter. You can also copy a substitution variable from the Object Lookup dialog box:

1. On the **Cube Views** page, click **Object Lookup** to open the dialog box.
2. Under **Object Type**, select **Substitution Variables**.
3. Select a substitution variable from the list. Use the Filter field to find a specific option.
4. Click **Copy to Clipboard** and then paste the substitution variable where needed.

## Samples Tab

The Samples tab holds example syntax for more complex queries and calculations in the following categories:

- Member Expansions
- Where Clause Expressions
- Time Expressions

- Calculations (GetDataCell Expressions and Column/Row Expressions)
- Custom Member List Expressions
- XFBR, XFCell, and XFGGetMemberProperty

Add an expression from the Samples tab:

1. In the **Member Filter Builder** dialog box, click the **Samples** tab.
2. Double-click a sample to populate it in the **Member Filter** field.

## Where Clauses

Where clauses are commonly used in reporting to create a more flexible query. The where clause can pull members based on properties:

- Text properties
- Portions of the description or name (for example, starts with or contains)
- Security
- Account types (account dimension only)
- Intercompany (entity and account dimension only)
- Specific currency (entity dimension only)
- In use property
- Has children
- Has a member formula

**NOTE:** Click the **Samples** tab to view common where clauses.

Add a where clause from the Samples tab:

1. In the **Member Filter Builder** dialog box, click the **Samples** tab.
2. Expand the **Where Clause Expressions** list.

3. Double-click a sample to populate it in the **Member Filter** field.

## Cube View Performance

Cube view performance includes the following information:

- [Database Sparsity](#)
- [Row and Column Suppression](#)
- [Suppressed Members](#)
- [Sparse Row Suppression](#)
- [Cube View Paging](#)

### Database Sparsity

Sparsity is the ratio of data record volumes in the cube compared to the dimensions modeled in the cube. We see sparsity when the data unit (combination of cube, entity, parent, consolidation, scenario, and time) has sparsely populated data intersections across the account-type dimensions (account, intercompany, flow, and user-defined). The absence of data records can affect reporting performance because it is difficult to render reports if data is sparse.

Avoid sparsity in your application design when possible. Use analytic blend, extensibility, or other design frameworks to ensure the Dimension Library is designed for your reporting structure and to minimize sparsity. However, even with an optimal design, sparsity can still occur if a large report pulls in a lot of dimensions. Suppression settings can help improve reporting performance if this is the case.

### Row and Column Suppression

Use row and column suppression in larger reports to make them easier to read while still enabling the report builder to create a low maintenance report based on the metadata design.

To adjust the suppression settings:

1. On the **Cube Views** page, under **Cube View Groups**, select a cube view.
2. Select **Rows and Columns** to expand the slider.
3. Select the row or column and then click the **Data** tab. The following settings are available.

## Presenting Data With Books, Cube Views and Dashboards

---

Data	
Row	Row1
General	
Navigation Links	
Suppression	
Suppress Invalid Rows	False
Suppress NoData Rows	False
Suppress Zero Rows	False
Use Suppression Settings On Parent Rows	True
Zero Suppression Threshold	
Allow Insert Suppressed Member	(Use Default)
Column	Col1
General	
Navigation Links	
Suppression	
Suppress Invalid Columns	False
Suppress NoData Columns	False
Suppress Zero Columns	False
Use Suppression Settings On Parent Columns	True
Zero Suppression Threshold	
Use To Determine Row Suppression	True
Allow Sparse Row Suppression	True

- **Suppress Invalid Rows/Columns:** Set to True to suppress any invalid cells.
- **Suppress NoData Rows/Columns:** Set to True to suppress any cells without data. When data entry is required, the recommended setting is False.
- **Suppress Zero Rows/Columns:** Set to True to suppress any cells containing zeros.
- **Use Suppression Settings on Parent Rows/Columns:** This property relates to the prior properties and controls whether the parent members in this member filter use the same settings.

- **Zero Suppression Threshold:** When Suppress Zero Rows/Columns is set to True, enter a value to suppress all numbers below it. All numbers below the specified number are recognized as zeros. For example, entering 499.99 results in every number lower than that value being recognized as zero and therefore suppressed.
- **Allow Insert Suppressed Member** (rows only): Use this setting to access a member currently suppressed for data entry purposes. You can only use this with cube views and form templates.
  - **All:** Enables visibility to all cube view row expansions
  - **False:** All row expansions remain suppressed
  - **Nested:** Enables visibility of the row expansions two through four
  - **Innermost:** Enables visibility of the row expansion that is at the bottom level
- **User To Determine Row Suppression** (columns only): Set to True to improve performance on large cube views by enabling the designer to better define how to apply row suppression.
- **Allow Sparse Row Suppression** (columns only): Provides performance improvements for cube views that use multiple nested row dimensions and works in conjunction with a General Settings property. When set to True, sparse row suppression is applied to the entire cube view. It can be turned on and off for specific columns.

## Suppressed Members

This section includes instructions for how to manage suppressed members.

### Apply Modify Suppression Property

With this setting, you can choose whether rows are suppressed from the data explorer grid. This feature is useful for a small cube view that will not generate a large number of additional rows if the suppression is turned off.

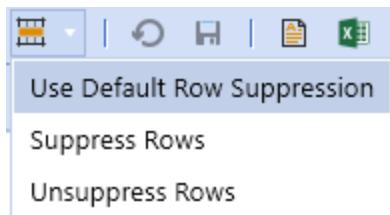
1. On the **Cube Views** page, under **Cube View Groups**, select a cube view.
2. Go to **General Settings > Common**.

3. Under **Restrictions**, set the **Can Modify Suppression** property to **True**.
4. Click **Save**.

## Modify Suppression

To modify suppression, you must first apply the correct property. See [Apply Modify Suppression Property](#).

1. Open a cube view in Data Explorer view.
2. Select the **Row Suppression** drop-down menu.



3. Choose the option to apply.
  - **Use Default Row Suppression:** Apply the cube view suppression settings.
  - **Suppress Rows:** Suppress any rows with zeros or no data regardless of the cube view settings.
  - **Unsuppress Rows:** Unsuppress all rows that were suppressed with zeros, no data, or invalid data regardless of the cube view settings.

## Apply Property to Allow Insert Suppressed Members

You can insert suppressed members so that their data is entered in the cube view.

1. On the **Cube Views** page, under **Cube View Groups**, select a cube view.
2. Go to **Rows and Columns > Data > Suppression** and confirm that **Allow Insert Suppressed Member** is set to **All**.

**NOTE:** The Allow Insert Suppressed Member option is only available for

rows.

3. Click **Save**.

## Insert Suppressed Members

To insert suppressed members, you must first apply the correct property. See [Apply Property to Allow Insert Suppressed Members](#).

1. Open a cube view in Data Explorer view.
2. Right-click in the account or entity.
3. Click **Insert Suppressed Member**.
4. To the right of the account or entity field, click **Select Member**.
5. Select an item and click the arrow to add it to the **Result List**.
6. Click **OK** and then click **OK** to confirm.
7. Enter data as needed in the white cells.
8. Click **Save** and then click **OK**.

## Sparse Row Suppression

Designs for analytic reports typically have multiple dimensions nested in rows. The combination of members generated from the nested expansions can easily result in billions of potential expanded rows, many of which may not have data. In these designs, standard row and column suppression (invalid, no data, or zero) in the cube view would still require each of those billion rows to be inspected individually for data. So, we recommend sparse row suppression be enabled to enhance the performance of a large cube view when, due to widespread database sparsity, the report is likely to return many records without data and take a long time to run. Sparse row suppression evaluates the data records of the cube view intersections and filters records with no data (not zeros) before rendering the cube view.

**IMPORTANT:** Sparse row suppression cannot be applied to dynamically calculated data through dynamically calculated members and cube view math. Avoid errors by correctly applying the settings on the dynamically calculated columns.

**NOTE:** When sparse row suppression is applied, OneStream assesses all row data before displaying the cube view. Cells containing dynamic calculations are populated on-the-fly. When OneStream assesses the rows of data, the rows containing dynamically calculated data are omitted from the cube view because the data is not saved in the database.

To enable sparse row suppression in cube views:

1. On the **Cube Views** page, under **Cube View Groups**, select a cube view.
2. Go to **General Settings > Common**.
3. Under **Suppression**, set **Allow Sparse Row Suppression** to **True**.
4. Select **Rows and Columns** to expand the slider. Select the row and then the **Data** tab.
5. Ensure the additional row suppression properties you will use are also set to **True**.

**NOTE:** Any row assigned a suppression setting will be enabled for sparse row suppression. If no suppression is applied, sparse row suppression will not be applied.

6. In the **Rows and Columns** slider, select the column and then the **Data** tab.
7. Under **Suppression**, set **Allow Sparse Row Suppression** to **True**.
8. Ensure the additional column suppression properties you will use are also set to **True**.

**NOTE:** If columns contain dynamically calculated data, set **Use to Determine Row Suppression** and **Allow Sparse Row Suppression** to **True** to avoid an error.

## Cube View Paging

Cube view paging is only applied to the data explorer view of a cube view and is used to enhance the performance of cube views containing more than 10,000 unsuppressed rows. The cube view will attempt to return up to 2000 unsuppressed rows within a maximum processing time of 20 seconds. The purpose of paging is to protect the server from large cube views that could affect application performance.

Evaluations are performed on the potential size of the rows and the processing requirements to determine if paging is enabled and the number of rows returned.

- Evaluation 1 – Enable Paging: An evaluation is performed on the entire cube view to determine the total number of possible unsuppressed rows that will be generated. If the total number of potential unsuppressed rows is less than 10,000, no paging will be enabled.
- Evaluation 2 – Paging Enabled: If the total number of unsuppressed rows is greater than or equal to 10,000, paging is enabled.
- Evaluation 3 – Paging: Once paging begins, the cube view evaluates the rows attempting to return a minimum of 20 to a maximum of 2000 unsuppressed rows. In the case of nested dimensions on rows, the evaluation starts on the left most dimension expansion, as defined in the cube view. After a maximum processing time of 20 seconds, the first page of the cube view will be returned for display containing only the rows that completed processing during the time constraint. For this reason, cube view pages are not a fixed number of rows. The rows are ultimately determined by their time to process. This also relates directly to the percentage display, because each page is generated by processing time requirements, and the last page is not known while the cube view is running. Therefore, this percentage is not intended to be a precise measurement.

When a row is defined with nested dimensions, the paging evaluation is performed on the left most dimension. For each expansion of the left most dimension, the paging will not progress to the next sibling until all the records are returned by all the other dimension expansions to be completed.

To set the properties, on the **Cube Views** page, under **Cube View Groups**, select a cube view. Go to **General Settings > Common**. The following settings are available.

- **Max Unsuppressed Rows Per Page:** Determines how many rows are written before the cube view starts paging (default is -1). The maximum value is 100,000.
- **Max Seconds To Process:** Determines how many seconds the cube view processes before it starts paging (default is -1). The maximum value is 600 seconds.

## Advanced Cube Views

There are several advanced uses with cube views such as setting them to change by Point of View or Workflow. Using advanced settings let you create and maintain fewer cube views.

### With Workflow

Set the POV, rows, and columns in the cube view so it is driven by the Workflow POV and the entities assigned to the workflow profile in use. By doing this, you can make forms, dashboards, cube views, and reports driven dynamically by the workflow profile.

### Workflow Entities

Using an expression such as E#Root.WFProfileEntities from within the Rows or Columns Tab shows the entity or entities assigned to that particular Workflow Profile at run time.

WFProfileEntities or similar expressions cannot be assigned to the POV because there can be more than one and the POV only requires a single member.

### Workflow Scenario

Under the cube view Point of View Slider, select the WF Member for the Scenario dimension, or use WFScenario|, or a similar Substitution Variable in Rows and Columns.

### Workflow Time

Under the cube view Point of View Slider, select the WF Member for the Time dimension, or use WFTime|, or a similar Substitution Variable in Rows and Columns.

### With Dashboard and Form Parameters

When using dashboards and forms, use parameters to focus the data to what is needed. For example, when a Dashboard is launched, it can prompt for the specific entity or region needed. See [Parameter Guides](#) for details.

Within the cube view, refer to parameters in the Point of View and Rows and Columns Slider to restrict the query to just the data expected. Surround the parameter name in pipes and exclamation points (for example, |!ParameterName!|).

The following image shows an example of using the entity and region parameters in the cube view's Point of View:



### Point Of View

Cube	GolfStream
Entity Member	!WorkflowEntityPrompt!
Parent Member	
Consolidation Member	Local
Scenario Member	WF
Time Member	WF
View Member	YTD
Account Member	
Flow Member	None
Origin Member	Top
IC Member	Top
UD1 Member	Top
UD2 Member	Top
UD3 Member	!Region Parameter (Workflow)!
UD4 Member	Top

**TIP:** Use dashboard parameters as a single repository for parameters that are used in dashboards or forms. If a parameter is referred to within a cube view (for example, `!ParameterName!`) and there is not a parameter by that name associated with the form, it will search through the application's dashboard parameters for one with that name and use it.

## Changing Member Display Name

Change the name returned with a member script in a cube view by adding `:Name("enter name here")` at the end of the member filter. The double quotes around the `Name()` value are optional.

For example

`A#60999:Name("Net Sales")` or `A#60999:Name(Net Sales)`

## Format a Cube View

This section includes the following information to format a cube view:

- [Overview](#)
- [Basic Formatting](#)
- [Rename Rows and Columns](#)
- [Create Conditional Formatting](#)
- [General Settings](#)
- [Format a Report Page](#)

## Overview

Formatting settings and overrides combine to create the cell format. You can apply formatting and isolate it to nested expansion levels on rows or columns by using property filters found in the conditional formatting dialog box.

Apply property filters in this order:

1. Application properties standard report settings
2. Cube view default settings
3. Column settings
4. Row settings
5. Column row overrides
6. Row column overrides

The primary output for a cube view is the data explorer grid. You can also export to Excel or use a report viewer.

## Basic Formatting

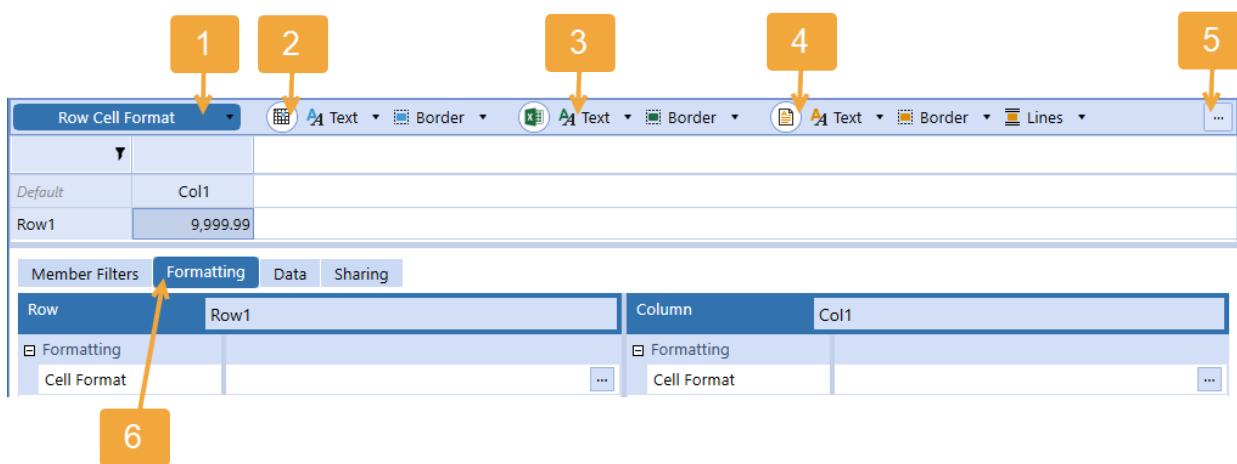
Cube view formatting controls how cube views display. Most of the format settings transfer to Excel and can still be overridden with the standard Excel styles. The formatting also transfers to reports where the cube view is being used in a data adapter.

## Presenting Data With Books, Cube Views and Dashboards

---

Cube view formatting can be selected for the entire cube view or specific rows, columns, headers, and individual cells. The formatting options enable the number formats, percentage signs, scaling, currency symbols, colors, fonts, and font size to be unique to the business needs. Label row and column sets to support cube view maintenance when editing, formatting, and applying calculations.

In the cube view, select the **Rows and Columns** slider. Then click an individual cell or header and then the **Formatting** tab to view and edit the toolbar options for data explorer, Excel, and report formats. The icons in the toolbar include the most common formatting properties. Each icon coincides with the cell selected in the Rows and Columns slider. For example, if a column header is selected, the formatting changes made will only affect the specific column header.



1. Selector
2. Data explorer formatting options
3. Excel export formatting options
4. Report formatting options
5. Advanced formatting options
6. Formatting tab

### Set Up Basic Formatting for a Cube View

1. On the **Cube Views** page, under **Cube View Groups**, select a cube view.
  2. Select **Rows and Columns** to expand the slider.
  3. Select the row, column, or cell to edit.
- NOTE:** Click the **Default** cell to update the default settings for all headers and cells.
4. Click the **Formatting** tab.
  5. Select the options to change the header and cell format settings for the data explorer, Excel, and report output types.

#### Data Explorer

The formatting settings available are organized in the following groups.

- **Text:** Font, color, size, bold, and italic settings. When you select a cell, the following formatting properties become available in the Text drop-down menu: number, zero offset, scale, flip sign, show percent sign, and show currency.
- **Border:** Background color and gridline color
- **Column:** Whether the column is visible and its width

There are different settings available for the headers and cells.

Cube View Header Format		Text	Border	Column	Text	Border	Column	Text	Border	Lines	Column
Default	Col1										
Row1	9,999.99										

Row Cell Format		Text	Border	Text	Border	Text	Border	Text	Border	Lines	
Default	Col1										
Row1	9,999.99										

#### Excel

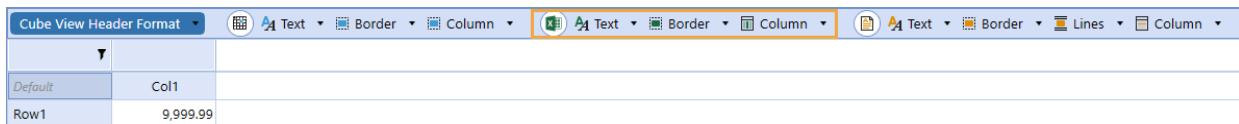
The formatting settings available are organized in the following groups.

## Presenting Data With Books, Cube Views and Dashboards

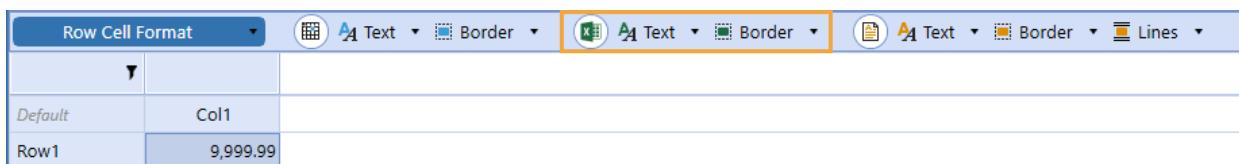
---

- **Text:** Color, horizontal alignment, vertical alignment, indent level, and wrap settings. When you select a cell, the following formatting properties become available in the Text drop-down menu: number and use scale.
- **Border:** Background color and the color and line styles of the cell borders
- **Column:** Column width

There are different settings available for the headers and cells.



Cube View Header Format		A1 Text	Border	Column	A1 Text	Border	Column	A1 Text	Border	Lines	Column
Default		Col1									
Row1		9,999.99									



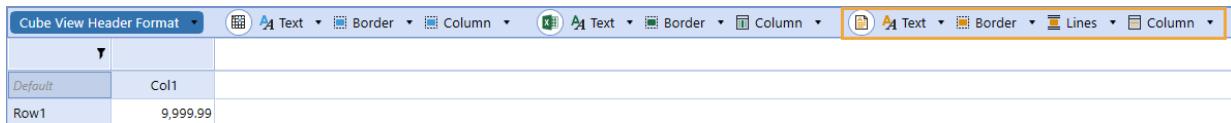
Row Cell Format		A1 Text	Border	A1 Text	Border	A1 Text	Border	Lines	
Default		Col1							
Row1		9,999.99							

### Report

The formatting settings available are organized in the following groups.

- **Text:** Color, alignment, size, and underline settings. When you select a cell, the following formatting properties become available in the Text drop-down menu: no data number, use numeric binding, and numeric binding.
- **Border:** Background color and cell borders
- **Lines:** Settings for the top and bottom lines for the first row, padding, color, and thickness
- **Column:** Column width

There are different settings available for the headers and cells.



Cube View Header Format		A1 Text	Border	Column	A1 Text	Border	Column	A1 Text	Border	Lines	Column
Default		Col1									
Row1		9,999.99									

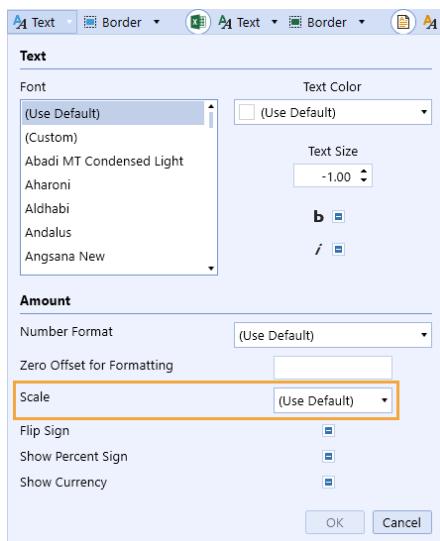
Row Cell Format	

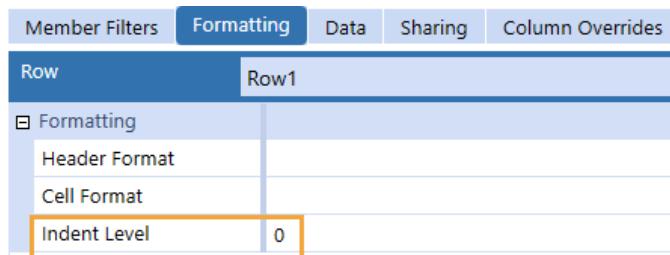
Default	Col1
Row1	9,999.99

## Apply Scaling, Indents, and Currencies

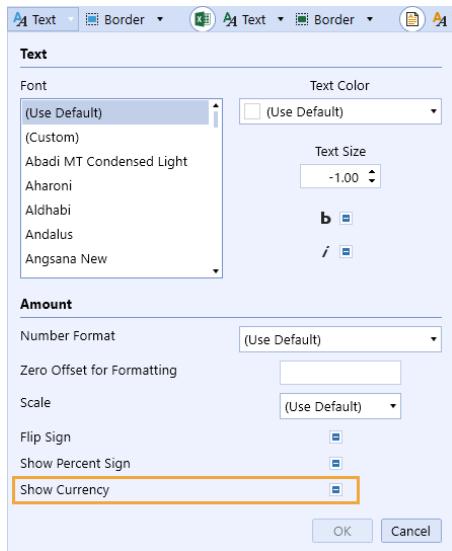
- **Scale:** Open the **Data Explorer Text** drop-down menu to view the **Scale** setting. You can also apply it to Excel and report formatting.



- **Indent level:** On the **Formatting** tab for a row or column header, view the **Indent Level** setting.



- **Show currency:** Open the **Data Explorer Text** drop-down menu to view the **Show Currency** setting.



## Header Format Details

The formatting properties for a cube view header include default, row, and column headers. These properties are in the Rows and Columns slider. Not all properties are available for each header.

1. In the **Rows and Columns** slider, select the **Default** cell to apply the formatting settings to all of the cube view headers.

**NOTE:** You can select an individual row or column header instead if you want to edit the settings for a specific header.
2. In the **Formatting** tab, click the ellipsis to the right of **Header Format**.
3. Click **Format** to view the options. The Excel section only affects how the cube view data display in Excel.
  - **Custom Parameters:** Click the ellipsis to select and assign a custom parameter such as a cube view style to the cube view header.

- **RowExpansionMode:** Control expansion of nested rows. This determines how rows are displayed in the data explorer grid. The default sets this property to False.
- **ShowDimensionImages:** Set to False to hide the dimension icons in the data explorer grid row and column headers. The default setting will display the dimension icons.
- **IsColumnVisible:** Set to False to hide specific columns. Set this property at run time with a parameter to show and hide detail.
- **ColumnWidth:** Enter a numerical value for column width. If the header label exceeds the column width, the header will automatically wrap the text.
- **IsRowVisible:** Set to False to hide specific rows. Set this property at run time with a parameter to show and hide detail.
- **TreeExpansionLevel1-4:** Set the number of rows from 0 to 4 to reveal the amount of nested data available in the cube view. A value set to zero will collapse all data. A value set to greater than the expansion level will expand all. If no value is set for an expansion level, it defaults to the normal RowExpansionMode behavior.

**NOTE:** For TreeExpansionLevel to work, RowExpansionMode from the Default Header must be set to "Use Default" and contain a .Tree Member Expansion. Using Collapse All or Expand All overrides this setting.

- Font and color options are available. For the FontFamily, any .Net font installed on the Excel client operating system is included in the font family. The FontSize default is 11.
- **ExcelMaxOutlineLevelOnRows** and **ExcelMaxOutlineLevelOnCols:** Up to six outline levels can be used for the creation of collapsible and expandable Excel groups of rows or columns when exporting. Enter the number of outline levels needed. This relates to the Excel Indent Level setting that is applied in the header format on each row or column, so the number entered here should equal the highest setting applied on the rows or columns.
- **ExcelExpandedOutlineLevelOnRows** and **ExcelExpandedOutlineLevelOnCols:** When a cube view is exported to Excel, these settings control to which outline level the file is initially opened. The default is 1, which means each grouping is fully collapsed.
- **ExcelColumnWidth:** Default Excel column width in pixels unless overridden in columns.
- **ExcelOutlineLevelCol** and **ExcelOutlineLevel:** Default Excel outline level for rows and columns is 1 unless specified here or overridden in rows or columns.

- **ExcelHorizontalAlignment** and **ExcelVerticalAlignment**: Control how the data will be aligned horizontally and vertically using the Excel standard alignment options.
- **ExcelIndentLevel**: The number of characters to indent the text or value.
- **ExcelWrapText**: This determines if text like headers will be wrapped in the output.
- Color and border options are available.
- **ReportSeparateBandPerCubeViewRow**: This is a setting to optimize performance for large cube views that will be dynamically generating a data explorer report with many row definitions with unique formatting. Default setting is (Use Default), which is a conditional setting. This applies to when a data explorer report is run from a cube view and background temporary database tables in memory, one for each row name in the cube view are created. If there are more than 100 row definitions, fewer report bands (separate temporary in-memory tables) are created, combining rows where formatting is the same. Set to True to create a separate table per cube view row definition, which may affect performance with hundreds or thousands of row definitions. Set to False to combine row definitions of the same format into fewer temporary in-memory tables.
- Page and column header and table layout options are available.
- **ReportColumnWidth** and **ReportRowHeight**: Sets the width of a column or the height of a row by number of pixels.
- **ReportTopLinesOnFirstRowOnly** and **ReportBottomLinesOnLastRowOnly**: If the row definition results in multiple rows, a top line will be placed on the first line only or a bottom line will be placed on the last line only.
- **ReportBandHeightBeforeFirstRow** and **ReportBandHeightAfterLastRow**: Allows a number to be entered in pixels to add before the first row or after the last row of where a row definition starts.

**NOTE:** The top and bottom report bands used for lines and spacing are removed if there are no data rows or for suppressed rows.

- **ReportRowPageBreak**: Applies a page break where appropriate for this row.
- **ReportRowContentTop**: Specifies the vertical position of the row relative to the row above it. If it is set to zero, the top of the row immediately follows the row above it with no vertical spacing.
- **ReportRowContentHeight**: The height of the row in pixels.

- **ReportRowPaddingTop** and **ReportRowPaddingBottom**: Control the extra space (padding) above and below a row in a report.
- **ReportFontSize**: The point size of the font on the report if it differs from the font size displayed in data explorer.
- **ReportTextAlignment**: Controls how the data are aligned in a report.
- **ReportUnderline**: Creates a simple underline of report values.
- Color options are available for text and background.
- **Top Lines** and **Bottom Lines**: Used for underlines and overlines. The Line1 and Line2 options can be used together to create a double underline or double overline. For the lines to have a small gap between each column, use the ReportBottomLine1PaddingLeft property or one similar.
- **Borders**: Allows a line to be drawn around the border of a cell.

## Cell Format Details

The format properties for a cell include font, colors, background, and grid lines.

1. In the **Rows and Columns** slider, select the **Default** cell to apply the formatting settings to all of the cube view cells.

**NOTE:** You can select an individual cell instead if you want to edit the settings for a specific cell.
2. In the **Formatting** tab, click the ellipsis to the right of **Cell Format**.
3. Click **Format** to view the options. The Excel section only affects how the cube view data display in Excel.
  - **Custom Parameters**: Click the ellipsis to select and assign a custom parameter such as a cube view style to the cube view cell.
  - **NumberFormat**: Uses the Microsoft .NET standard number format syntax, which allows different formats to be specified for positive and negative numbers separated by a semi-colon (not available in Excel). For example:

- #,### ;(#,##);0 would show the number using a comma as the thousands separator, no degrees of precision, parenthesis around negative numbers, and a zero for null values. The pattern of this format choice is Positive;Negative;Null values with each separated by a semicolon. After the first #,### there is a space. This enables the numbers to line up with negative numbers due to the parenthesis.
  - #,###.% ;(#,##.%);” – “ would show a percentage with a comma as the thousands separator, one degree of precision, negative percentages in parenthesis and a dash for null values.
  - N2 would show the data as a numeric value with two degrees of precision. Negative numbers are presented with a minus sign.
  - P1 would show a percentage with one degree of precision.
- **ZeroOffsetForFormatting:** Related to NegativeTextColor, which is determined by whether a number is less than zero. For example, if sales were less than 100, rather than 0, they could be displayed in red. A valid setting is any number other than zero.
  - **Scale:** -12 to +12 are the valid values for the scale. For example, to show a number in thousands, the scale should equal 3, or for millions, the scale should equal 6.
  - **FlipSign:** This will flip the display value between positive and negative. This is useful for reports where certain expense numbers are stored as positive or negative and need to be shown on the report.
  - **ShowPercentageSign:** Determines whether a percentage sign is displayed.
  - **ShowCurrency:** Shows the currency code (for example, EUR). This is not available in Excel.
  - Font and color options are available. For the **FontFamily**, any .Net font installed on the Excel client operating system is included in the font family. The **FontSize** default is 11. **NegativeTextColor**, **WritableBackgroundColor**, and **SelectedGridLinesColor** can optionally override the format for negative numbers, writeable data cells, and selected data cells.
  - **ExcelNumberFormat:** Apply to control how numbers display when exporting to Excel. A number format can have up to four sections of code separated by semicolons. These code sections define the format for positive numbers, negative numbers, zero values, and text, in that order.

<POSITIVE>;<NEGATIVE>;<ZERO>;<TEXT>

For example, use these code sections to create the following custom format. Note that this format can include the underscore to create a space after the trailing positive number and can control the color of the negative number to be red:

#,##0.00 );[Red](#,##0.00);0.00

**NOTE:** ExcelNumberFormat settings [<#,##0,,0 ;[Red](#,##0,,0)] on a cube view scales numbers when the cube view is exported to Excel.

- **ExcelUseScale**: Determines if Excel uses the scale property.
  - Alignment, color, and border options are available.
  - **ReportNoDataNumberFormat**: By default, NoData (null) cells display as empty in the data explorer and as zeros in a report. However, any .NET number format text can be specified to format those zero values differently in the report. For example, type NODATA in that setting. To display empty text, type # in that setting. If the number format setting of the cell already does something with null values, this property does not need to be filled.
  - **ReportUseNumericBinding** and **ReportNumericBindingFormat**: These are related to the export of dashboard reports to Excel, so numbers can be rendered in the proper number format rather than appear in Excel as text. Set ReportUseNumericBinding to True to use numeric amounts instead of OneStream text-based formatting when generating a cube view report. The ReportNumericBindingFormat setting must follow a specific syntax. This number format is related to the report engine, so Excel standard number formats are not used. For example, to represent a number format for positive, negative, and no values:

{0:#,# ;(#,#);" - "}

This feature provides the ability to generate numbers instead of text when exporting a dashboard report to Excel. However, it cannot be used for calculation status and annotation data cells because those features cannot be represented as numbers. In those cases, use column overrides to display the values as text.

- Position, font, color, top lines, bottom lines, and borders are available. See [Header Format Details](#).

## Member Filters

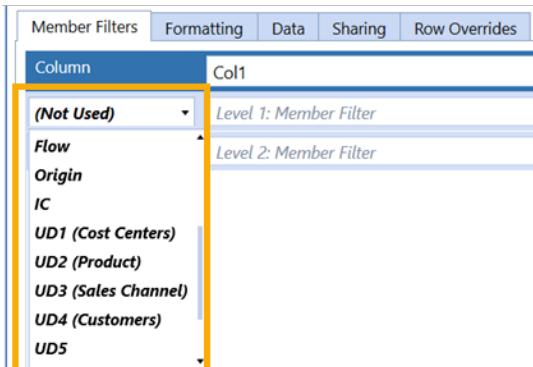
In a cube view, you can nest up to four different dimensions as rows and two different dimensions as columns.

**NOTE:** When nesting cube view columns, if the top column header is longer, it will span the nested column headers when displayed on a cube view report.

**Dimension Type:** Select a dimension type from the drop-down list. Indicates which dimension type will display on the selected row or column.

**Member Filter:** Enter a member filter to express the specific members needed from the dimension.

**User Defined Description – Cube View:** In the Cube View Editor, the custom user-defined descriptions will display in the Designer and Advanced tabs, in the Member Filters dimension type:



In the Advanced tab, Row and Column dimension type:

## Presenting Data With Books, Cube Views and Dashboards

---

Row Members	
Member Expansion 1	
Primary Dimension Type	UD1 (Cost Centers)
Member Filter	IC
Indent Level	UD1 (Cost Centers)
Nested Member Expansion 2	UD2 (Product)
Primary Dimension Type	UD3 (Sales Channel)
Member Filter	UD4 (Customers)
Nested Member Expansion 3	UD5
Primary Dimension Type	UD6
Member Filter	UD7
Nested Member Expansion 4	
Primary Dimension Type	(Not Used)

### Data

**Can Modify Data:** If False, specific rows or columns are read-only. If True, the False setting will override it. However, if Can Modify Data is False, the entire cube view is read-only and this setting will not override if True.

**Text Box:** Makes the cube view cell numerical. Only numerical values can be entered in the data cell at run time. This is the default setting.

**Combo Box:** Select this when a List Parameter is used in a cube view data cell. Enter the name of the List Parameter in the property below.

**Date:** Enables a calendar in the cube view cell and allows you to select a specific date in the data cell.

## Presenting Data With Books, Cube Views and Dashboards

---

	Jan 2011	Feb 2011	Mar 201				
Total Inventories	3,714.00	1,500.00	0.00				
Raw Materials Inventory	700.00	700.00	0.00				
Work in Progress Inventory	800.00	800.00	0.00				
Finished Goods Inventory	680.00	0.00	0.00				
Supplies - Inventory	965.00	0.00	0.00				
In Transit Inventory	569.00	0.00	0.00				
R&D Materials	1,508.00	-1,508.00	0.00				
R&D Services	23,651.00	-23,651.00	0.00				
Outside Engineering Services	12,658.00	-12,658.00	0.00				
Federal Income Tax Provision	5,000.00	0.00	-5,000.00				
State/Local Income Tax Provision	4,300.00	-4,300.00	0.00				
DateAcct	9/6/2017	8/27/2017	1/1/1900				
◀ September - 2017 ▶							
	Sun	Mon	Tue	Wed	Thu	Fri	Sat
35	27	28	29	30	31	1	2
36	3	4	5	6	7	8	9
37	10	11	12	13	14	15	16
38	17	18	19	20	21	22	23
39	24	25	26	27	28	29	30
40	1	2	3	4	5	6	7
Now				Clear			

**Date Time:** Enables a calendar and a time display and allows you to select a specific date and time in a data cell.

## Presenting Data With Books, Cube Views and Dashboards

---

	⌚ Jan 2011
⌚ Total Inventories	3,714.00
⌚ Raw Materials Inventory	700.00
⌚ Work in Progress Inventory	800.00
⌚ Finished Goods Inventory	680.00
⌚ Supplies - Inventory	965.00
⌚ In Transit Inventory	569.00
⌚ R&D Materials	1,508.00
⌚ R&D Services	23,651.00
⌚ Outside Engineering Services	12,658.00
⌚ Federal Income Tax Provision	5,000.00
⌚ State/Local Income Tax Provision	4,300.00
⌚ TimeAcct	9/5/2017 1:52 PM ⚡

**NOTE:** Selecting Now in the calendar enters the current date or current date and time.

**List Parameter:** Enter a parameter to create a drop-down list of choices in a cube view data cell.

Report Status	
	⌚ Feb 2011
⌚ Report Status	
⌚ Montreal	Not Started
⌚ Quebec City	Under Review
⌚ Augusta	Completed
⌚ Carlsbad	Under Review
⌚ Houston Heights	No Data ▾
⌚ South Houston	No Data
⌚ Frankfurt	Not Started Under Review Completed

**Enable Report Navigation Link (Rows Only):** Set to True to enable Navigation Link Drill Downs in dashboards. Enter a value in the Dashboard to Open in Dialog field. Set to False if this function is not used for the cube view. See "Navigation Links" on page 618.

**Dashboard to Open in Dialog (Rows Only):** Name of the dashboard or parameter used for the Report Navigation Link. Use a parameter for a cube view acting as a row template across multiple cube views.

**Linked Cube Views:** Enter a comma-separated list of cube views that will be available to open when viewing a cube view in the Data Explorer Grid. The cube views specified in this field apply to the selected row or column and are available when you right-click on any data cell in the specified row/column.

**NOTE:** Row settings override column settings which override cube view settings.

## Rename Rows and Columns

By default, a cube view has standard headers to represent rows and columns. All the header settings control the row and column headers presented when the cube view is run if the default is not used. See [General Settings](#).

If the header name for a row or column must be different than the dimension name or description, use the rename function. The rename function is an expansion of the end of a member filter that enables you to enter what you want displayed. For example, A#CashBalance:Name(Cash Balance) will display Cash Balance as its header.

Another way to use the rename function is to include the XFMemberProperty, which returns member properties. For example:

A#6009.base:Name(XFMemberProperty(DimType=Account, Member=|MFAccount|, Property=AccountType))

1. Use the rename function to pull in the member property: **Name(XFMemberProperty**
2. Define the dimension type: **(DimType=Account,**
3. Determine what will be in the member filter: **Member=|MFAccount|,**
4. Pull in the property: **Property=AccountType))**

## Create Conditional Formatting

Conditional formatting provides the ability to format headers or cells based on defined criteria. Conditional formatting in cube views supports data analysis by highlighting cells or ranges of cells, identifying key values, and using data bars, color scales, and icon sets that correspond to specific variations in the data. Conditional formatting is applied and visible in data explorer, Excel, and report views. The conditional formatting criteria are applied to cube view rows or columns on the formatting property within the cube view editor.

Conditional formatting is available on the formatting elements of a cube view.

- Cube view default formatting
- Row and column headers
- Row and column data cells
- Row and column overrides

Conditional formatting follows the cube view processing order of operations:

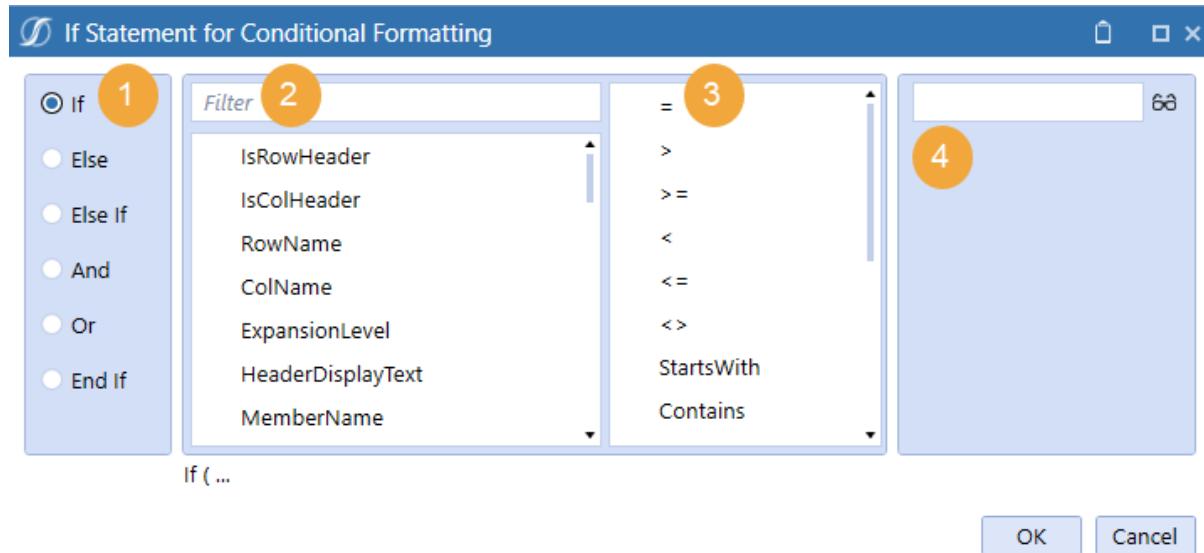
1. Number format as defined in application properties
2. Cube view default
3. Column formatting
4. Row formatting
5. Column overrides
6. Row overrides

The combination of formats and overrides equals the format for the cell when rendered. Formatting can be applied and isolated to the nested expansion levels on rows and columns by using property filters in the dialog box. This feature is a cube view formatting option.

**NOTE:** If there are any existing formats before applying conditional formatting, they will be retained if the range of cells containing the conditional formats does not meet the conditions of the rule. All styles from the cube view and the selection styles that had been applied to that range are overridden by conditional formatting.

### Conditional Formatting Properties

General and conditional formatting can be applied in the Header Format and Cell Format dialog boxes. The Condition and Format buttons launch additional dialog boxes with available options. The dialog box for conditional formatting is divided into sections.



1. **Conditional statement:** Defines the options for conditional statements to apply constraints to formatting.
2. **Filter:** Applies the criteria that pertain to your statement.
3. **Operator:** Contains criteria filters to use in your conditional statement.
4. **Text/object:** Find the text or object to be tested.

### Header Property Filters

Header Property Filter	Description
IsRowHeader	Boolean, determines if an object is a row header field (appropriate for Cube View Default).

Header Property Filter	Description
IsColHeader	Boolean, determines if an object is a column header field (appropriate for Cube View Default).
RowName	Cube view row name
ColName	Cube view column name
ExpansionLevel	Determines the expansion level for rows (1-4) and columns (1-2). Related to isolating the row expansion headers.
HeaderDisplayText	Custom descriptions used with the :Name() function
MemberName	Metadata member labels
MemberDescription	Metadata member descriptions
MemberShortDescription	Metadata member short descriptions (only Time dimension)
IndentLevel	Indentation level derived as a formatting setting or that is system-generated based on tree expansions

## Expansion Specific Property Filters

Expansion Specific Property Filter	Option	Description
RowE expansion level - criteria	Row expansions 1–4	Identify nested or expansion rows with specified criteria
ColE expansion level - criteria	Column expansion 1 and 2	Identify nested or expansion columns with specified criteria

### Cell Format Property Filters

Cell Format Property Filter	Description
IsNoData	Test for no data
IsRealData	Test for stored data, ignoring derived Zero-View data
IsDerivedData	Test for derived data, commonly resulting from Scenario Zero-View settings
IsRowNumberEven	Test for the row number as an expansion or on a fixed row
ExpandedRowNum	Test the count of expanded rows, Zero-based. This property filter is based on the total cube view count of rows generated from each row and its expansions.
CellAmount	Test cell data amount
CellStorageType	Test the method used to store data

### Define Properties for Conditional Formatting

Conditional formatting can be applied based on a data point. Cell formatting conditions can occur based on member names or descriptions, indent levels, row and column names, and expanded and even numbered rows.

Conditional formatting can also be applied based on members. The ability to format based on the name or description is enhanced by using a standardized metadata naming convention. Summary level members with keywords such as “Total” or prefixes or suffixes such as “Tot” could be used in conditional formatting. Members can be formatted using dynamic criteria, such as StartsWith or EndsWith. HeaderDisplayText differs from the MemberName and MemberDescription property filter because it references the custom name parameter in a member filter. Dynamic criteria can be applied to the name and description property filters to apply the required formatting.

1. On the **Cube Views** page, under **Cube View Groups**, select a cube view.
2. Select **Rows and Columns** to expand the slider.
3. Select a row or column.
4. Click the **Formatting** tab.
5. Click the ellipsis to launch the dialog box.
6. Click **Condition**, set the requirements, and then click **OK**.
7. Click **Format**, set the requirements, and then click **OK**.
8. Click **Condition** and then **End If** to close the condition.
9. Click **OK** twice.

## Examples Applying Conditional Formatting

Below are common usage examples of conditional formatting.

### Indent Level

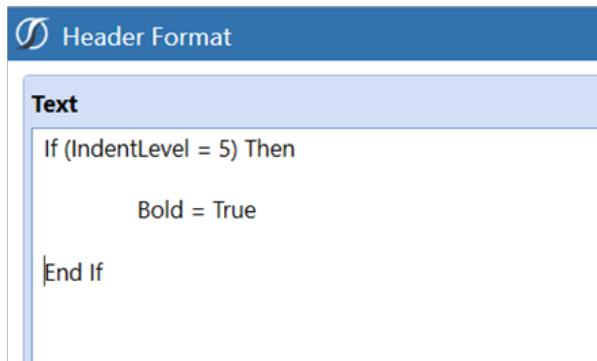
The IndentLevel Property filter will dynamically format from defined rows or expansions. Indentation is zero-based. The formatting can be applied to the default or to rows. This solution can speed formatting for summary level dimension members.

## Presenting Data With Books, Cube Views and Dashboards

---

		Jan 2018
[-]  Net Income		44,530,432.42
[-]  Earnings Before Taxes		48,038,554.85
[-]  Earnings Before Interest and Taxes		48,987,212.85
[-]  Total Operating Income		50,832,352.09
[-]  Gross Income		50,945,236.19
[-] <b>Net Sales</b>		50,945,236.19
[-] <b>Cost of Goods Sold</b>		
[-]  Total Operating Expenses		112,884.10
[-] <b>Total Operating Exp Before Allocation</b>		112,884.10
[-] <b>Total Allocations</b>		
[-]  Total Allocations In		
[-]  Allocations Out		

Apply the formatting to the cube view's Header section or to specific rows.



### Examples of Using Conditional Property Filters Conditional “Traffic-Lighting”

“Traffic-Lighting” is data related and therefore applied as a cell format. The designer has a choice to apply the conditional formatting to either a row or a column. The order of operations for formatting can impact the decision. Row overrides are the final layer of formatting applied to a cube view and would not be impacted by other more general formatting.

The screenshot shows a Data Explorer interface with a title bar "Data Explorer - CellAmountTrafficLightAsOverride". Below the title bar is a toolbar with various icons. The main area displays a data grid with four columns: Actual, Plan, BW Diff, and Var %. The "Actual" column has a dropdown menu open, showing "Jan 2018" selected. The "Plan" column also has a dropdown menu open, showing "Jan 2018" selected. The "BW Diff" and "Var %" columns have dropdown menus open, both showing "Jan 2018" selected. The data grid contains five rows of sales data:

	Actual	Plan	BW Diff	Var %
60000 - Operating Sales	57,982,341.55	60,275,344.46	-2,293,002.91	-3.80
2000_100 - Third Party Sales	50,945,236.19	54,511,402.72	-3,566,166.53	-6.54
2000_200 - OEM Sales	4,606,156.98	4,145,541.28	460,615.70	11.11
2000_300 - Subassembly Sales	382,840.04	287,130.03	95,710.01	33.33
2000_400 - Parts Sales	2,048,108.34	1,331,270.42	716,837.92	53.85

1. Conditional formatting can be applied to the rows or columns. A definition applied to the row Formatting tab would apply to all columns. A row override would isolate the formatting to a specified column(s).
2. The CellAmount filter is used within multiple If/ElseIf statements to define the various tests required for the report.

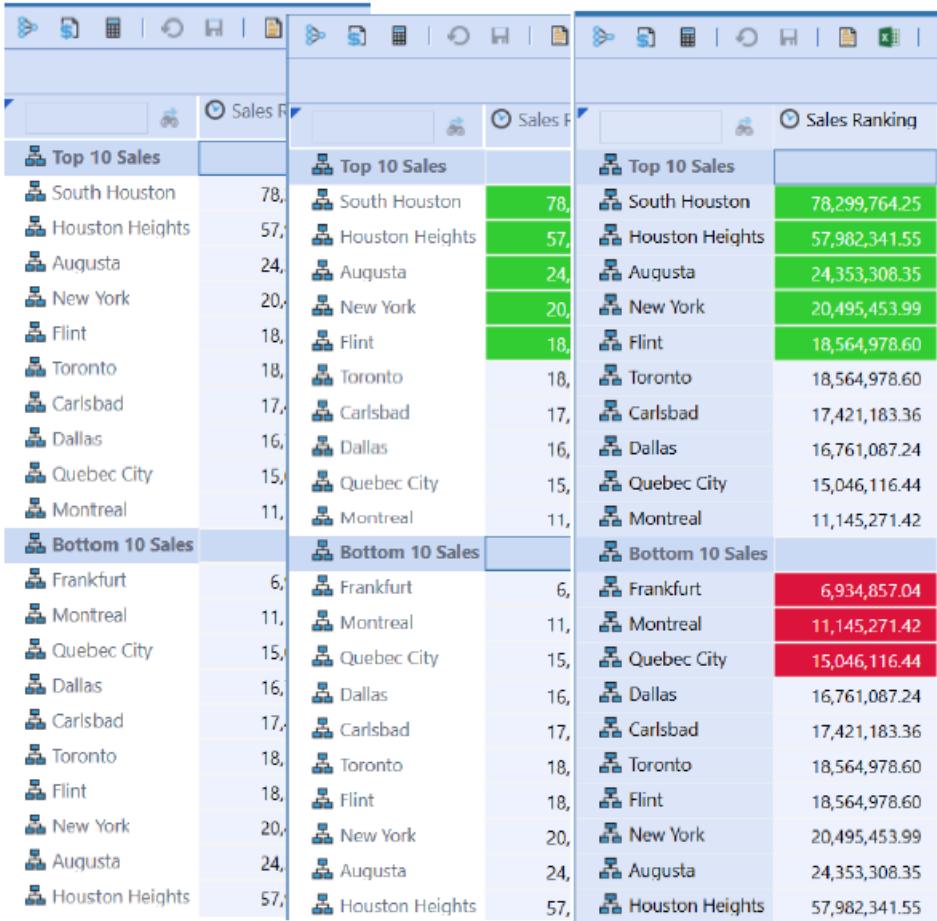
### ExpandedRowNum Expansion Range

Conditional formatting provides the ability to format an expanded range of data cells using the ExpandedRowNum Property Filter. This is useful in formatting to support “Top 10” type reports. This filter best supports cube view designs that use known expansions, such as ranking business rules or member lists. If applied to specific rows, the formatting applies to the defined row, but the Expansion Number Reference relates to the entire cube view. Formatting defined on subsequent rows will be impacted if the expansion members change on previous rows. The ExpandedRowNum Property

Filter can also be applied as a default cube view format, in which the definition will apply to all rows. This would require If/ElseIf type statements to support all rows.

## Presenting Data With Books, Cube Views and Dashboards

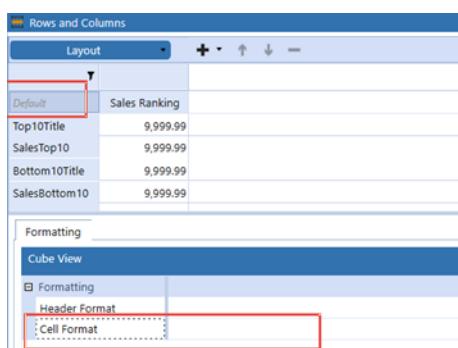
---



The figure consists of three side-by-side screenshots of a report interface. Each screenshot shows a table with a header 'Sales Ranking' and two sections: 'Top 10 Sales' and 'Bottom 10 Sales'. The cities listed in the table are South Houston, Houston Heights, Augusta, New York, Flint, Toronto, Carlsbad, Dallas, Quebec City, and Montreal. The 'Top 10 Sales' section lists cities from 78 down to 11. The 'Bottom 10 Sales' section lists cities from 6 up to 57. The data is presented in a grid format with some cells highlighted in green (Top 10) and red (Bottom 10).

Sales Ranking	
Top 10 Sales	
South Houston	78,
Houston Heights	57,
Augusta	24,
New York	20,
Flint	18,
Toronto	18,
Carlsbad	17,
Dallas	16,
Quebec City	15,
Montreal	11,
Bottom 10 Sales	
Frankfurt	6,
Montreal	11,
Quebec City	15,
Dallas	16,
Carlsbad	17,
Toronto	18,
Flint	18,
New York	20,
Augusta	24,
Houston Heights	57,

1. Design or open a report which supports formatting for ExpandedRowNum , such as a ranking report in the example.
2. Determine how the formatting should be applied, as a default, row or column. The example will use the cube view default formatting.



## Presenting Data With Books, Cube Views and Dashboards

---

3. Cell format is applied for conditional formatting using the ExpandedRowNum filter. Being zero-based, and having to account for each row and its expansions:

- a. Condition1 – Rows < 6 is defined because the text header row “Top10Title” initiates the count at zero.
- b. Condition2 – Rows > 11 and < 15 is defined because all the rows up to Row4, “SalesBottom10”, reflect rows 0-11 on the cube view. The conditional row references must reflect all the cube view expansions.

### Test for Row IsRowNumberEven

The IsRowNumberEven property filter may be useful to vary formatting by the even/odd numeric expansion of rows on a cube view. For example, to replicate a “green-bar” style report the IsRowNumberEven would be a suitable filter.

	Jan 2018	Feb 2018	Mar 2018	Apr 2018	May 2018
Net Income	44,530,432.42	0.00	0.00	0.00	0.00
Earnings Before Taxes	48,038,554.85	0.00	0.00	0.00	0.00
Earnings Before Interest and Taxes	48,987,212.85	0.00	0.00	0.00	0.00
Total Operating Income	50,832,352.09	0.00	0.00	0.00	0.00
Gross Income	50,945,236.19	0.00	0.00	0.00	0.00
Net Sales	50,945,236.19	0.00	0.00	0.00	0.00
Operating Sales	50,945,236.19	0.00	0.00	0.00	0.00
Third Party Sales	50,945,236.19	0.00	0.00	0.00	0.00
Total Operating Expenses	112,884.10	0.00	0.00	0.00	0.00

In this example the formatting can be applied to the cube view default cell format since it will globally apply to all rows.

Rows and Columns

Layout

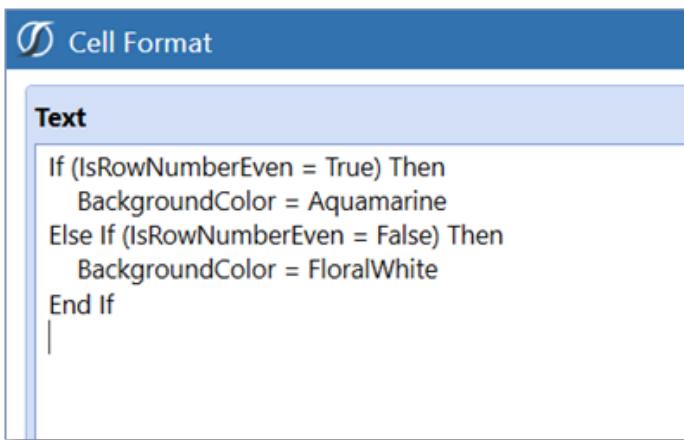
Default	Col1
Row1	9,999.99
Row2	9,999.99

Formatting

Cube View

Header Format

```
If (IsRowNumberEven = True) Then BackgroundColor = Aquamarine Else Backg...
```



### MemberName, MemberDescription or HeaderDisplayText

The ability to format based on the Name or Description is greatly enhanced by adhering to a standardized metadata naming convention. Summary level members having keywords such as "Total" or prefixes or suffixes such as "Tot" could be used in conditional formatting. Members could be formatted using dynamic criteria, such as StartsWith or EndsWith.

HeaderDisplayText differs from the MemberName and MemberDescription Property filter in that it references the custom Name parameter in a Member Filter.

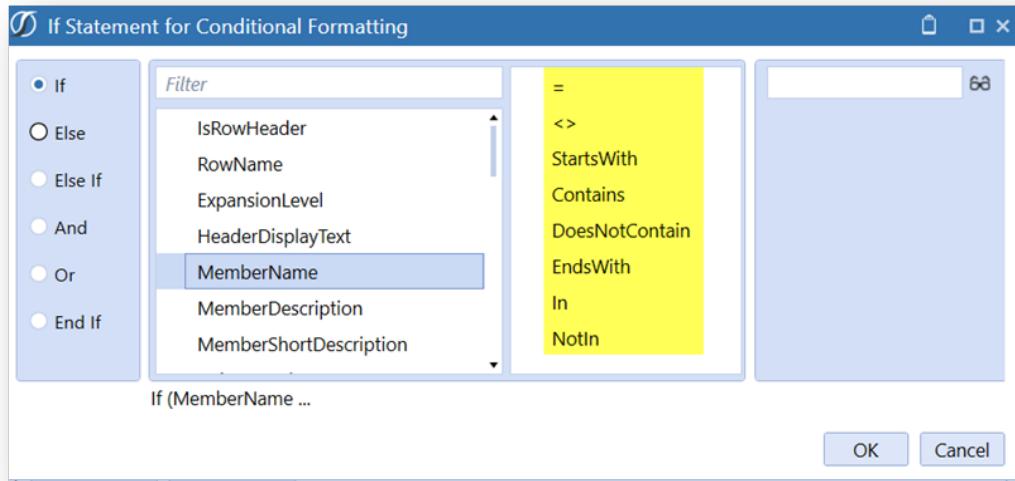
The screenshot shows the 'Member Filters' dialog box with the following structure:

- Row: Row1
- Account: A#69000:Name('Pretax Income')
- (Not Used): Level 2: Member Filter
- (Not Used): Level 3: Member Filter

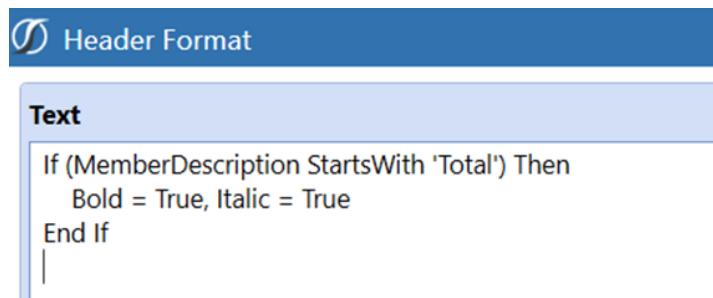
Dynamic criteria can be applied to the Name and Description property filters to apply the required formatting.

## Presenting Data With Books, Cube Views and Dashboards

---



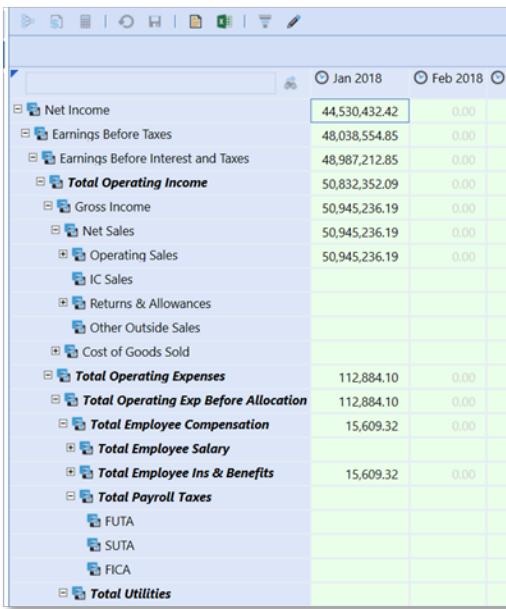
Apply conditional formatting to row header.



The result is a dynamically formatted report.

## Presenting Data With Books, Cube Views and Dashboards

---



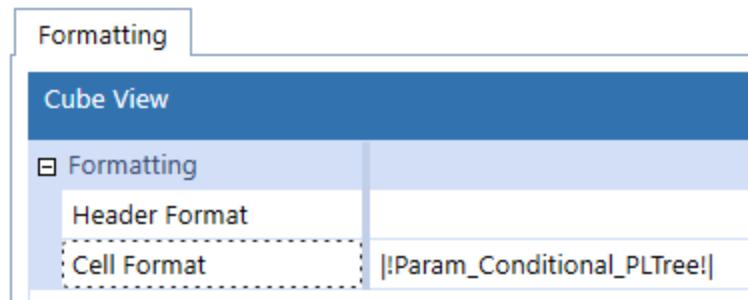
### Parameter Formatting

Conditional formatting definitions can be applied to a cube view as a Literal parameter. To apply, select the format string from the cube view, press Ctrl-C, select the Default value property of a Literal Value-type parameter and press Ctrl-V. Once saved, this parameter can be referred to in a cube view format. As shown, the reference of `!Param_Conditional_PLTree!` in a cube view format would apply the associated format string.

General (Parameter)	
Name	Param_Conditional_PLTree
Description	!Param_Conditional_PLTree!
User Prompt	
Maintenance Unit	XFS Parameters
Sort Order	0
Data Source	
Parameter Type	Literal Value
Default Value	If (RowE2MemberName <> 'None') And (CellAmount < 2000000) Then Bold = True, Italic = True, NumberFormat = [#,\#\#\#,0.000000;(#,\#\#\#,0.00000);0.00000] End If

If (RowE2MemberName <> 'None') And (CellAmount < 2000000) Then  
    Bold = True, Italic = True, NumberFormat = [#,\#\#\#,0.000000;(#,\#\#\#,0.00000);0.00000]  
End If

OK Cancel



## General Settings

This section explains how to view, define, and apply standard report setting properties. These settings determine the information you see and the actions you can take in the cube view. To edit the following properties, on the **Cube Views** page, under **Cube View Groups**, select a cube view and then **General Settings** to expand the slider.

- **Sharing:** Share all or specific rows or columns from another cube view. Sharing can increase efficiency and consistency when building and managing cube views.
- **Common:** These options control general, shortcut, restrictions, suppression, and paging properties.
  - **Is Visible in Profiles:** When set to True, the Cube View is visible in the Cube View Profile. When false, the Cube View is not visible within the Cube View Profile. In these cases, the Cube View would not be visible in Workflow, Excel, Dashboards, or OnePlace.
  - **Page Caption:** Appears at the top of the data explorer grid when viewing the output of a cube view. If left blank, it will get its value from the cube view description. If the description is blank, it will display the cube view name.
  - **Is Shortcut:** Determines whether the cube view is a shortcut.
  - **Shortcut Cube View Name:** Enter the name of another cube view that will open when this cube view is being used as a shortcut.
  - **Literal Parameter Values:** Enter a comma-separated list of name-value pairs for parameter values to be used when this cube view is used as a shortcut to open another cube view. Piped variable names cannot be used (for example, `!ParamView!`) in this setting.

Example: Param1=Value1, Param2=[Value 2]

- **Can Modify Data:** If set to True, the cube view can be modified in OnePlace. If set to False, the cube view is read only.

**NOTE:** You can still make annotations to accounts in a cube view if this property is set to False.

- **Can Calculate, Can Translate, and Can Consolidate:** If set to True, you can right-click on a cell and calculate, translate, or consolidate data in OnePlace.

**NOTE:** These properties coincide with the Can Calculate from Grids Scenario security group. If set to True but the user is not in the group for that given scenario member, they cannot calculate from a cube view grid.

- **Can Modify Suppression:** Set to True to enable a row suppression icon in the data explorer grid toolbar and enable users to turn the suppression settings on and off when viewing a cube view.

- **Include Cell Attachment Status:** If set to True, a cell containing a data attachment will display a red tick mark. If set to False, there will be no indication of a data attachment in the cell. In large data sets, if set to True, it can affect performance.

- See [Sparse Row Suppression](#) and [Cube View Paging](#).

- **Header Text:** The default is a cube view with standard row and column headers. All the header settings control the row and column headers presented when the cube view is run if the default is not used.
- **Header Size:** Column Header Heights align column headers with the bottom of a report. The default setting is -1 (which means they are auto-sized), so for bottom alignment to work in column headers, the height of the column header needs to be set. This is helpful for wrapped headers, which may cause the columns to expand.

Row Header Widths are viewed in the cube view. Excel Row Header Widths are viewed when exporting to or rendering a cube view in Excel. Report Header Widths are viewed when a report on-the-fly is generated from a cube view. These can be changed from the default value. The -1 default value means that these Row Header Widths are auto-sized. Otherwise, set a positive number of at least 100 to determine the number of pixels wide (96 pixels = 2.5 cm). These settings can change for a maximum of six nested rows. The same can be done when reporting to Excel by using the ExcelRowHeader1Width property. The default value for this is also -1.

**NOTE:** If a cube view Row Header Width is not specified, text and font information from all the expanded headers is used to automatically determine the width. The automatic row width maximum is a half of a page. Any text longer than a half of a page is wrapped.

- **Header Overrides:** By default, the software determines which dimensions to show for row and column headers based on the member expansions specified in the Rows and Columns slider. These dimensions can also be selected manually.

Report Column Index for Row Headers can be used for a report to display the row headers after the column instead of on the left side. The default value is -1, but setting this to a positive number changes the cube view column index. Specifying several columns greater than the existing number of columns reverts the row headers back to the left side of the report. Hidden columns are not included in the column count.

**NOTE:** If a cube view report exceeds the width of the page, the software automatically adds a page break on the appropriate column and repeats the row headers on the following page. If Report Column Index for Row Headers or Auto Fit to Page Width are enabled, the report does not repeat the row headers.

- **Report:** These properties apply advanced formatting to cube views.
  - **Custom Report Task:** Custom report formatting can be applied to cube views using a cube view extender business rule or inline formula.
    - Select **No Task** if there are no cube view business rules or inline formulas running on this report.

- Select **Execute Cube View Extender Business Rule** to call a cube view extender.
- Select **Execute Cube View Extender Inline Formula** to include an inline formula.
- **Business Rule:** This property is only enabled when **Execute Cube View Extender Business Rule** is selected as the custom report task. Click the ellipsis and select the cube view extender business rule.
- **Formula:** This property is only enabled when **Execute Cube View Extender Inline Formula** is selected as the custom report task. Click the ellipsis and select the cube inline formula.

Additional options are available for report formatting to set paper size, page orientation, and margins, and to automatically fit the information to a specific page width or number of pages.

- **Excel:** Controls whether gridlines are displayed in the Excel file.
- **Navigation Links:** Determines the information you can view in a cube view in the data explorer grid.
  - **Linked Cube Views:** Enter a comma separated list of cube views that will be available to open when viewing a cube view in the data explorer grid. The cube views specified in this field apply to the entire cube view and are available when you right-click any cube view data cell. See [Linked Cube Views](#).
  - **Linked Dashboards:** Enter a comma separated list of dashboards that you can right-click to open when viewing a cube view in the data explorer grid.
  - **Include Default NavLink Parameters:** Set to **True** in source cube views to use NavLink parameters - with a intuitive, standard syntax, as Bound parameters for any dimension. This simplifies cube view design and reduces error as you:
    - Do not have to define parameters for navigation, and can use a standard syntax to ensure parameter names match across cube views and other objects.

- Can navigate cube views from any dimensional intersection to analyze data that is refined by passed point of view selections, in linked cube views. See [Analyze Data in Linked Cube Views With Passed Point of View Selections](#).
- **Bound Parameter Names for Navigation Links:** Lists all the dimensions that may be assigned to a row when the **Report Navigation Links** property is set to **True** and represents the member name being passed from this report to another using navigation links.
- **Bound Parameter Names:** For a linked cube view to open and display the correct member data for a selected cell, parameters with the member need to be specified using a bound parameter. This lists all the dimensions that may have a parameter included in a linked cube view.

## Navigation Links

Navigation links let you set up one dashboard to launch another in order to drill into more detail or related detail on a certain row of data from a report, while other times it can create a chart of the reviewed data.

The following information provides a detailed example including the steps one must take to create Navigation Links. The implementation of this feature will differ, but for this example an Income Statement Summary Dashboard will launch another Dashboard with more detail:



IS Summary		
	Mar 2011	Mar 2010
Net Income	5,272,583.39	4,915,072.97
Earnings Before Taxes	10,534,767.03	9,260,789.56
Earnings Before Interest and Taxes	11,957,754.03	10,522,018.12
Total Operating Income	15,256,465.17	13,425,689.35
Total Other Income (Expense)	-3,298,711.14	-2,903,671.23
Interest Income	Show Detail For 'Total Other Income (Expense)'.	
IC Interest Income	0.00	0.00
Interest Expense	1,739,538.00	1,539,793.44
IC Interest Expense	0.00	0.00

Click **Total Other Income (Expense)** to display the following dashboard:

## Presenting Data With Books, Cube Views and Dashboards

---



### IS Content

	Mar 2011	Mar 2010
Total Other Income (Expense)	-3,298,711.14	-2,903,671.23
Exchange Rate Gain/(Loss)	-602,366.40	-530,082.43
Gain/(Loss) on Sale of Assets	-300,269.40	-264,237.07
Dividends from Investments in Subs	0.00	0.00
Gain/(Loss) on Commodities	0.00	0.00
Unrealized Gain/(Loss) on Investments	-1,879,970.40	-1,654,373.95
Other Rev/(Exp)	-516,104.94	-454,977.78

**NOTE:** This only works in dashboard mode.

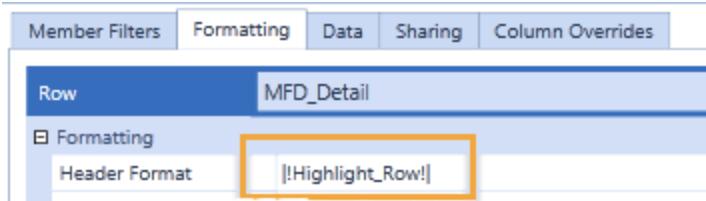
Below is a sample cube view called **IS Summary** which contains income statement summary accounts. First, select the row to be highlighted for navigation. Set Enable Report Navigation Link to True and type the name of the dashboard to open in the Dashboard To Open In Dialog field. The specified dashboard (not a cube view) opens when this highlighted row is selected from this cube view's related dashboard.

A screenshot of a software interface for configuring a cube view. At the top, there are tabs for 'Member Filters', 'Formatting', 'Data', and 'Sharing'. Below these, there is a table with a single row labeled 'Row' and 'MFD\_Detail'. Under the 'Row' column, there are sections for 'General', 'List Parameter', and 'Report Navigation Links'. The 'Report Navigation Links' section is highlighted with an orange box. It contains two fields: 'Enable Report Navigation Link' set to 'True' and 'Dashboard To Open In Dialog' set to 'IS Content'. The rest of the table rows are empty.

This next step is optional. Add a header format using a cube view style. This displays the row in a color different from the other rows to indicate the row is clickable. In this case, `!Highlight_Row!` is related to a parameter that was made in a sample application. When this cube view is run, the format of `TextColor=Blue` activates for this row.

## Presenting Data With Books, Cube Views and Dashboards

---



The cube view passes the clicked account to the other dashboard and there the account and its children can be viewed. Now, determine how the account is to be passed. Go to the General Settings slider and go to the Report Navigation Links section. A value of ClickedAccount was entered for the Account Bound Parameter Name which is going to be passed from one dashboard to another. More than one dimension can be passed if it is defined in rows, but in this case this does not need to happen.

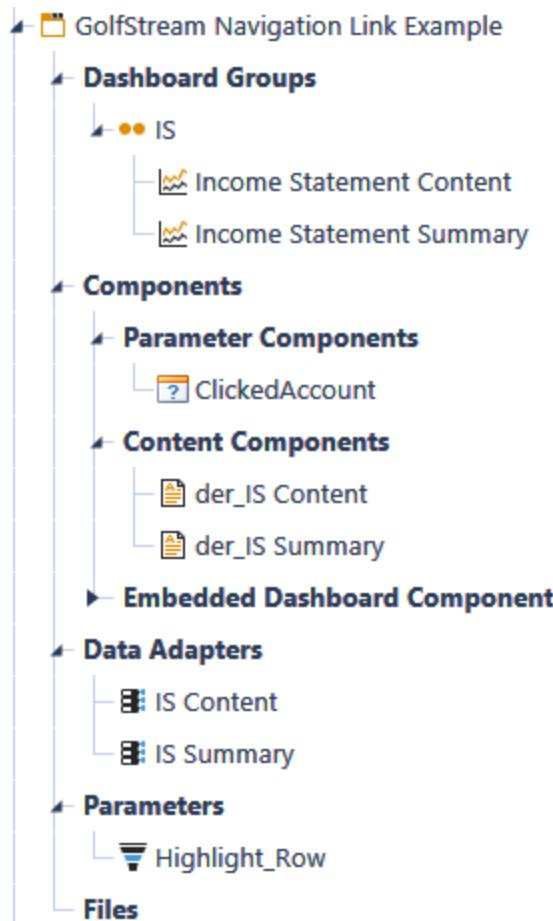
The screenshot shows the 'General Settings' slider with the 'Report Navigation Links' section selected. Under 'Row Header Report Navigation Links', the 'Account Bound Parameter Name' is set to 'ClickedAccount'.

Copy this cube view to have the same columns and POV settings for the drilled cube view containing the contents. This one is called IS Content because it contains the content of the drilled data. Whatever it is called, be consistent and come to an agreement with the project team.

In the new cube view (called IS Content in this case), remove the rows that were there and add the rows that will be seen when this Dashboard appears. In this case, refer to the parameter for ClickedAccount and add the ChildrenInclusive extension.

The screenshot shows the 'Row' section of a cube view configuration. It includes four rows under the 'Account' column, each with a dropdown menu. The first row has the formula 'A#;!ClickedAccount!;.ChildrenInclusive'. The subsequent rows are labeled '(Not Used)' and contain the formulas 'Level 2: Member Filter', 'Level 3: Member Filter', and 'Level 4: Member Filter' respectively.

Under dashboards, a maintenance unit  was created and called GolfStream Navigation Link Example, which stores all the objects needed to get this to work properly.



Starting from the bottom of the example above is the parameter  needed to highlight the row.

## Presenting Data With Books, Cube Views and Dashboards

---

General (Parameter)	
Name	Highlight_Row
Description	
User Prompt	
Maintenance Unit	GolfStream Navigation Link Example
Sort Order	0
Data Source	
Parameter Type	Literal Value
Default Value	TextColor=Blue
Result Format String Type	Default
Result Custom Format String	

The data adapters  are required to point to the cube views. First, set up one for IS Content. Set the Command Type to Cube View and set the cube view to IS Content.

Next, set up the data adapter for IS Summary. Set the Command Type to Cube View and set the cube view to IS Summary. Set the **Include Row Navigation Link** property to True to drill from Cube Views in a Dashboard.

Include Member Details	False
Include Row Navigation Link	True
Include HasData Status	True

Now, set up two content components for a dashboard. Name these with the prefix of der\_ because they are data explorer reports and not charts or another form of component. Again, come up with a naming schema with the project team, so the result is organized. In this case, click  to create a component and chose Data Explorer Report as the type. Do this for both IS Summary and IS Content and click  in the dashboard toolbar to attach the appropriate data adapter to each.

Select Parameter Components and create a new component  with a type of Supplied Parameter. In the Bound Parameter field, enter the name of the parameter being passed, which is ClickedAccount in this example. Do this for each parameter being passed from dashboard to dashboard.

Create a dashboard group (called IS in this example) and two dashboards. Create one for the launching dashboard (for example, IS Summary) and one for the launched dashboard (for example, IS Content). In this example, a layout of Uniform is being used, but the use of this feature may vary.

Under Dashboard Components, attach the data explorer report component. Do this for both IS Summary and IS Content. For just the IS Summary (the launched dashboard), also attach the parameter component being passed (ClickedAccount in this example). This allows that account to be passed from the initial dashboard to the other.

## Linked Cube Views

Linked cube views provide the option to launch a cube view from another when viewing it in the Data Explorer grid, spreadsheet tool, or Excel Add-In. You can right-click a cube view's data cell and open a separate cube view to access more detail and visibility. This applies to any cube view, such as those for data entry forms. See:

- [Create a sample linked cube view](#)
- [Link a cube view to another entire cube view](#)
- [Link a cube view to specific rows or columns](#)
- [Nest linked cube views](#)

**TIP:** For information about evaluating source cube view data in a linked cube view using passed point of view selections, see [Analyze data with passed point of view selections](#).

### Create a Sample Linked Cube View

The following describes how to create a sample linked cube view. In this example, an Income Statement Summary cube view has an Account Detail cube view assigned to all data cells and a Sales by Product cube view linked to specific rows. You can also add a second level of linked cube views for an even more granular analysis.

## Presenting Data With Books, Cube Views and Dashboards

---

Income Statement Summary for Total GolfStream				
	Jan 2011	Feb 2011	Mar 2011	
60999 - Net Sales	55,524,469.79	134,405,136.90	271,444,823.46	
43000 - Cost of Goods Sold	32,331,583.56	77,442,019.33	179,453,475.33	
<b>61000 - Gross Income</b>	<b>23,192,886.23</b>	<b>56,963,114</b>		
54400 - Total Operating Exp Before Allocation	17,381,215.25	41,588,438		
54350 - Total Allocations				
<b>62000 - Total Operating Income</b>	<b>5,811,670.98</b>	<b>15,374,676</b>		
62999 - Total Other Income (Expense)	-837,759.24	-2,781,438		
<b>63000 - Earnings Before Interest and Taxes</b>	<b>4,973,911.74</b>	<b>12,593,238</b>		
63100 - Interest Income	132,019.25	316,000		
55000 - Interest Expense	730,807.50	1,751,300		
<b>64000 - Earnings Before Taxes</b>	<b>4,375,123.49</b>	<b>11,158,438</b>		
56999 - Total Income Taxes	11,527,132.77	11,778,300		
65200 - Extraordinary Income/Expense (net of tax)	-88,375.00	-212,300		

A context menu is open over the cell containing '11,158,438' in the 'Earnings Before Taxes' row. The menu includes options like Calculate, Translate, Consolidate, Data Attachments For Selected Cell, Data Attachments For Selected Data Unit, Cell POV Information, Cell Status, Data Unit Statistics, and two links: 'Navigate To 'AccountDetail'' and 'Navigate To 'ProductDetail''. The last two items are highlighted with a yellow box.

## Link a Cube View to an Entire Cube View

1. Select the cube view to be linked and apply to the entire cube view, specific columns, or specific rows. If the linked cube view applies to the entire cube view, you can right-click any data cell in the Data Explorer grid, spreadsheet tool, or Excel Add-In to access the cube view. If it only applies to columns or rows, the linked cube view is available when you right-click on a data cell in that row or column.

**NOTE:** Column settings override cube view settings and row settings override column settings.

2. Once the base cube view is determined, in this example IncomeStatementSummary, create a cube view that will display more detailed data.

The AccountDetail cube view will display Account details based on the selected data cell. The IncomeStatementSummary's data displays Parent accounts for a specific time, so the AccountDetail's data will display the selected Account's children for that same time period.

## Presenting Data With Books, Cube Views and Dashboards

---

3. In the AccountDetail cube view, configure the columns and rows to query the correct data.

The screenshot shows the configuration interface for the Account Detail cube view. At the top, there are tabs: Member Filters, Formatting, Data, and Sharing. The Data tab is selected. Below the tabs, there are two sections: Row and Column. The Row section has a header 'Drill Account Detail' and a dropdown menu set to 'Account'. The value field contains the expression 'A#|!DrillAccount!|.TreeDescendantsInclusive'. The Column section has a header 'Time' and a dropdown menu set to 'Time'. The value field contains the expression 'T#|!DrillTime!|'. Both sections have edit icons on the right.

!DrillAccount! is a Bound parameter and will display the correct account data based on the selected account in the IncomeStatementSummary cube view.

The screenshot shows the configuration interface for the Time column of the Account Detail cube view. It has a header 'Time' and a dropdown menu set to 'Time'. The value field contains the expression 'T#|!DrillTime!|'. An edit icon is visible on the right.

!DrillTime! is a Bound parameter and will display the correct Time data based on the selected time in the IncomeStatementSummary cube view

4. Ensure the AccountDetail's POV matches the IncomeStatementSummary's POV with the exception of Account and Time. Enter the !DrillAccount! and !DrillTime! parameters into these member fields to ensure that the AccountDetail cube view displays data based on the selected Account and Time in the IncomeStatementSummary.
5. Add any required formatting to the AccountDetail cube view. In this example, the following was added to Page Caption:

The screenshot shows the configuration interface for the Page Caption. It has two fields: 'Page Caption' and 'Account Detail for |CVAccOUNTDESC|'. The 'Page Caption' field is empty, and the 'Account Detail for' field contains the expression '|CVAccOUNTDESC|'.

6. To assign the AccountDetail to the entire IncomeStatementSummary cube view, select IncomeStatementSummary and click **General > Navigation Links**. Enter the names of the linked cube view in **Linked Cube Views** or click the ellipsis to find the cube view using Object Lookup.

7. Enter the Bound parameters which will pass from one cube view to the next. In this example, a Bound parameter was entered for Account and Time so both members are based on the selected IncomeStatementSummary data cell.

General Settings	
Sharing	
Common	
Header Text	
Header Size	
Header Overrides	
Report	
Excel	
Navigation Links	
<b>Navigation Links</b>	
Linked Cube Views	AccountDetail
Linked Dashboards	MainTemplate
Include Default NavLink Parameters	True
<b>Bound Parameter Names</b>	
Cube Bound Parameter Name	
Entity Bound Parameter Name	
Parent Bound Parameter Name	
Consolidation Bound Parameter Name	
Scenario Bound Parameter Name	
Time Bound Parameter Name	DrillTime
View Bound Parameter Name	
Account Bound Parameter Name	DrillAccount
Flow Bound Parameter Name	
Origin Bound Parameter Name	
IC Bound Parameter Name	
UD1 Bound Parameter Name	
UD2 Bound Parameter Name	
UD3 Bound Parameter Name	
UD4 Bound Parameter Name	
UD5 Bound Parameter Name	
UD6 Bound Parameter Name	
UD7 Bound Parameter Name	
UD8 Bound Parameter Name	

Run IncomeStatementSummary, noting that you can right-click cells to access the AccountDetail cube view.

## Presenting Data With Books, Cube Views and Dashboards

---

Income Statement Summary for Total GolfStream				
	Jan 2011	Feb 2011	Mar 2011	
60999 - Net Sales	55,524,469.79	134,405,136.90	271,444,823.46	
43000 - Cost of Goods Sold	32,331,583.56	77,442,019.33	179,453,475.22	
61000 - Gross Income	23,192,886.23	56,963,117.57	91,99	
54400 - Total Operating Exp Before Allocation	17,381,215.25	41,588,423.50	123,06	
54350 - Total Allocations				
62000 - Total Operating Income	5,811,670.98	15,374,694.07	-31,07	
62999 - Total Other Income (Expense)	-837,759.24	-2,781,406.69	-4,86	
63000 - Earnings Before Interest and Taxes	4,973,911.74	12,593,287.39	-35,96	
63100 - Interest Income	132,019.25	316,677.00	52	
55000 - Interest Expense	730,807.50	1,751,538.00	2,89	
64000 - Earnings Before Taxes	4,375,123.49	11,158,426.39	-38,33	

Right-click context menu:

- Print
- Calculate
- Translate
- Consolidate
- Data Attachments For Selected Cell
- Data Attachments For Selected Data Unit
- Cell POV Information
- Cell Status
- Data Unit Statistics
- Navigate To 'AccountDetail'

AccountDetail - Linked From IncomeStatementSummary - T#2011M3:A#43000

Account Detail for Cost of Goods Sold

	Mar 2011
43000 - Cost of Goods Sold	179,453,475
41000 - Operating Cost of Goods Sold	179,315,575
42000 - IC Cost of Goods Sold	137,900

## Link a Cube View to Specific Rows or Columns

This topic describes how to assign a linked cube view to specific rows. The ProductDetail cube view will display product details based on the IncomeStatementSummary's selected data cell. The IncomeStatementSummary's data displays parent accounts for a specific time, so the ProductDetail's data displays the selected account's product sales.

In the ProductDetail cube, configure the rows/columns, POV, and formatting to ensure that the user sees the correct data. See the AccountDetail instructions above for details.

To assign the ProductDetail cube view to IncomeStatementSummary rows, select the IncomeStatementSummary cube view, then go to the Linked Cube Views property on the Data tab in the Rows/Columns. Select the row or column and enter the name of the cube view.

## Presenting Data With Books, Cube Views and Dashboards

---

The screenshot shows a Data Explorer interface with a grid of data and a navigation panel on the right.

**Data Grid:**

Default	Time
Net Sales_T	9,999.99
COGS_T	9,999.99
Gross Income_T	0.00

**Navigation Panel:**

- Member Filters
- Formatting
- Data** (selected)
- Sharing
- Column Overrides

**Row Properties:**

Row	Net Sales_T
<input checked="" type="checkbox"/> General	
<input type="checkbox"/> Navigation Links	

**General Settings (under Row > General):**

Enable Report Navigation Link	False
-------------------------------	-------

**Navigation Links (under Row > Navigation Links):**

Dashboard To Open In Dialog	
Linked Cube Views	AccountDetail, ProductDetail

**NOTE:** Due to overrides, assign the AccountDetail cube view to this property too. Previously, AccountDetail was assigned to the entire cube view. However, because a cube view is being assigned to a row, the cube view settings are overridden and only display the cube view specified for this property. For both cube views to be available on the row, specify them both.

If any Bound parameters need to be entered (in this example, the same Bound parameters are used for both cube views), enter them under Navigation Links in General Settings.

In the example above, the ProductDetail Cube View was assigned to the Net Sales row. When a user right-clicks on a Net Sales cell in the Data Explorer grid, Spreadsheet tool, or Excel Add-In, this cube view is available.

## Presenting Data With Books, Cube Views and Dashboards

---

Income Statement Summary for Total GolfStream				
	Jan 2011	Feb 2011	Mar 2011	
60999 - Net Sales	55,524,469.79	134,405,136.00	271,444,922.46	
43000 - Cost of Goods Sold	32,331,583.56	77,442,019.00		
<b>61000 - Gross Income</b>	<b>23,192,886.23</b>	<b>56,963,117.00</b>		
54400 - Total Operating Exp Before Allocation	17,381,215.25	41,588,423.00		
54350 - Total Allocations				
<b>62000 - Total Operating Income</b>	<b>5,811,670.98</b>	<b>15,374,694.00</b>		
62999 - Total Other Income (Expense)	-837,759.24	-2,781,406.00		
<b>63000 - Earnings Before Interest and Taxes</b>	<b>4,973,911.74</b>	<b>12,593,287.00</b>		
63100 - Interest Income	132,019.25	316,677.00		
55000 - Interest Expense	730,807.50	1,751,537.00		
<b>64000 - Earnings Before Taxes</b>	<b>4,375,123.49</b>	<b>11,158,426.00</b>		

A context menu is open over the cell containing '11,158,426.00' in the Mar 2011 column. The menu items are: Calculate, Translate, Consolidate, Data Attachments For Selected Cell, Data Attachments For Selected Data Unit, Cell POV Information, Cell Status, Data Unit Statistics, Navigate To 'AccountDetail', and Navigate To 'ProductDetail'. The 'Navigate To' items are highlighted with orange boxes.

## Nest Linked Cube Views

You can link a cube views to cube views that are already linked. In this example, the ProductDetail\_2 cube view is linked to ProductDetail and displays specific product data tied to a selected ProductDetail cell.

1. In ProductDetail\_2, configure the rows and columns as needed.

Row	Drill Product Detail_2
UD2	U2#!DrillProduct!.Children

2. Enter a `!DrillProduct!` Bound parameter to display specific product details. For example, if you right-click a Clubs cell in the ProductDetail cube view and select the ProductDetail\_2, sales details for Clubs displays.

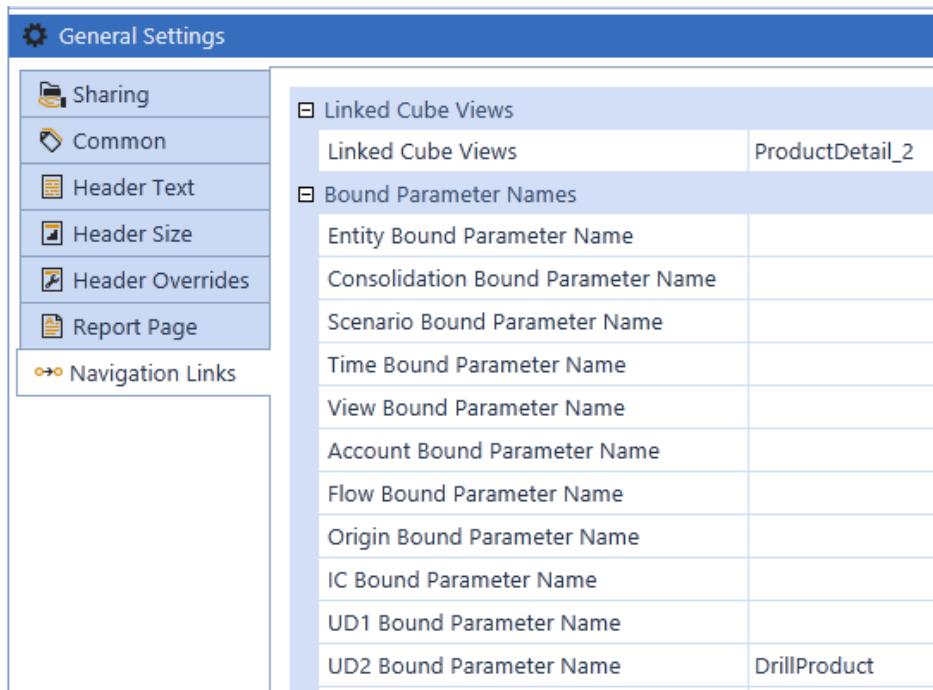
Column	Time
Time	T#!DrillTime!

3. Ensure a `!DrillTime!` parameter is also used.

4. Set the POV for ProductDetail\_2 so it matches the ProductDetail POV, except for the User Defined 2 member. In this example, products are viewed using the UD2 member. Enter `!DrillProduct!` in User Defined 2.
5. Add any required formatting to the ProductDetail\_2, such as page captioning:

Page Caption	Product Detail for  CVUD2
--------------	---------------------------

6. Open ProductDetail and navigate to Linked Cube View under General Settings > Navigation Links.
7. Click the ellipsis and select the ProductDetail\_2 cube view. This link the cube view to the entire ProductDetail cube view.
8. Enter the Bound parameter from ProductDetail\_2.



9. Run ProductDetail from IncomeStatementSummary, right-click cells and select **Navigate To ProductDetail\_2**.

## Analyze Data in Linked Cube Views With Passed Point of View Selections

Use the **Include Default NavLink Parameters** property to enhance cube view design and navigation for optimal data analysis. Set the property to **True** to use the following standard syntax to more easily look up parameters and ensure parameter names match across linked cube views and other objects.

```
| !dimensionNavLink! |
```

Point of view selections you specify in a source cube view pass to the linked cube view, for a refined, focused analysis. For example, to pass the Time dimension from a source cube view to a linked cube view, specify `| !TimeNavLink! |` in the linked cube view's POV. The data in the cell point of view in the source cube view (2018Q1 as shown below) passes to the linked cube view's point of view at runtime.

To use passed point of view selections in a linked cube view:

1. Click **Application > Cube Views**, and then the cube view containing the source data to access and analyze further in a linked cube view.
2. Click **General Settings > Navigation Links**.
3. Set **Include Default NavLink Parameters** to **True** and click **Save**.
4. Open the linked cube view in which to retrieve and evaluate the source data based on point of view selections. Set **Include Default NavLink Parameters** to **False**.

## Presenting Data With Books, Cube Views and Dashboards

---

5. Click **POV** , specify `!dimensionNavLink!` for each dimension to analyze and then click **Save**.

Designer Advanced

Name	Description
Linked_NavigationLinks	

**POV**

Entity Member	
Parent Member	<code>!EntityNavLink!</code>
Consolidation Member	
Scenario Member	Actual
Time Member	<code>!TimeNavLink!</code>
View Member	<code>!ViewNavLink!</code>
Account Member	<code>!AccountNavLink!</code>
Flow Member	None
Origin Member	Top
IC Member	Top
UD1 Member	Top
UD2 Member	Top
UD3 Member	-

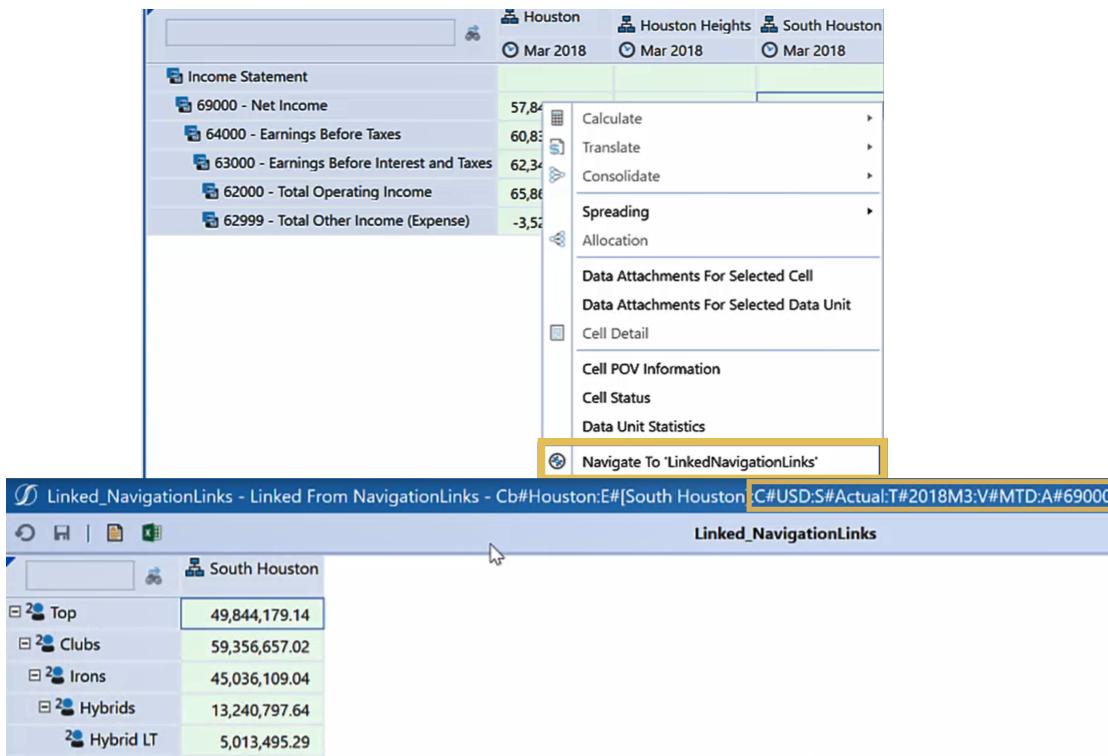
General Settings

6. Return to the source cube view and click **Open Data Explorer**.

## Presenting Data With Books, Cube Views and Dashboards

---

7. Right-click cells to navigate to the linked view, retrieving data for the specified point of view parameters that are also identified in the header.



## Format a Report Page

The following settings in the cube view address page formatting:

- Header (text, size, overrides)
- Report
- Headers and footers

Ensure the general settings are correct before formatting your page. See [General Settings](#).

To edit the header and report settings, on the **Cube Views** page, under **Cube View Groups**, select a cube view and then **General Settings** to expand the slider.

## Header Settings

Row and column headers can be adjusted to refine your cube view.

- **Header Text:** Decide if you want to see the name, description, or both for each of the dimension types. Also, set the Culture property of a cube view, which determines the language for the report. Ensure the Member Descriptions Culture properties are set up correctly before setting the Culture property.
- **Header Size:** Provide a height for columns and width for rows. The rows can be edited separately by Data Grid (Row Header), Excel (Excel Row Header), and PDF (Report Row Header). Each of these items defaults to -1, but you can change them by typing a pixel number.
- **Header Overrides:** Change which dimensions display in the rows and columns. The default for a cube view is to display what is available in the rows and columns. However, you can apply Header Overrides to change the order or add dimensions.

The Report Column Index for Row Headers can also be set here. This changes the location of the row headers from their usual place (the left). Type the number of columns that should display before the row headers are shown. This only applies to the PDF version of the report.

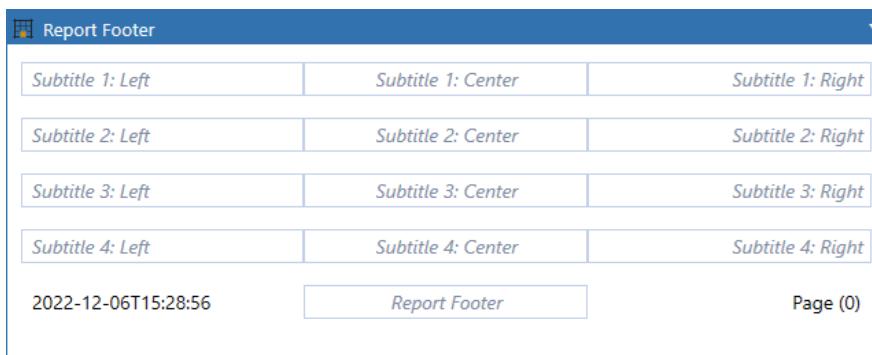
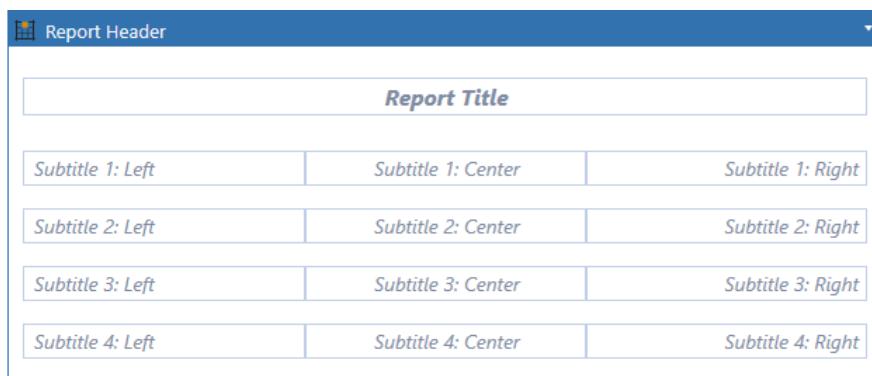
## Report Settings

These properties apply advanced formatting to cube views.

- **Custom Report Formatting:** Write or assign cube view extender rules to provide intricate cube view formatting. This only applies to the PDF version.
- **Paper:** Select the paper size.
- **Landscape:** Select the page orientation.
- **Margin:** Set margins on individual reports. Type a pixel number to override the default settings of the page in the general settings.
- **Auto Fit Settings:** Set **Auto Fit to Page Width to True** if the report should re-size to fit all columns on one page. Use the **Auto Fit Number of Pages Wide** property to type the number of pages to display the columns.

### Header and Footer Settings

Report headers and footers control what displays at the top and bottom of the page when a cube view is run as a data explorer report. Standardize headers and footers across reports and make as dynamic as possible to reduce reporting maintenance. On the Designer tab of the Cube View, Report Header and Report Footer are separate sliders. Enter text in the fields or use substitution variables.



Use substitution variables to reduce the maintenance of headers and footers. Substitution variables are short scripts that use pipe characters to include a predefined substitution variable. They come with every installation of OneStream and cannot be edited, so you do not need to create or maintain them. Substitution variables can be used throughout the application. For cube views, you can use them for:

- Headers and footers
- Rows and columns

- Cube view page captions

To copy and paste a substitution variable into your application:

1. On the **Cube Views** page, click **Object Lookup** to open the dialog box.
2. Under **Object Type**, select **Substitution Variables** to see the list of options.
3. Select the name of the substitution variable.
4. Click **Copy to Clipboard**.
5. Paste the substitution variable where needed.

## Calculations

When you create a calculation in a cube view, the data is generated on-the-fly and not stored in the database. When you run the cube view, the calculation returns a value.

You can use calculations in a cube view to calculate:

- Variances
- Sums
- Differences
- Ratios

Calculations are useful for variance reporting, What If scenarios in cube views, and KPI calculations.

## Types of Cube View Calculations

There are three ways to create calculations in cube views. Each option uses the `GetDataCell` function, which retrieves specific cells and performs math or business rule operations.

The most common ways to create a calculation in a cube view use:

- "GetDataCell Expressions" on the next page
- "Column and Row Expressions" on page 648

- "Dynamic Member Formulas and Cube Views" on page 656

Choose the option that has the least maintenance and highest usability.

**NOTE:** You can also create more complex calculations with business rules.

## GetDataCell Expressions

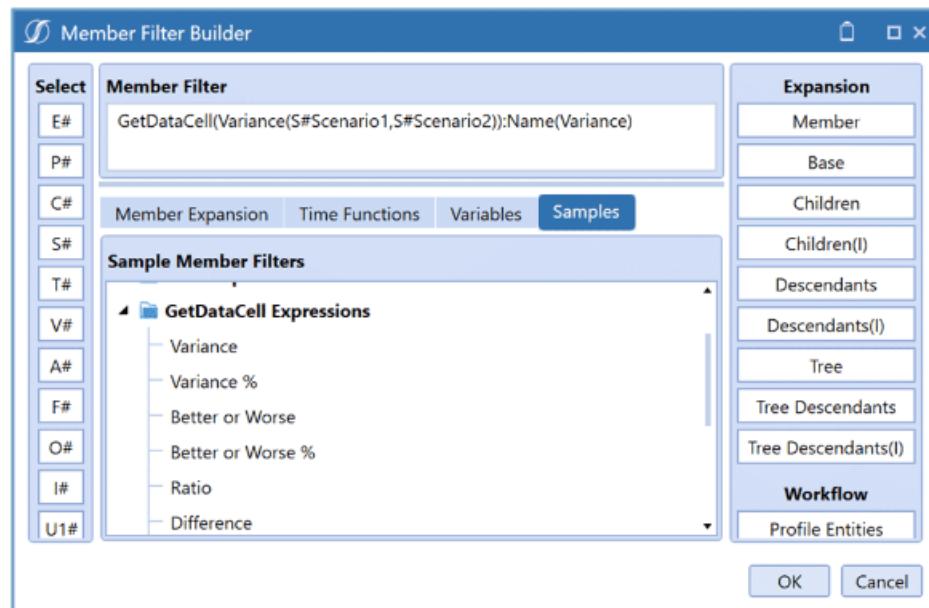
Write GetDataCell expressions in the cube view row or column. It will calculate at runtime. The expressions require a reference to specific dimension members to return a data point. Use GetDataCell expressions if you are running calculations on members that are not currently in the cube view or if the expression is easy to create and maintain. GetDataCell expressions are simpler versions of dynamic calculations.

**NOTE:** Use one of the other calculation types if you are querying multiple member combinations.

## Samples Tab

In the Member Filter Builder, use the Samples tab to create simple GetDataCell expressions in your cube view.

In Member Filter Builder, click **Samples**.



## Presenting Data With Books, Cube Views and Dashboards

---

There is a list of preexisting GetDataCell expressions you can use. For detailed information, see the following table.

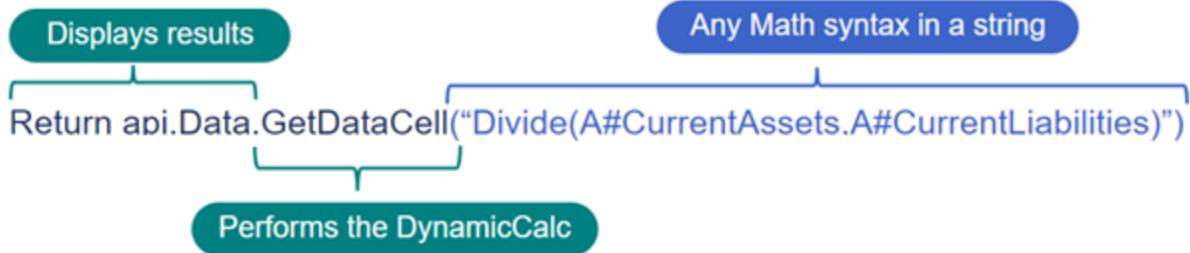
GetDataCell Expressions

Function Name	Description	Example
Variance	Calculates the difference as a ratio between two scenarios using the Variance function.	GetDataCell(Variance(S#Scenario1, S#Scenario2)):Name(Variance)
Variance%	Calculates the difference as a percentage between two scenarios using the VariancePercent function.	GetDataCell(VariancePercent (S#Scenario1, S#Scenario2)):Name(Var%)
Better or Worse	Calculates the difference between two scenarios while considering account types using the BWDiff function.	GetDataCell(BWDiff(S#Scenario1, S#Scenario2)):Name(BW Diff)
Better or Worse%	Calculates the difference as a percentage between two scenarios while considering account types using the BWPercent function.	GetDataCell(BWPercent(S#Scenario1, S#Scenario2)):Name(BW%)
Ratio	Calculates the ratio between two scenarios using the Divide function.	GetDataCell(Divide(S#Scenario1, S#Scenario2)):Name(Ratio)

Function Name	Description	Example
Difference	Calculates the difference between scenarios using the Subtraction function.	GetDataCell(S#Scenario1-S#Scenario2):Name(Difference)
Sum	Calculates the sum of two scenarios using the Addition function.	GetDataCell(S#Scenario1+ S#Scenario2:Name(Total))
Custom Function (condensed syntax)	Calls a Finance business rule using the BR# function.	GetDataCell(BR#[MyBusinessRuleName, MyFunctionName]):Name(Custom Function)
Custom Function	Calls a Finance business rule using the BR# function.	GetDataCell(BR#[BRName=MyBusinessRuleName, FunctionName=MyFunctionName]):Name(Custom Function)
Custom Function with Parameters	Calls a Finance business rule using the BR# function.	GetDataCell(BR#[BRName=MyBusinessRuleName, FunctionName=MyFunctionName, Name1=Value1, AnotherName=[Another Value]]):Name(Custom Function)

## Components of GetDataCell

A GetDataCell expression is the performance section of a dynamic calculation. It tells the dynamic calculation what it will be doing. The following example shows a common GetDataCell expression. GetDataCell expressions are simpler versions of dynamic calculations.



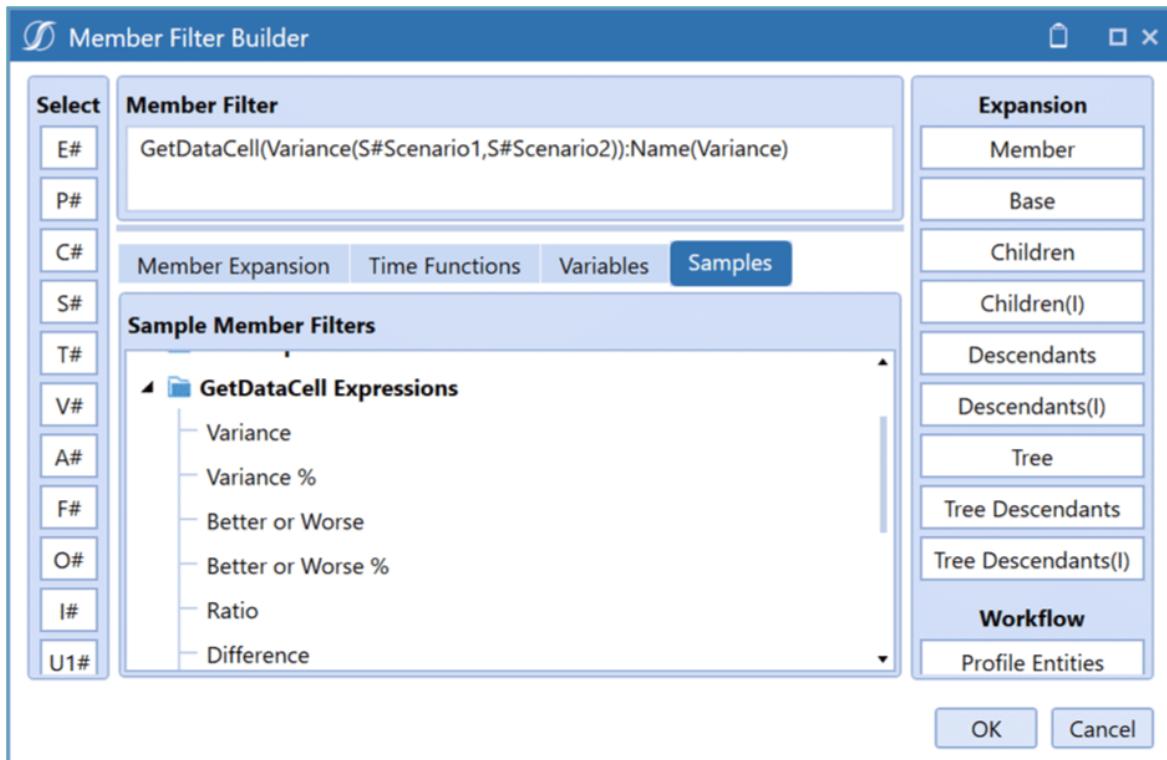
Return api.Data.GetDataCell("Divide(A#CurrentAssets,A#CurrentLiabilities)")

- **Return api.Data:** Tells OneStream which results to display.

**NOTE:** This is not required for GetDataCell expressions, but it is required for dynamic calculations.

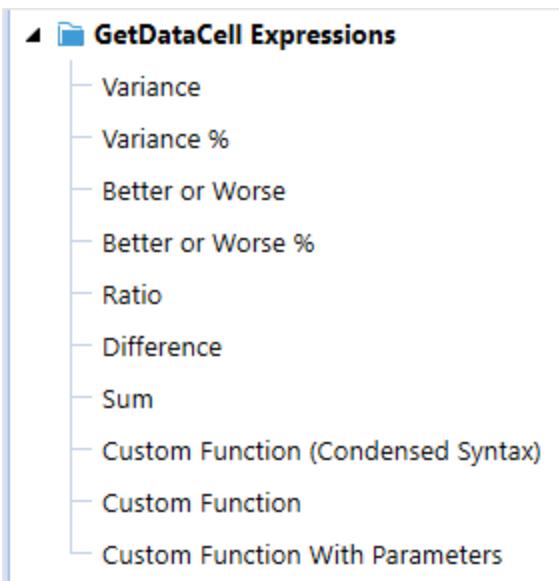
- **GetDataCell:** Performs the calculation.
- **("Divide(A#CurrentAssets,A#CurrentLiabilities):** Math syntax in a string format to calculate.

You can use the Samples tab to create simple GetDataCell expressions in your cube view.



## GetDataCell Types

There are several GetDataCell expressions. You can find a complete list under **OnePlace > Dashboards > Application Reports > Application Reports Dashboards > Report Groups > Application Analysis**.



- **Variance:** Calculate the difference as a ratio between two scenarios using the [Variance] function in a GetDataCell expression.

For example: `GetDataCell(Variance(S#Scenario1,S#Scenario2)):Name(Variance)`

- **Variance %:** Calculate the difference as a percentage between two scenarios using the [VariancePercent] function in a GetDataCell expression.

For example: `GetDataCell(VariancePercent(S#Scenario1,S#Scenario2)):Name(Var %)`

- **Better or Worse:** Calculate the difference between two scenarios while considering account types using the [BWDiff] function in a GetDataCell expression.

For example: `GetDataCell(BWDiff(S#Scenario1,S#Scenario2)):Name(BW Diff)`

- **Better or Worse %:** Calculate the difference as a percentage between two scenarios while considering account types using the [BWPercent] function in a GetDataCell expression.

For example: `GetDataCell(BWPercent(S#Scenario1,S#Scenario2)):Name(BW %)`

- **Ratio:** Calculate the ratio between two scenarios using the [Divide] function in a GetDataCell expression.

For example: `GetDataCell(Divide(S#Scenario1,S#Scenario2)):Name(Ratio)`

- **Difference:** Calculate the difference between two scenarios using the [Subtraction] function in a GetDataCell expression.

The example that appears when you double click this expression in Member Filter Builder:  
GetDataCell(S#Scenario1-S#Scenario2):Name(Difference)

- **Sum:** Calculate the sum of two scenarios using the [Addition] function in a GetDataCell expression.

For example: GetDataCell(S#Scenario1+S#Scenario2):Name(Total)

- **Custom Function (Condensed Syntax):** Call a Finance business rule using the [BR#] function in a GetDataCell expression.

For example: GetDataCell(BR#[MyBusinessRuleName, MyFunctionName]):Name(Custom Function)

- **Custom Function:** Call a Finance business rule using the [BR#] function in a GetDataCell expression. For example:

GetDataCell(BR#[BRName=MyBusinessRuleName, FunctionName=MyFunctionName]):Name(Custom Function)

- **Custom Function with Parameters:** Call a Finance business rule using the [BR#] function in a GetDataCell expression. For example:

GetDataCell(BR#[BRName=MyBusinessRuleName, FunctionName=MyFunctionName, Name1=Value1, AnotherName=[Another Value]]):Name(Custom Function)

## Create a GetDataCell Calculation

GetDataCell expressions run a calculation directly in a cube view by referencing specific dimension members.

In the Member Filter Builder, use the Samples tab to create simple GetDataCell expressions.

Some common errors when developing GetDataCell expressions are:

- Not having the correct dimension listed. Make sure to use the Member Filter Builder if you are not sure what to use.
- Not having the correct number of parentheses. Make sure to use the correct number of opening and closing parentheses.

To add a row expression to a column in a cube view:

## Presenting Data With Books, Cube Views and Dashboards

---

1. In OneStream, go to **Application > Presentation > Cube Views**.
2. Select a cube view group and then a cube view.

The screenshot shows the OneStream Cube Views Designer interface. The left sidebar lists various cube view groups and specific cube views, with 'Headcount' selected. The main area is divided into sections: 'Designer' (selected), 'Advanced', 'Name' (Headcount), 'Description', 'POV', 'General Settings', 'Report Header', and 'Rows and Columns'. The 'Rows and Columns' section contains a table with data. Below it are 'Formatting' and 'Report Footer' sections.

Default	Start	Changes	End	Check Sum	Comments
MFG	9,999.99	9,999.99	9,999.99	9,999.99	9,999.99
MFD_Detail	9,999.99	9,999.99	9,999.99	9,999.99	9,999.99
G&A	9,999.99	9,999.99	9,999.99	9,999.99	9,999.99
G&A_Detail	9,999.99	9,999.99	9,999.99	9,999.99	9,999.99
Selling	9,999.99	9,999.99	9,999.99	9,999.99	9,999.99
Selling_Detail	9,999.99	9,999.99	9,999.99	9,999.99	9,999.99

3. Expand the Rows and Columns slider.

## Presenting Data With Books, Cube Views and Dashboards

---

The screenshot shows the 'Headcount' cube view in 'Designer' mode. The 'Rows and Columns' section is selected. A table is displayed with columns: Start, Changes, End, Check Sum, and Comments. The data rows are MFG, MFD\_Detail, G&A, G&A\_Detail, Selling, and Selling\_Detail, all showing values of 9,999.99.

Default		Start	Changes	End	Check Sum	Comments
MFG		9,999.99	9,999.99	9,999.99	9,999.99	9,999.99
MFD_Detail		9,999.99	9,999.99	9,999.99	9,999.99	9,999.99
G&A		9,999.99	9,999.99	9,999.99	9,999.99	9,999.99
G&A_Detail		9,999.99	9,999.99	9,999.99	9,999.99	9,999.99
Selling		9,999.99	9,999.99	9,999.99	9,999.99	9,999.99
Selling_Detail		9,999.99	9,999.99	9,999.99	9,999.99	9,999.99

4. To add a row to insert the CVC/CVR calculation, in Rows and Columns, click the plus sign.

The screenshot shows the 'Headcount' cube view in 'Designer' mode. The 'Rows and Columns' section is selected. A table is displayed with columns: Start, Add Row, Add Column, and End. The data rows are MFG, MFD\_Detail, G&A, G&A\_Detail, Selling, and Selling\_Detail, all showing values of 9,999.99. An 'Add Row' button is highlighted with a cursor.

Default		Start	Add Row	Add Column	End
MFG		9,999.99	9,999.99	9,999.99	9,999.99
MFD_Detail		9,999.99	9,999.99	9,999.99	9,999.99
G&A		9,999.99	9,999.99	9,999.99	9,999.99
G&A_Detail		9,999.99	9,999.99	9,999.99	9,999.99
Selling		9,999.99	9,999.99	9,999.99	9,999.99
Selling_Detail		9,999.99	9,999.99	9,999.99	9,999.99

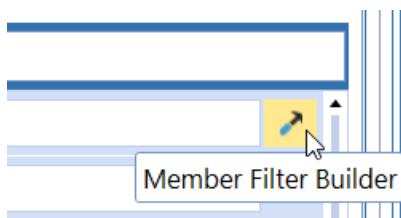
5. In Member Filters, rename the row.

## Presenting Data With Books, Cube Views and Dashboards

---

The screenshot shows the 'Rows and Columns' configuration interface. At the top, there's a toolbar with a 'Layout' dropdown and various icons for managing rows and columns. Below this is a table with six columns: Start, Changes, End, Check Sum, and Comments. The table contains data for several categories: Finance, MFG, MFD\_Detail, G&A, G&A\_Detail, and Selling. All values in the table are 9,999.99. At the bottom of the interface, there are tabs for Member Filters, Formatting, Data, Sharing, and Column Overrides, with 'Member Filters' currently selected. Under the 'Member Filters' tab, there's a row labeled 'Row' with a dropdown set to 'Finance'. Below this are three levels of member filters, each with a dropdown set to '(Not Used)'. The first level is labeled 'Level 1: Member Filter', the second 'Level 2: Member Filter', and the third 'Level 3: Member Filter'.

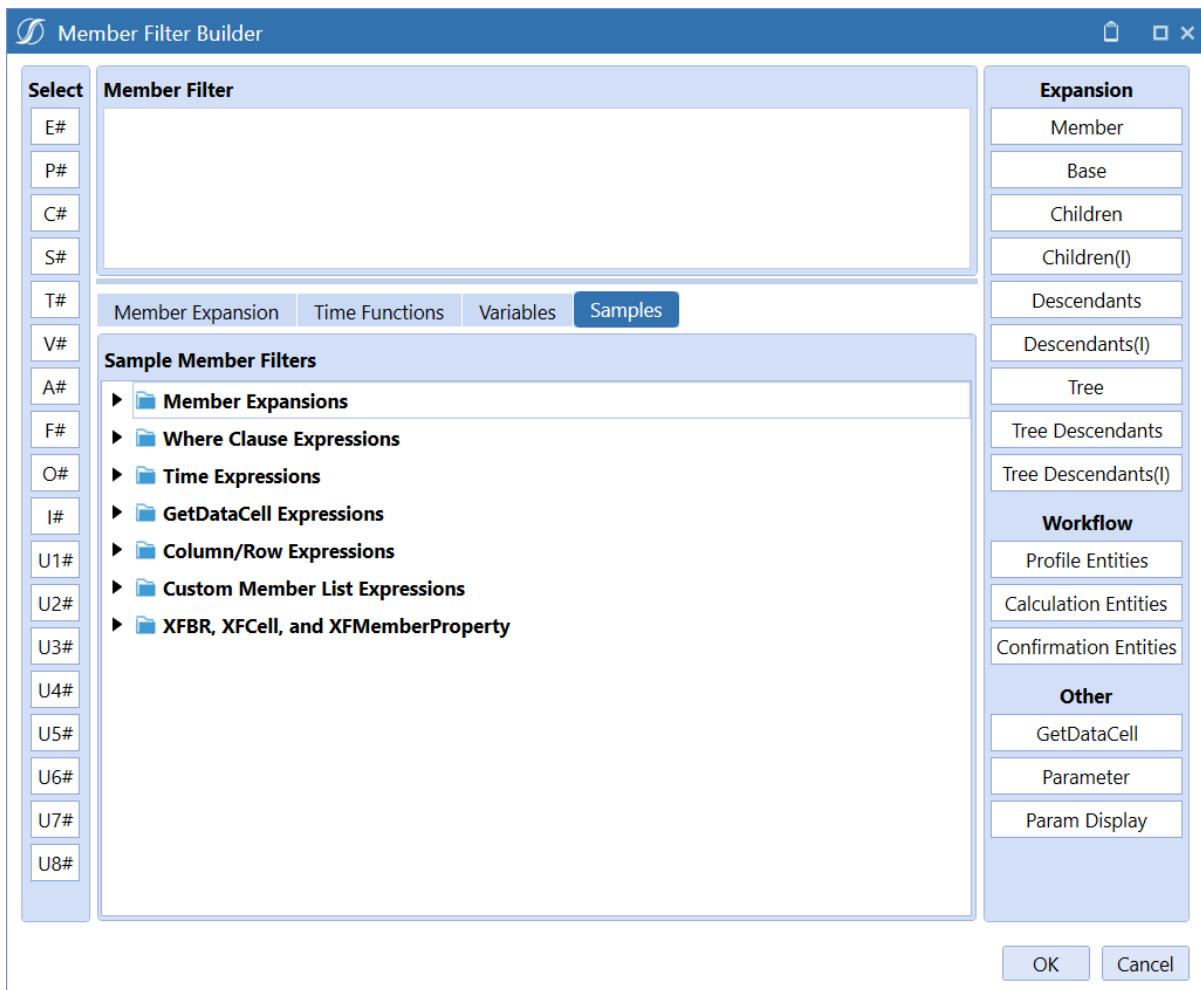
6. Click **Member Filter Builder** for the row to insert a calculation.



7. Delete any existing content in the Member Filter field.
8. Click **Samples**.

## Presenting Data With Books, Cube Views and Dashboards

---



9. Expand **Column/Row Expressions**.
10. Click **Sum of Rows (CVR)**. The GetDataCell syntax is generated and displays in Member Filter.
11. Update the row names to add the rows together. Rename Row1 to Cash and Row2 to AR.
12. Replace the name at the end of the expression to reflect the row name. The expression is complete.
13. Click **OK** and then click **Save**.
14. Click **Open Data Explorer** to run the cube view and verify the result.

### Troubleshoot GetDataCell Errors

These are some common errors when using GetDataCell expressions:

- Incorrect dimension: Make sure you have the correct dimension listed. If you are unsure, use the Member Filter Builder.
- Incorrect use of parentheses: Verify that you have the correct number of parentheses and that they are positioned correctly to avoid a script error.

### Column and Row Expressions

Cube views accept calculated columns and rows using the GetDataCell function and references to cube view column and row names. This is known as Column/Row math.

Write column and row expressions in the cube view row or column itself. They calculate at runtime.

These expressions require a reference to the specific cube view column or row name to return a data point.

Use column and row expressions if you need to perform math across columns and rows. A common use case is with variance columns. Column and row expressions are useful if the query is more complex or if the expression is easy to understand and maintain.

**NOTE:** Use one of the other expression types if you are using multiple member expansions, especially when performing math on rows.

### Create Cube View Column and Row Calculations

Cube views accept calculated columns or rows using the GetDataCell() function and references to cube view column or row names. This is also known as cube view column/row math.

These are based on the names given to a cube view column, for example "Col1" or "TimePeriods" or cube view row, for example "Row2" or "Accounts" by the creator of the cube view.

**NOTE:** Using the column or row names may be more useful than using member filters.

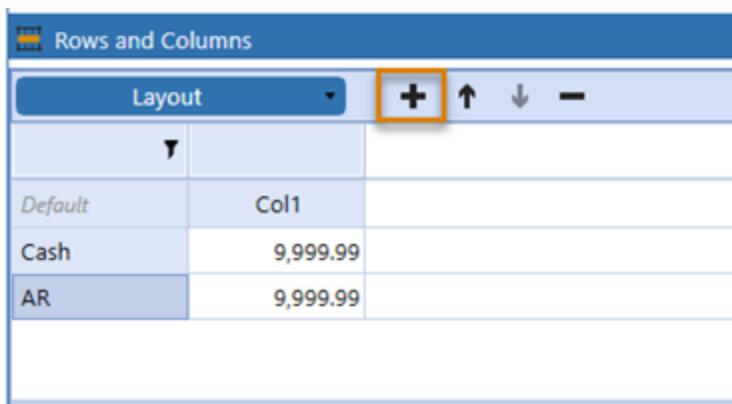
- Cube view columns: Uses the GetDataCell function to perform math on columns.
- Cube view rows: Uses the GetDataCell function to perform math on rows.

There are several variations on this method, depending on whether the expression refers to columns, rows, or a combination.

When naming a new calculation, do not include spaces or special characters that could make the column/row calculations read incorrectly.

### Create a CVC/CVR Calculation

1. Add a row or column to insert the calculation.



Rows and Columns	
Layout	
	▼
Default	Col1
Cash	9,999.99
AR	9,999.99

2. Rename the row relevant to your calculation. In this case, the is CurrAssets.

## Presenting Data With Books, Cube Views and Dashboards

---

The screenshot shows a 'Rows and Columns' interface with a 'Layout' tab. Below it is a table:

	Col1
Default	9,999.99
Cash	9,999.99
AR	9,999.99
CurrAssets	9,999.99

Below the table is a 'Member Filters' pane with tabs: Member Filters, Formatting, Data, Sharing, Column Overrides. The 'Member Filters' tab is selected. A row is selected in the table, and its name, 'CurrAssets', is highlighted with a yellow box in the 'Member Filters' pane. The pane also contains four levels of member filters:

- Row: CurrAssets
- Account: A#AR
- (Not Used): Level 2: Member Filter
- (Not Used): Level 3: Member Filter
- (Not Used): Level 4: Member Filter

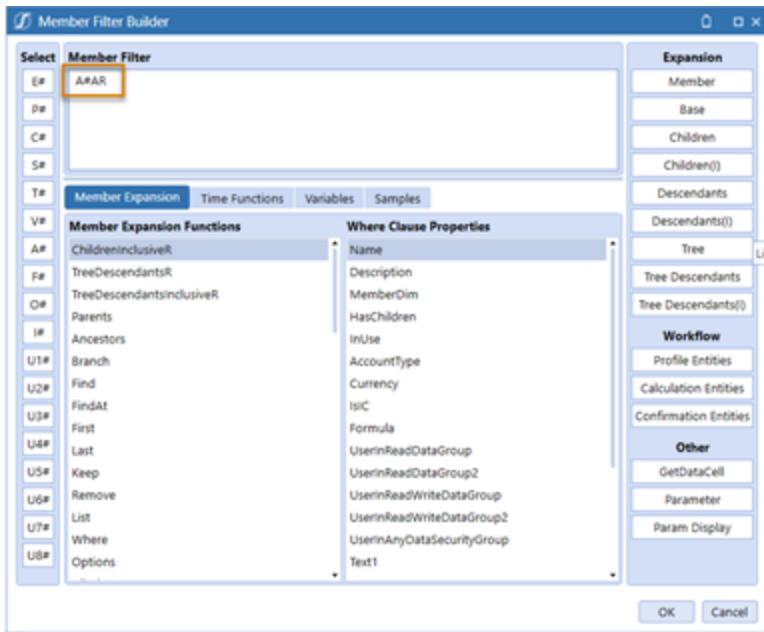
3. Open the Member Filter Builder for the row or column where you want to insert a CVC/CVR calculation.

The screenshot shows the same 'Member Filters' pane as before, but now the 'Level 4: Member Filter' entry has been deleted, indicated by a small delete icon in the top right corner of its row.

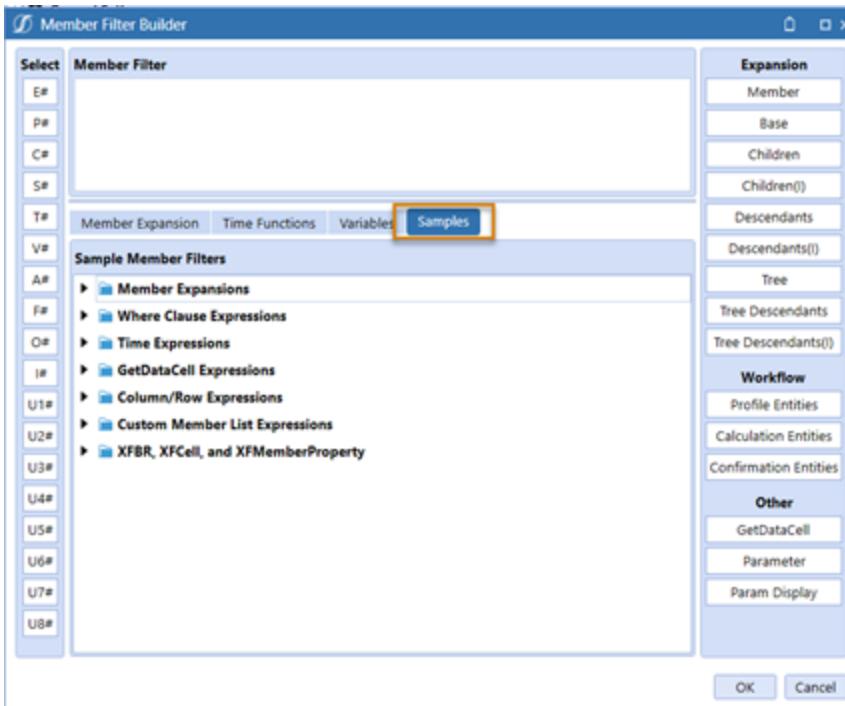
4. Delete what is in the Member Filter pane.

## Presenting Data With Books, Cube Views and Dashboards

---



5. Navigate to the Samples tab.



## Presenting Data With Books, Cube Views and Dashboards

---

6. Expand Column/Row Expressions.

The screenshot shows the 'Samples' tab selected in the top navigation bar. Under 'Sample Member Filters', the 'Column/Row Expressions' section is expanded, indicated by a minus sign icon. A yellow box highlights this section. Below it, a list of expressions is shown, with 'Sum of Rows (CVR)' also highlighted by a yellow box.

- ▶ Time Expressions
- ▶ GetDataCell Expressions
- ◀ Column/Row Expressions**
- Difference Between Columns (CVC)
- Column Math for Sum (CVC)
- Sum of Rows (CVR)
- Variance using Column and Index (CVC)
- Variance using Column and First Index (CVC)
- Row Variance (CVR)
- Row Variance % (CVR)
- Column Division (CVC)
- Column Better/Worse (CVC)
- Column Better/Worse % (CVC)
- Rows and Column Intersections (CVRC)
- Row and Column Intersection with Current (CVRC)

▶ Custom Member List Expressions

7. Double-click Sum of Rows (CVR). This will populate the GetDataCell expression that you need.

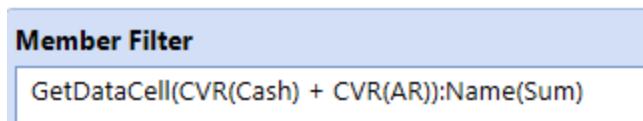
The screenshot shows the 'Samples' tab selected in the top navigation bar. Under 'Sample Member Filters', the 'Column/Row Expressions' section is expanded. The 'Sum of Rows (CVR)' item is selected and highlighted with a yellow box.

- ▶ Time Expressions
- ▶ GetDataCell Expressions
- ◀ Column/Row Expressions**
- Difference Between Columns (CVC)
- Column Math for Sum (CVC)
- Sum of Rows (CVR)**
- Variance using Column and Index (CVC)
- Variance using Column and First Index (CVC)
- Row Variance (CVR)
- Row Variance % (CVR)
- Column Division (CVC)
- Column Better/Worse (CVC)
- Column Better/Worse % (CVC)
- Rows and Column Intersections (CVRC)
- Row and Column Intersection with Current (CVRC)

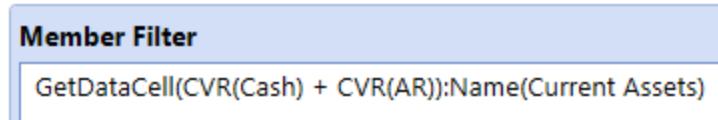
## Presenting Data With Books, Cube Views and Dashboards

---

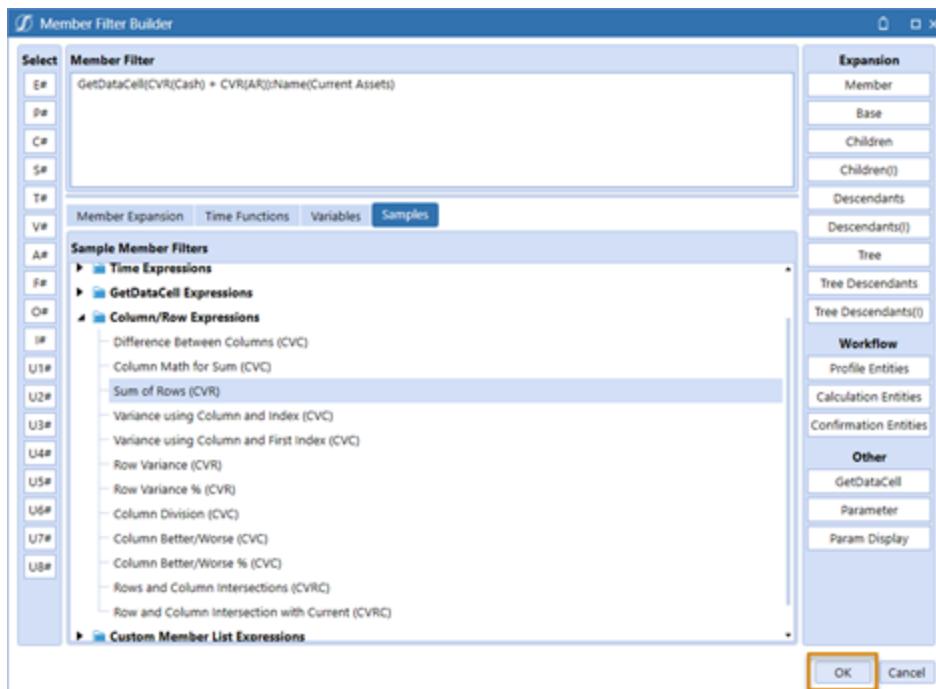
8. Update the row names to add the rows together. Rename Row1 to Cash and Row2 to AR because Cash and AR are row header names used in the cube view.



9. Replace the name at the end of the expression to display Current Assets.



10. Click OK.



11. Click Save.

12. Run the cube view.

## CVC: Cube View Column Calculations

The example below shows a Column Math example of the difference between columns (CVC).

The syntax is `GetDataCell(CVC(SomeColumnName) - CVC(SomeOtherColumnName)):Name(Header Name)`

The example of simple member math includes the `:Name()` function typically applied to a column in a cube view:

```
GetDataCell(CVC(Col1) - CVC(Col2)):Name(Variance)  
GetDataCell(CVC(Col1) + 1):Name(Column Plus One)  
GetDataCell(CVC(Col1) * (-1)):Name(Column with Sign Flipped)
```

## CVR: Cube View Row Calculations

The example below shows a Row Math example of Sum of Rows (CVR).

The syntax is similar, but instead of CVC, a calculated row uses CVR for retrieving the value of a row in a formula.

An example of the syntax: `GetDataCell(CVR(SomeRowName) + CVR(SomeOtherRowName)):Name(HeaderName)`

**NOTE:** If a column name is numeric (for example, 500), single quotes are required when specifying the row name. Square brackets are allowed, but not required. For example: `GetDataCell(CVR('123') - CVR(['4,567'])):Name(Difference)`

When you need a column and row index: Col1 has a member filter of S#Actual, S#Budget, meaning it will return two columns. In this case, a variance between Actual and Budget scenarios can be shown like this: `GetDataCell(CVC(Col1, 1) - CVC(Col1, 2)):Name(Variance)`.

Alternatives: `GetDataCell(CVC(Col1, First) - CVC(Col1, 2)):Name(Variance)` `GetDataCell(CVC(Col1) - CVC(Col1, 2)):Name(Variance)`

## Column Math Example with Division and Other Advanced Functions

Member Expansions may require some more intricate formulas. This may require using indexes or a CVRC expression. Use functions like Divide to avoid divide by zero situations.

```
GetDataCell(Variance(CVR(Col1,2), CVR(Col1,1)):Name(Variance)GetDataCell  
(VariancePercent(CVR(Col1,2), CVR(Col1,1)):Name(Variance %) GetDataCell(Divide(CVC  
(Col3), CVC(Col2))):Name(Ratio) GetDataCell(BWDiff(CVC(Col1), CVC(Col2))):Name  
(BetterWorse Difference)GetDataCell(BWPercent(CVC(Col1), CVC(Col2))):Name(BetterWorse  
%)
```

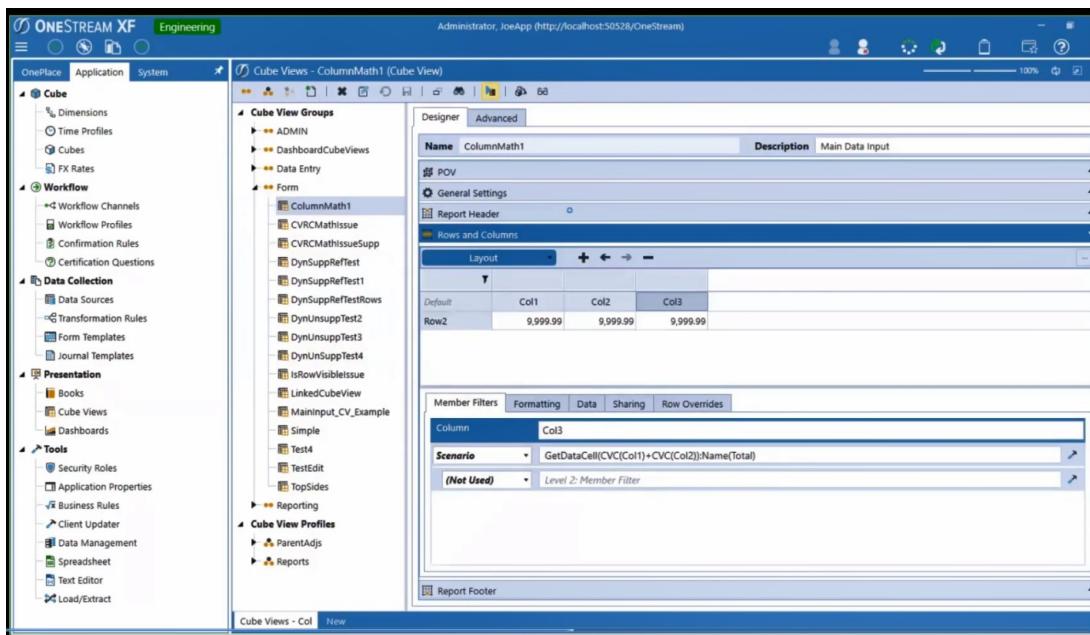
## Create Advanced Column Row Expressions

Use an index or a CVRC expression to create more intricate formulas. The Member Filter Builder has complex options in the Samples tab.

## Reference Hidden Columns with Cube View Column Math

Use the **CVMathOnly** option to perform math on columns when the columns are not visible.

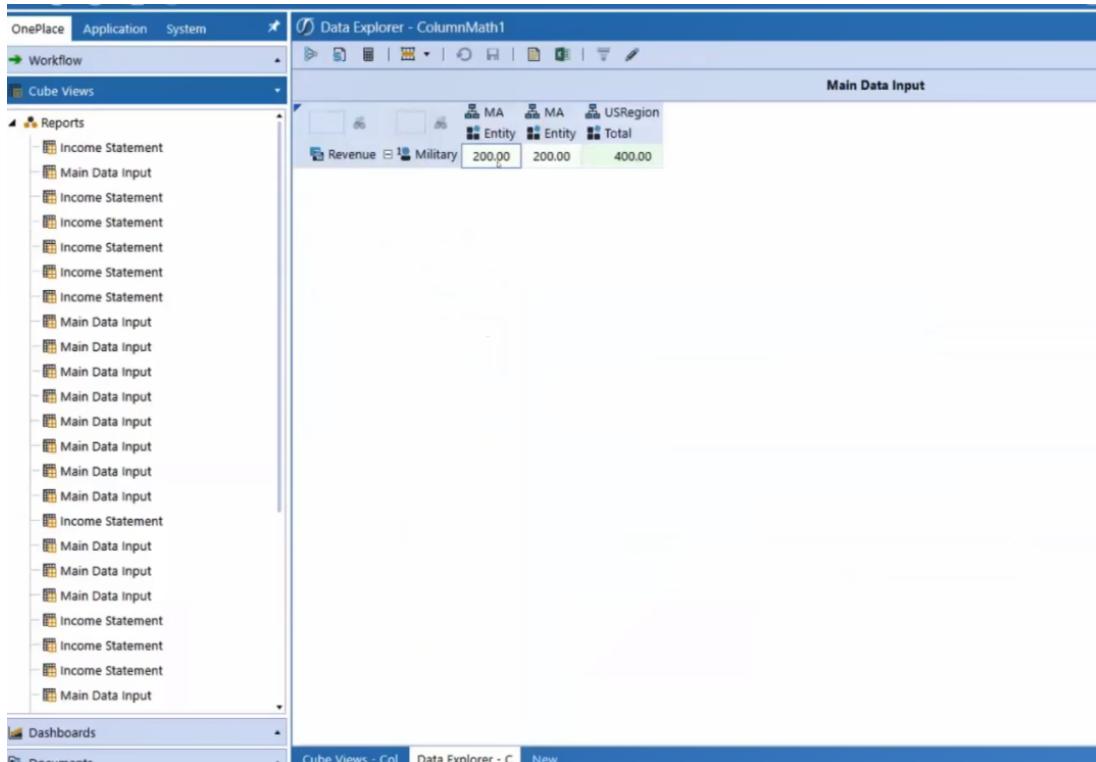
1. From the Application tab, under Presentation, click **Cube Views**.
2. Expand Cube View Groups, expand a specific cube view group and select a cube view.
3. Click **Rows and Columns**. In the example below there are three columns, where Column 3 has the `GetDataCell()` function applied for the total of Column 1 and Column 2.



## Presenting Data With Books, Cube Views and Dashboards

---

4. Click **Open Data Explorer** to generate the Cube View and see the data. In the example below you see that it is adding the value 200.00 from Column 1 to the value 200.00 from Column 2 to generate the total of 400.00 in Column 3.



5. Click **Edit**.
6. Select **Column 1** and at the bottom of the window click the **Formatting** tab.
7. In Header Format, click the ellipsis. The Header Format window opens.
8. Click **Format**.
9. In the General section, find **isColumnVisible** and select **CVMathOnly**.

## Dynamic Member Formulas and Cube Views

Dynamic calculations enhance the consolidation performance because the amount is calculated when requested for display and is not written to the database.

Use dynamic calculations in cube views and business rules.

- There is no performance impact on consolidation times.
- You can make dynamic calculations drillable so users can see the source for the calculated result when drilling down.
- Use these calculations for ratios or variances.

In the dimension library, write dynamic calculations on specific members. They will calculate at runtime.

Dynamic calculations have a syntax similar to GetDataCell expressions because they also require reference to specific dimension members to return a data point.

**NOTE:** A difference between dynamic calculations and GetDataCell expressions is that dynamic calculations require proper business rule logic.

Use dynamic calculations if you need a repository of commonly used calculations which allows for consistency and reduces maintenance.

Create common calculations in the UD8 dimension. It allows you to have the most flexibility with your rule syntax.

Use another calculation type if the calculation is only used on a specific cube view because you may not want to create a member in the dimension library. Also, consider if you will grant a report builder access to the dimension library to view, create, and edit these calculations.

## Create Dynamic Calculations

Dynamic calculations enhance the consolidation performance because the amount is calculated when requested for display and is not written to the database.

Account, Flow, and all UDs can contain dynamically calculated members.

- There is no performance impact on consolidation times.
- You can make dynamic calculations drillable so users can see the formulas running.
- These calculations can be used for ratios or variances.

Two properties must be set to make the calculation dynamic: Account Type and Formula Type. Ensure that the Formula Type is set to DynamicCalc. If using on an Account dimension type, both the Account type and Formula Type need to be set to DynamicCalc.

## Presenting Data With Books, Cube Views and Dashboards

---

Account Type	DynamicCalc
Formula Type	DynamicCalc

Here are some common examples of how dynamic calculations are used:

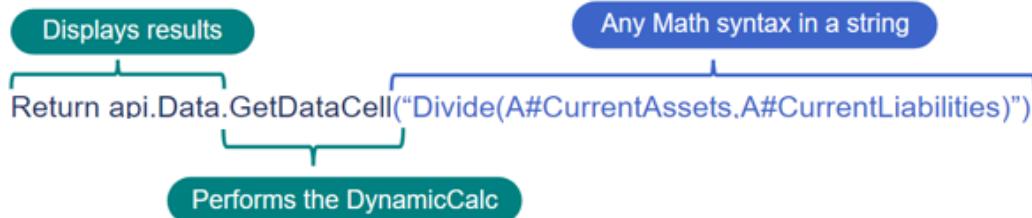
- `Api.Data.Calculate("Target = A + B")`
- `Api.data.calculate("A#InsuranceExp = A#Payroll * A#InsuranceCostPcent")`
- `Api.Data.GetDataCell("A + B")`
- `Return Api.data.GetDataCell("A#GrossSales – A#ReturnsAndAllowances")`

### Calculated Members and Example

In the Dimension Library, there are likely a few members that will be calculated using a member formula. Dynamically calculated members are members that use the GetDataCell function to return values based on math when they are queried.

These members require that a Formula Type of DynamicCalc be defined before writing a member formula.

**NOTE:** If this property is not defined, the member will return an error and the rule will not compile.



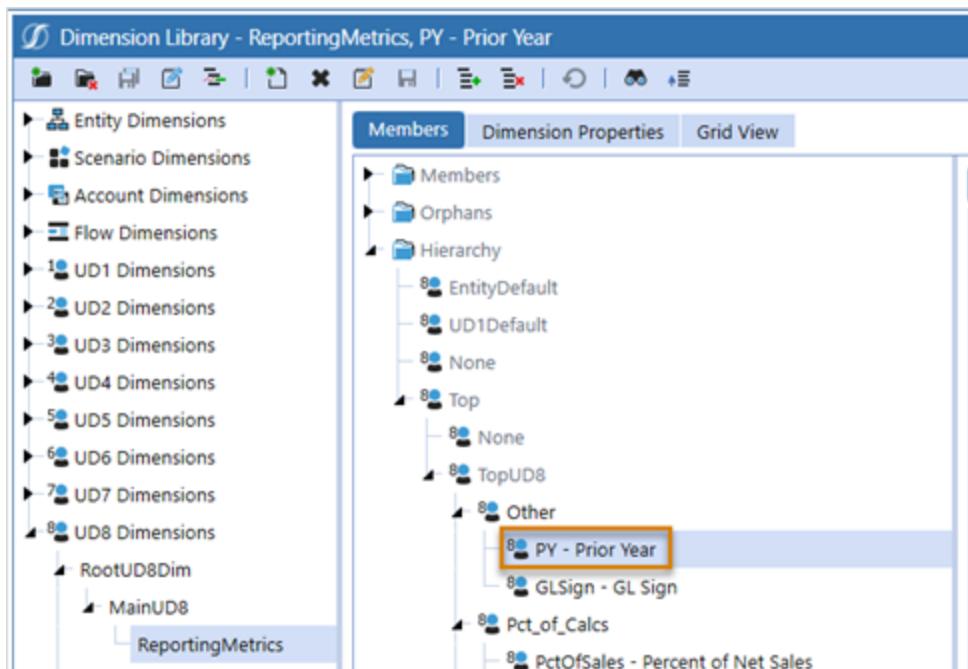
- **Display Results:** This section tells the system how to display the results of the function.
- **Performs the DynamicCalc:** This section tells the system to perform the DynamicCalc using the GetDataCell function.
- **Any Math syntax in a string:** This tells the system what math you want it to perform.

### Define a Dynamic Calculation

Before you add a dynamic calculation to a cube view, you must define it or write the simple dynamic calculation.

In this example, we are going to navigate to a UD8 member that you want to update to define the simple dynamic calculation.

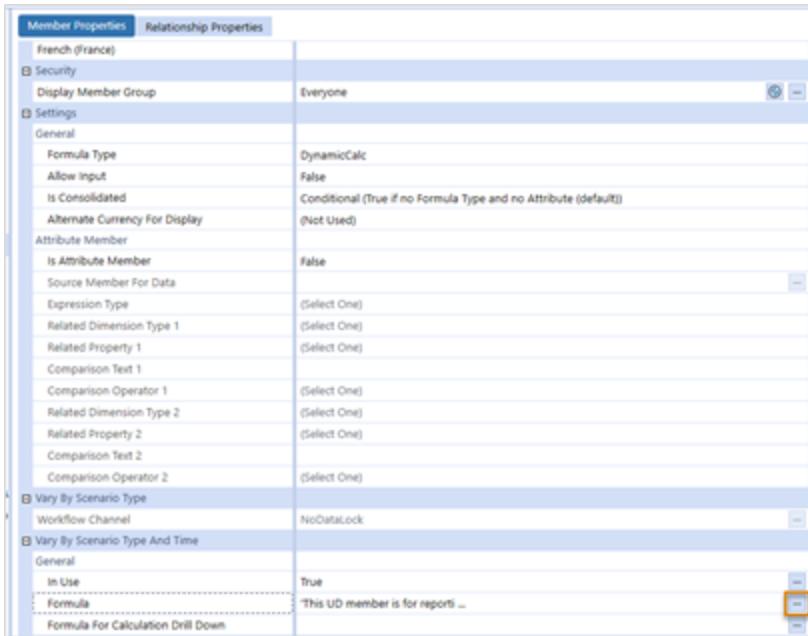
1. Open the UD8 member.



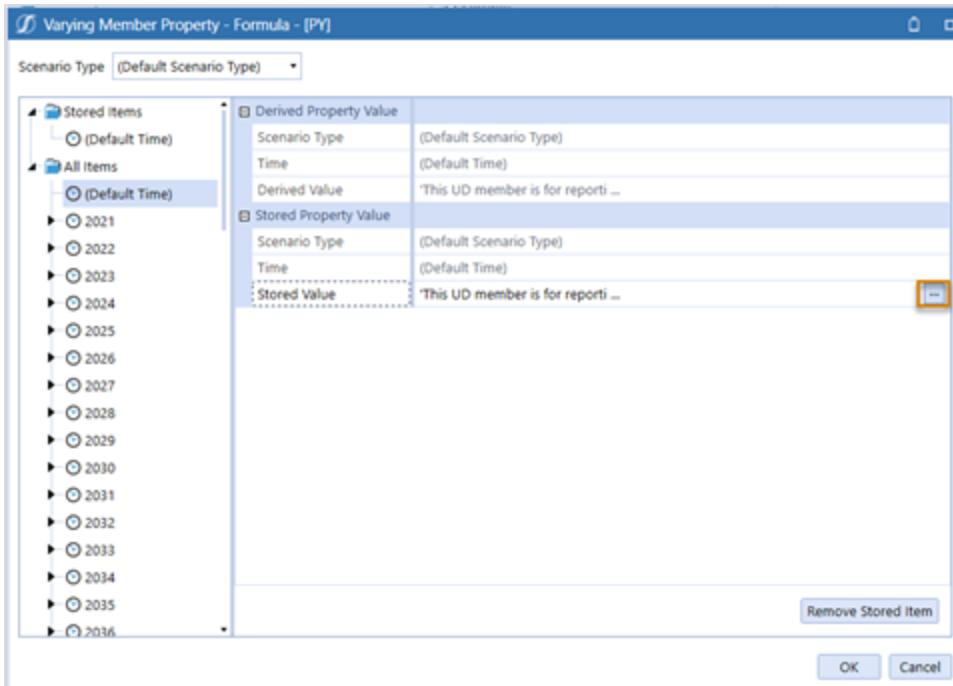
2. Open the formula editor for the member and ensure that the formula type is set to DynamicCalc.

## Presenting Data With Books, Cube Views and Dashboards

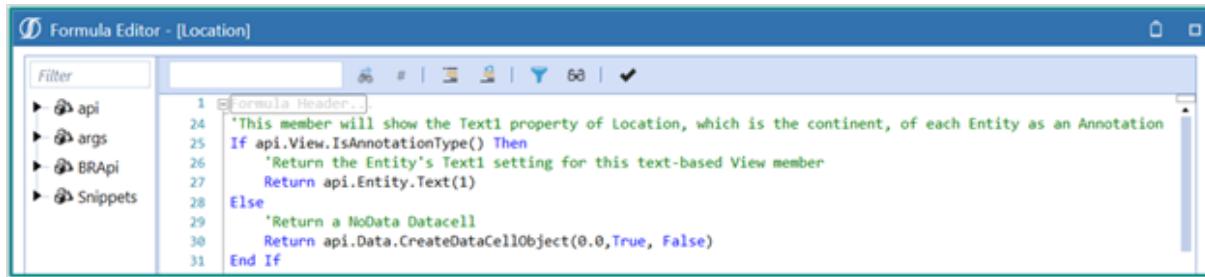
---



3. Open the stored value.



Now you can write a dynamic calculation. In this example, the formula will display an entity's Text1 value if an Annotation type View member is referenced.



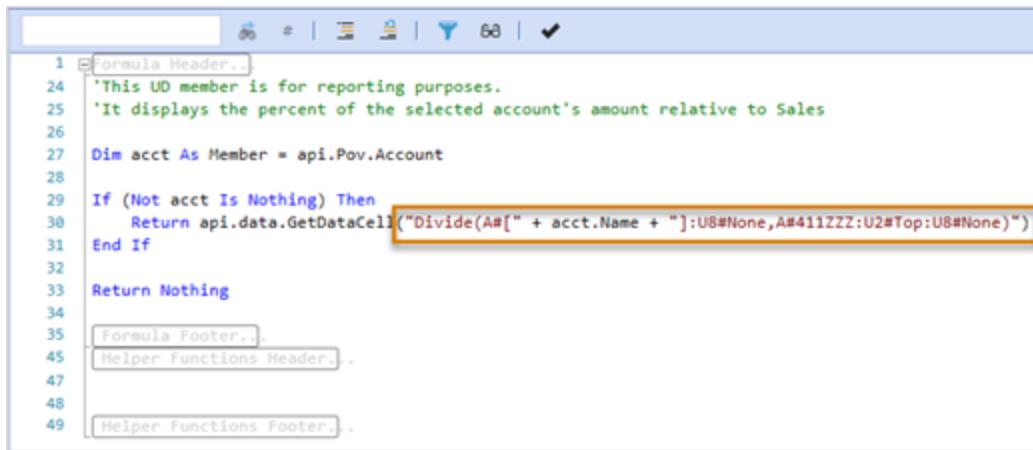
## Write a Dynamic Calculation

You can write a simple dynamic calculation before adding it to a cube view.

1. Open the formula editor for the member that you want to update and navigate to **Snippets > Presentation Helpers > Reporting**.
2. Select the dynamic calculation you want to use.
3. Copy and paste the syntax.



4. Edit the dimensions of the pasted syntax to make sure they are correct.



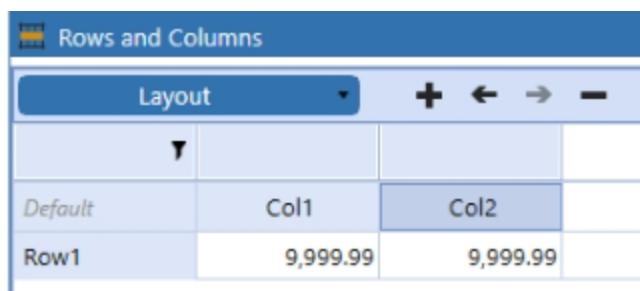
A screenshot of a code editor window titled "Formula Member...". The code is written in VBA-like syntax. A specific line of code, "Return api.data.GetDataCell("Divide(A#" + acct.Name + "] : U8#None, A#411ZZZ:U2#Top:U8#None)")", is highlighted with a yellow box. The code includes comments: "'This UD member is for reporting purposes.'", "'It displays the percent of the selected account's amount relative to Sales'", and "Dim acct As Member = api.Pov.Account". There are also sections labeled "Formula Header...", "Formula Footer...", "Helper Functions Header...", and "Helper Functions Footer...".

5. Double-check your syntax.
6. Click **OK**.
7. Close the Formula Member Property.
8. Save the member.

## Add a Dynamic Calculation to a Cube View

After you define and write a dynamic calculation, you can add the calculation to a cube view.

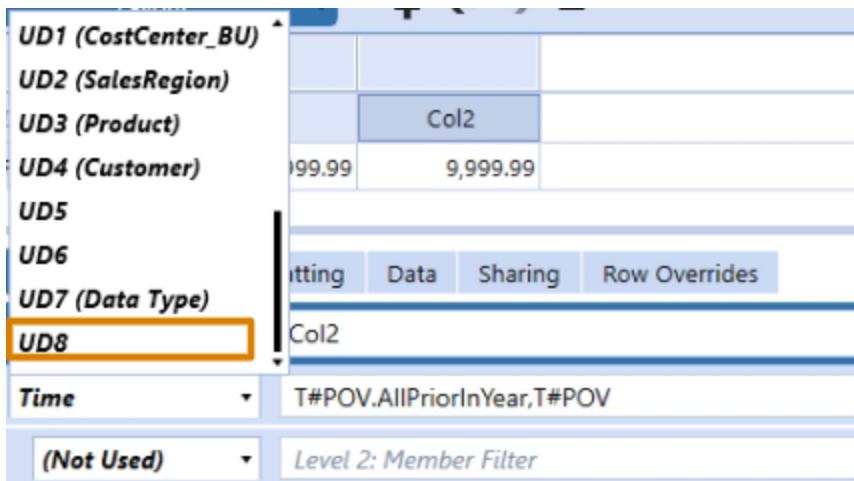
1. In OneStream, go to **Application > Presentation > Cube Views**.
2. Select a cube view group and then a cube view.
3. Create a new column. You will add a dynamic calculation to the second column. All the selections and properties of Col1 are carried over to the new Col2. Our reporting dynamic calculations are located in UD8. Therefore, the dimension will need to change.



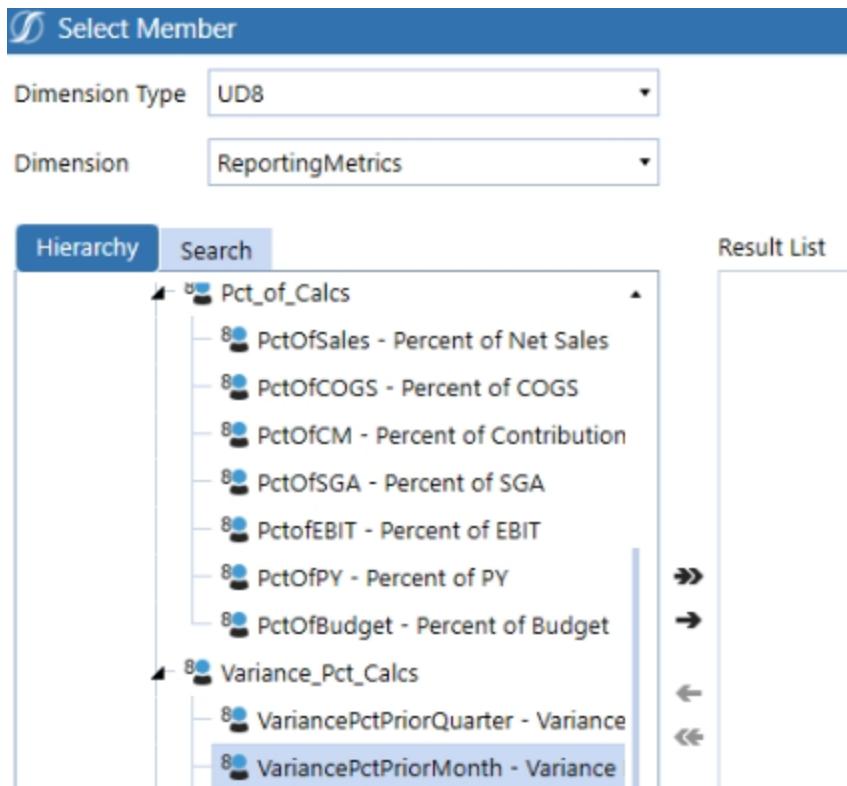
A screenshot of the "Rows and Columns" dialog in OneStream. The title bar says "Rows and Columns". Below it is a toolbar with "Layout" and various icons. The main area shows a 2x2 grid of cells. The first row has two columns: "Default" and "Col1". The second row has two columns: "Row1" and "Col2". The "Col2" cell is highlighted with a blue background. The "Layout" toolbar icon is also highlighted with a blue box.

Default	Col1	Col2
Row1	9,999.99	9,999.99

4. Open the column dimension drop-down and select UD8.



5. Click the Member Filter Builder.
6. Delete the current filter.
7. Select the U8# dimension token.
8. Select VariancePctPriorMonth and move the selection over.

A screenshot of a software interface titled "Select Member". At the top, there are two dropdown menus: "Dimension Type" set to "UD8" and "Dimension" set to "ReportingMetrics". Below these are two tabs: "Hierarchy" (selected) and "Search". The "Hierarchy" tab displays a tree structure of reporting metrics under two main categories: "Pct\_of\_Calcs" and "Variance\_Pct\_Calcs". Under "Pct\_of\_Calcs", there are six items: "PctOfSales - Percent of Net Sales", "PctOfCOGS - Percent of COGS", "PctOfCM - Percent of Contribution", "PctOfSGA - Percent of SGA", "PctofEBIT - Percent of EBIT", and "PctOfPY - Percent of PY". Under "Variance\_Pct\_Calcs", there are two items: "VariancePctPriorQuarter - Variance" and "VariancePctPriorMonth - Variance". The "VariancePctPriorMonth - Variance" item is highlighted with a blue selection bar at the bottom.

9. Select the VariancePriorMonth member and move the selection over.
10. Click **OK**.
11. Click **OK** to confirm the Member Filter selection.
12. Save your changes.
13. Run the cube view as a Data Explorer Grid. The cube view renders and displays the dynamic calculations in the two far-right columns.

February 2023	March 2023	April 2023	May 2023	June 2023	June 2023	Variance Pct to Prior Month	Variance to Prior Month
Top	Top	Top	Top	Top	Top		
335,697.13	486,871.11	633,992.29	780,653.83	929,445.93		0.19	148,792.10
201,680.40	281,831.08	355,957.90	427,900.89	500,390.22		0.17	72,489.33
199,017.42	274,508.15	342,498.23	406,948.51	470,464.01		0.16	63,515.50
337,868.40	488,569.98	634,572.77	779,481.60	925,741.91		0.19	146,260.32
335,697.13	486,871.11	633,992.29	780,653.83	929,445.93		0.19	148,792.10

## Override Complex Calculations

When you apply dynamic calculations to both rows and columns the row formula takes precedence, whether it is a dynamic calculation member or cube view math.

Rows and columns contain several override properties by row:

- When you use the row override in the column it will use the column formulas for the specified rows.
- When you use the column override in the rows it will use the formulas for the specified columns.

### Apply Overrides

1. In the cube view, select a column.
2. Select the Row Overrides tab to apply row overrides in the column.
3. Define the row range. Enter a row name or apply a range of row names to override the results for the specific rows in the column of the cube view.
4. Enter the member filter to override the value for the row range. This defines what dimension member data displays or specifies complex calculations for the row range.
5. Set the format for the row range, which includes selecting fonts, text colors, cell background color, grid lines, cell borders, and so on.
6. Select a value from the list parameter drop down list.

### Row/Column Overrides

Four overrides can be applied against different columns or rows and can be used to change the display for a given column or row. In this example, two columns for the Flash Scenario (Col1) vs. Budget (Col2) is needed. If Row1 is being used by the Net Sales Account (e.g. A#60999), and it needs to be compared to the Flash Net Sales which is a different account (e.g. A#FlashNetSales), then a column override would be used.

#### Row/Column Range

Type a column name to display a different value in the Cube View. In this example, Col1 would be typed in order to show a different value for Flash Scenario Column.

#### Member Filter

Type the Member Filter to override the value for the columns or rows. In this example, type A#FlashNetSales to override the results from A#60999.

The override results will display the value from A#FlashNetSales in Col1 for the Flash Scenario.

#### Cell Format

See Cube View Formatting.

#### List Parameter

See List Parameter under Data earlier in this section.

## Workspaces

A workspace is a framework for building software using software, creating a robust environment for developing products on the platform. It simplifies the development process and extends development capabilities for solution creators.

Workspaces store maintenance units and facilitate community development by providing an isolated environment for developers to segregate and organize solution objects. Maintenance units are stored, created, and maintained in workspaces, which vary by dashboard project need and application.

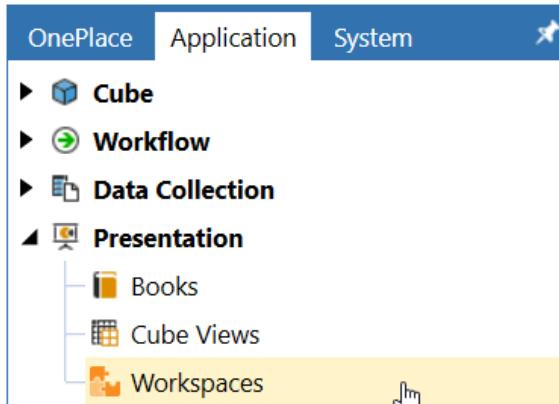
## Benefits

Workspaces provide an isolated environment in which solution developers and creators can develop multiple solutions to solve complex business processes. Workspaces provide the following benefits:

1. Isolation between workspaces, which allows developers to work on the same solution or dashboard in a sandbox-like environment.
2. Greater flexibility among developers and other team members when testing, making changes, and planning.
3. Maintenance Units, along with their objects, can have the same names in separate workspaces and do not need to be renamed. This reduces the likelihood of naming conflicts especially when importing and exporting objects from other applications or sources.
4. You can selectively share workspace objects such as embedded dashboards, parameters, file resources, and string resources with other workspaces. This lets you reuse objects rather than copying them.
5. Workspace objects can have the same names in different workspaces.
6. Sets the foundation for future functionality and ongoing development.
7. Product packaging mechanism for creating, deploying, and migrating solutions.

## The Workspaces Page

The Workspaces page, found on the Application tab and the System tab, is used to create application and system workspaces. The pages look similar between the two tabs.



## Security Roles

The following security roles and locations are necessary when creating workspaces.

Role	Location	Property
Application Security	<b>Application &gt; Tools &gt; Security Roles</b>	ManageApplicationWorkspaces
Application User Interface	<b>Application &gt; Tools &gt; Security Roles</b>	WorkspaceAdminPage
System Security	<b>System &gt; Administration &gt; Security&gt;System Security Roles</b>	ManageSystemWorkspaces
System User Interface	<b>System &gt; Administration&gt; Security &gt; System Security Roles</b>	SystemWorkspaceAdminPage
All Roles	<b>Application &gt; Tools &gt; Load/Extract</b>	Application Workspaces
	<b>System &gt; Tools &gt; Load/Extract</b>	System Workspaces

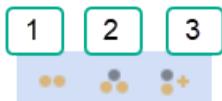
Cube views created prior to version 8.0 are accessible through the Default Maintenance Unit, found within the Cube Views page and the Default Workspace.

The Access Group and Maintenance Group for these cube view groups can be edited if you are assigned to the ManageCubeViews Application security role. If cube views are created within the Workspaces page outside of the Default Maintenance Unit, you cannot edit the security of these cube view groups.

With the Application security role, ManageApplicationWorkspaces provides full rights to edit anywhere outside the Default workspace and Default Maintenance Unit. To edit any assemblies, you must be a part of the AdministerApplicationWorkspaceAssemblies security role. This applies to all dashboard groups within the generic default workspace and all non-default workspaces. Maintenance Unit security determines access.

### Toolbar Buttons

Several buttons on the Workspaces toolbar work with both dashboards and cube views, depending on whether you are working with cube view groups and profiles or dashboard groups and profiles. Tooltips provide helpful hints and are accessed by hovering the cursor over the button.



1. **Create Group:** Create a new cube view group (if cursor is on the Cube View Groups icon) or dashboard group (if on the Dashboard Groups text of a maintenance unit)
2. **Create Profile:** Create a new cube view profile (if you have selected a cube view profile) or dashboard profile (if you have selected a dashboard profile).
3. **Manage Profile Members:** Displays the Profile Builder dialog if you've selected either a cube view profile or a dashboard profile. If the cube view or dashboard is not in the Default workspace, a tooltip will appear displaying the workspace prefix when you hover the cursor over the group name.
4. **Create Cube View:** Create a new cube view when viewing cube view groups.
  
5. **Delete Selected Item:** Delete selected component item. You can select multiple cube views and cube view groups. You can only delete empty Cube View Groups. Multiple cube views can only be selected and deleted if they are in the same Cube View Group.
6. **Rename Selected Item:** Opens the Rename dialog box so that you can rename the selected item.
7. **Copy:** Makes a copy of the selected items. Select more than one item of the same type by pressing and holding the Ctrl button and clicking each item with the mouse. You can also first select the items to copy then right-click to open a context menu. You can only copy items of the same type.
8. **Paste:** Pastes copied items into the selected destination. "\_Copy" is appended to the name of the newly pasted item.

9. **Open Data Explorer**: View the data explorer grid of the selected cube view. Results will display in a separate tab.
10. **Show Objects That Reference The Selected Item**: Opens the Object Usage Viewer and lists all objects that reference the selected item.
11. **Object Lookup**: Opens the Object Lookup dialog box letting you search for any object in the application.

## Workspaces Properties

Workspaces are defined and function according to the following conventions.

### Naming Conventions

All maintenance unit objects can have the same name as long as they are in separate workspaces.

### Default Workspace

A system generated workspace where you can create solutions. All existing maintenance units prior to version 7.3 were moved to the Default workspace.

- The Default workspace is always shared and is accessible by any other workspace, even if not explicitly added to the Shared Workspaces list.
- Referencing parameters do not consider workspaces and backward compatibility is enabled.

## Properties

The following properties help to define a workspace.

### Sharing

Sharing allows developers to selectively share dashboard items such as dashboards, parameters, components, and adapters across workspaces when necessary. You should implement all related items in the same workspace to minimize sharing of items from other workspaces. Sharing items across workspaces provides opportunities for re-use of items rather than copying them.

### Is Shareable Workspace

If this property is set to True, other workspaces can reference objects in the current workspace.

**NOTE:** The Default workspace is shareable by default and cannot be changed.

The Default workspace is predefined by OneStream and the Is Shareable Workspaceproperty is always set to True. All other workspaces can reference dashboards and parameters implemented in the Default workspace.

The Default workspace also behaves as if all other workspaces are part of its Shared Workspace Names list. Therefore, any dashboard or parameter defined in another shareable workspace can be referenced by items in the Default workspace. This approach simplifies backward compatibility since new workspaces can access all items that previously existed in the Default workspace. However, this requires an implementor to avoid using similar names for items in the Default and custom workspaces.

Data management sequences and extensible documents are currently implemented outside of workspaces, but they can reference parameters defined in any shareable workspace. For those items, the search order is determined as if the item, for example the cube view, was implemented in the Default workspace.

### Shared Workspace Names

This property indicates the workspace names for use in the current workspace. This is a comma-separated (no spaces) list of names that can be used to find embedded dashboards, parameters, file resources, string resources, and so forth while processing a dashboard in this workspace.

Referenced workspaces must have the Is Shareable Workspace property set to **True**. The search order for a requested item name is the current workspace followed by the Default workspace, then followed by the list of Shared Workspace Names in the order they were entered.

### Assemblies

#### Namespace Prefix

Specify different names used for referencing assemblies in a workspace.

#### Imports Namespace 1-8

Corresponds to the \_ImportsNamespace1 placeholder value within the import statements in code.

#### Workspace Assembly Service

Type a name for the workspace assembly service factory used to process the dynamic creation of dashboards within this workspace. This property must be configured for the service type assembly files to be used as intended. The syntax for this property is AssemblyName.AssemblyFileName.

### Text

#### Text 1-8

Create string values.

### Search Order

Sharing items across workspaces requires a defined search order when the same item name is used more than once. The following is the order of operations:

1. Search the workspace you are currently working in.
2. Search the Default workspace.
3. Search the Shared Workspace Names list.

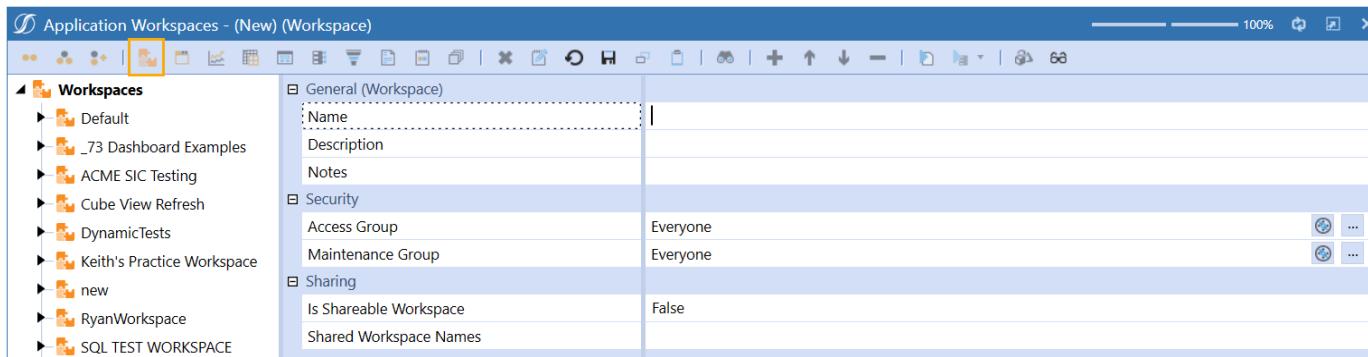
**NOTE:** If all items are defined in the Default workspace, search order is irrelevant.

### Workspaces Setup

Maintenance units created prior to version 7.4 can be found in the Default workspace.

To create a new workspace:

1. In OneStream, on the **Application** tab, click **Workspaces** under the **Presentation** section.
2. Select **Workspaces**.
3. Click **Create Workspace** and name the new workspace.



### Workspace Assemblies

Assemblies are integrated into workspaces to give you the freedom to write logic exactly where you need it. Here, you can write inline business rules using the Assemblies node within a given maintenance unit.

This applies to dashboard--specific business rules only and consists of the following types:

- Cube View Extender
- XFBRString
- Dashboard DataSet
- Dashboard Extender
- Spreadsheet

Workspace assemblies cater to the OneStream developer community for those who build solutions and create dashboards. They are similar to the Visual Studio product. Developers can add folders and within those folders, they can create any number of C# or Visual Basic files depending on the type of workspace. These files are then compiled into a single assembly.

The benefits of workspace assemblies extend to OneStream developers, customers, and partners. There are also a variety of service types that assist with the development of Solutions.

Because the power and flexibility this feature extends to those who use assemblies, it is important to:

- Inform developers on how to take advantage of assemblies
- Recognize evolving recommendations on using assemblies
- Highlight areas of assemblies of use to the broader community

## OneStream Developers

Developers include:

- **MarketPlace Developers:** Engineers creating solutions that reside within the MarketPlace for customers and partners supported by OneStream.
- **Solution Exchange Developers:** Anyone in the OneStream community who designs solutions featured in PartnerPlace or OpenPlace.
- **PreSales:** Sales engineers who create demonstrations and product proofs of concept.
- **Consultants:** Product implementers building custom solutions for clients.

- **Advanced Application Solutions:** Technical team building advanced solutions for customers.

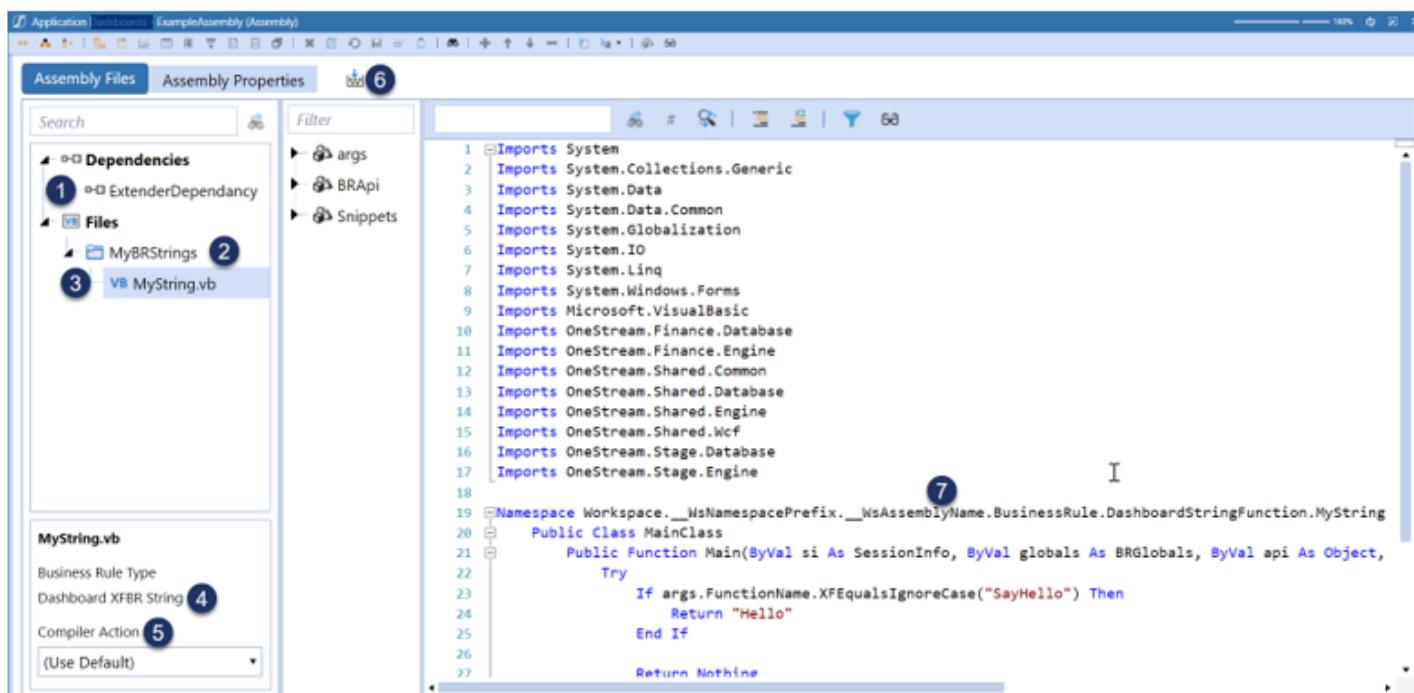
## Assembly Encryption

Assembly files can be encrypted. You must be a member of the EncryptBusinessRules security role. Within workspaces assemblies, you can create rules using either C# or Visual Basic. Right-click on the file name and select Encrypt. You will see the Encrypt File dialog box asking you to create a password using the same legacy rules. You can also decrypt files using the same method.

## Get Started With Assemblies

Assemblies are structured similarly to Visual Studio where you can build assemblies using files and folders. This workspace page is also similar to the Business Rules page in OneStream, allowing you to quickly create assemblies as needed.

The following image of a workspaces page shows the various components you will need to build an assembly.



1. **Dependencies:** Reference other assemblies from different workspaces. Right-click on **Dependencies** to create.
2. **Folders:** Organize assembly files.
3. **Files:** Written rules for the assemblies.
4. **Business Rule Type:** Type of business rule or service type. Decided upon during creation and cannot be changed after the file is created. This item will say Not Used if you have selected a service type rather than a business rule type.
5. **Compiler Action:** Action to perform on the file. Can be Use Default, Disabled, or Compile (this is the same as Use Default). If you select Disabled, you cannot compile the assembly to verify syntax. This value can be changed after creating the assembly file.
6. **Compile Icon:** Compiles all files in the assembly when clicked.
7. **NameSpace:** The NameSpace of any given file will look different than what you would see on the Business Rules page. The filename, in this example, MyString, is the last item in the list.

## Create an Assembly

Assemblies work similarly to Visual Studio Code, where there is a project with folders, each containing dependencies, such as assemblies and business rules. Assemblies let developers add references to known packages that are part of the standard server install, such as OpenXML. In the OneStream application, assemblies are located within Maintenance Units.

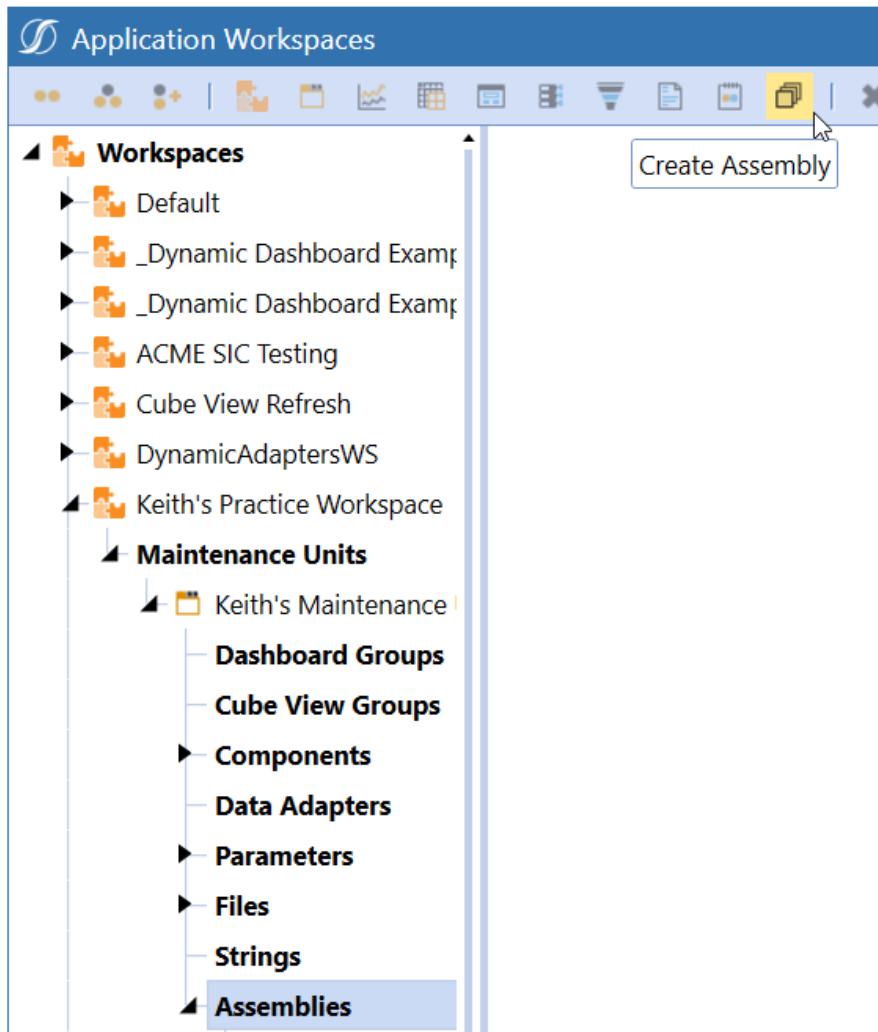
**NOTE:** If you need to reference an assembly from a business rule, use the following syntax: WS\ followed by the long workspace assembly name. For example, WS\Workspace.wsName.wsAssemblyName.

Follow these steps to create an assembly. In this example, you will use Visual Basic.

1. On the Application tab, click **Workspaces**.
2. Expand **Workspaces** and locate the workspace.
3. Expand the workspace, then expand **Maintenance Units**.
4. Expand the appropriate maintenance unit and click the **Assemblies** label.

You will not see this in the default maintenance unit, which is located in the Default workspace. This maintenance unit only stores cube view groups.

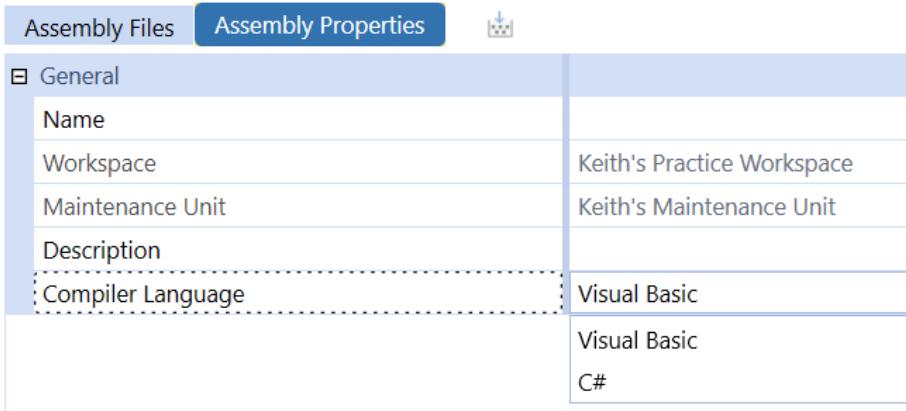
5. Click **Create Assembly** on the toolbar. Assemblies can be Visual Basic or C#.



6. On the Assembly Properties page, click the drop-down arrow in the **Compiler Language** field and select the type of assembly. It can be either Visual Basic or C#.

## Presenting Data With Books, Cube Views and Dashboards

---



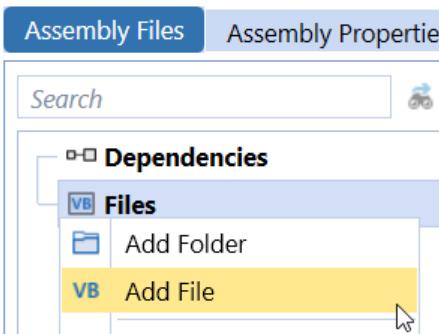
7. Give the assembly a name.
8. Optional. If you need to delete an assembly, while on the Assembly Properties page, click **Delete**.
9. Click **Save**.

You have now created an assembly.

### Create an Assembly File

The assembly file is where you write the logic. You can choose to write a Business Rule type or a rule of your own (Not Used). The Not Used option provides great flexibility and is a good choice for most developers. Additionally, some performance gains may be realized with this choice.

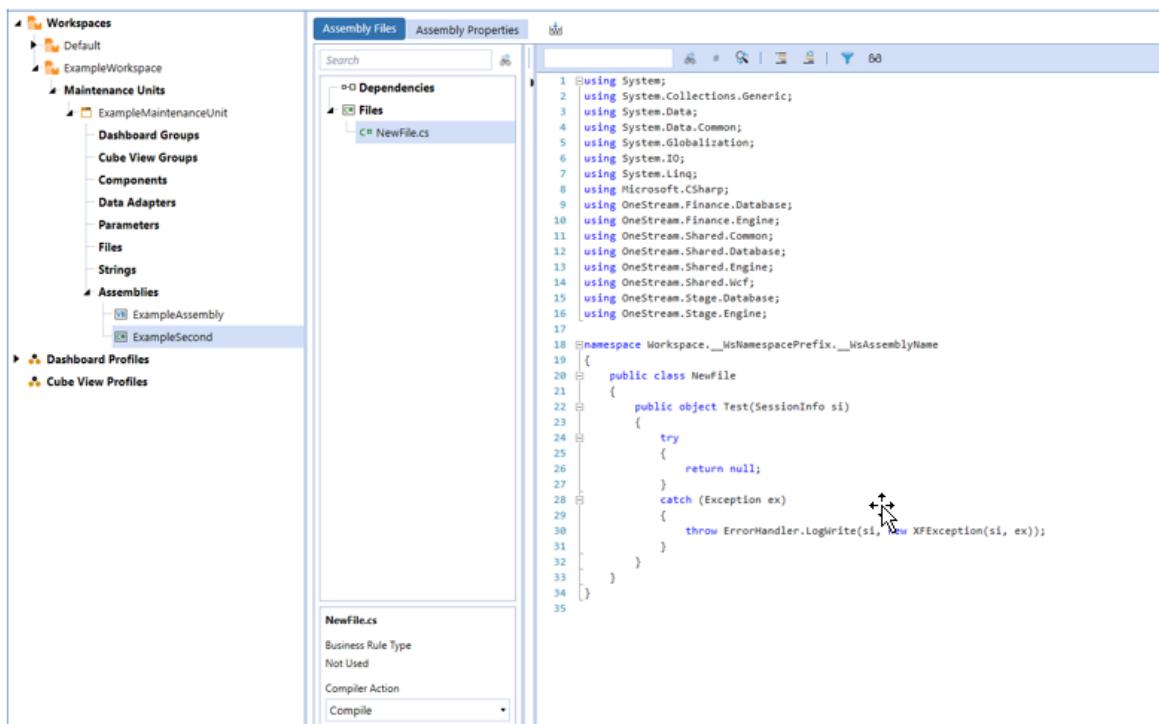
1. Choose an assembly and click the **Assembly Files** tab.
2. Right-click on **Files** and select **Add File**.



## Presenting Data With Books, Cube Views and Dashboards

---

3. In the **Add File** dialog box:
  - a. Type a file name.
  - b. Select a **Source Code** type. You can choose either **Business Rule** or **Service Type**.
  - c. Select a **Compiler Action**. You can choose to compile the file when the assembly is compiled. Use Default is the same as Compile.
4. Click **OK** then **Save**. Your workspace will look similar to the following.



**NOTE:** You can change Compile Action inside this window but you cannot change the rule type. To do so, you need to delete the file.

### Delete an Assembly File

1. Click on the file you would like to delete.
2. On the Assembly Properties tab, click **Delete**.

### Encrypt an Assembly File

You may want others to have access to your Assembly Files. Encryption protects specific assembly files from users who may or may not have access to this workspace. Security can be controlled in the following ways:

- Workspace level (access and maintenance security)
- Maintenance unit level (access and maintenance security)
- Application security roles that prevent others from writing assemblies in the application workspaces page (AdministerApplicationWorkspaceAssemblies)
- System security roles that prevent others from writing assemblies in the system workspaces page (SystemApplicationWorkspaceAssemblies)

You must be inside the EncryptBusinessRules application security property to perform the following steps.

1. Right-click on an assembly file.
2. Select **Encrypt File**.
3. Provide a password and click **OK**. After you have provided a password, the file will be encrypted.
4. Optional. If you would like to decrypt the file, right-click and select **Decrypt File**.

### Create an Assembly Folder

Assembly folders organize your assembly files. To create an assembly folder:

1. Select the assembly then click the **Assembly Files** tab.
2. Right-click the **Files** label and select **Add Folder**.
3. Give the folder a name and click **OK**.
4. Optional. To add files to the folder, right-click on the folder and select **Add File**.

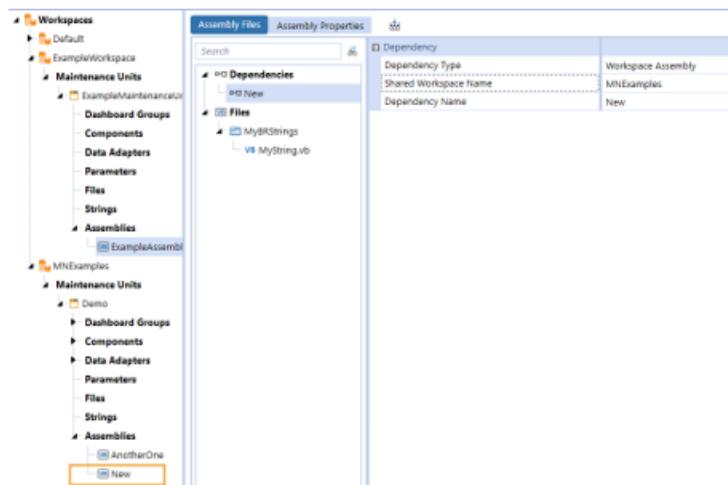
**NOTE:** You cannot drag and drop files into a folder.

### Create a Dependency

Dependencies are useful when you have methods within other assemblies or business rules.

1. Right-click the **Dependencies** label and select **Add Dependency**.
2. Choose a dependency type:
  - **Workspace Assembly**: Used when other assemblies might be included in this workspace. If you are creating a dependency with another workspace, you must provide the Shared Workspace Name and this workspace must have the Is Shareable Workspace property set to True.

Here is an example of how you would complete this dependency setup (multiple workspaces can be entered here in a comma delimited list).



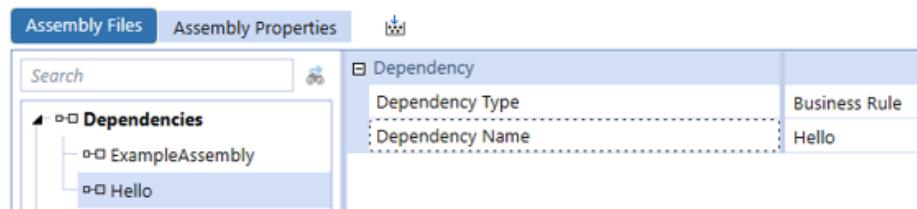
**NOTE:** If you are creating a dependency within the same workspace, do not fill in the Shared Workspace Name property.

- **Prepackaged Assembly**: Assembly automatically included with the OneStream installation.
- **Business Rule**: Create dependencies with any business rules located on the Business Rules page. Type the name of the business rule in the Dependency Name

## Presenting Data With Books, Cube Views and Dashboards

---

property and click **Save**.



After creating dependencies, you might receive an error when compiling the assembly if there is anything incorrect within the assembly or dependency itself.

Important notes when compiling with assemblies:

- If the Compiler Action is set to Disabled for any dependent assembly file, you will not see an error even if you have issues with your syntax. For example, the following dependent assembly would not produce a syntax error even though one is present.

The screenshot shows the 'Assembly Properties' dialog on the left and a code editor on the right. In the code editor, line 24 contains the invalid VB code 'feajsklifaeshjlkasfe'. A red arrow points from the 'Compiler Action' field in the properties dialog to this line of code. The 'Compiler Action' field is set to 'Disabled'.

```
Imports System.Linq
Imports Microsoft.VisualBasic
Imports OneStream.Finance.Database
Imports OneStream.Finance.Engine
Imports OneStream.Shared.Common
Imports OneStream.Shared.Database
Imports OneStream.Shared.Engine
Imports OneStream.Shared.Wcf
Imports OneStream.Stage.Database
Imports OneStream.Stage.Engine
Namespace Workspace._WsNamespacePrefix._WsAssemblyName.BusinessRule.DashboardStringFunction
    Public Class MainClass
        Public Function Main(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api
            Try
                If args.FunctionName.XFEqualsIgnoreCase("SayHello") Then
                    Return "Hello"
                    feajsklifaeshjlkasfe
                End If
            Return Nothing
            Catch ex As Exception
                Throw ErrorHandler.LogError(si, New XFEException(si, ex))
            End Try
        End Function
    End Class
End Namespace
```

- Assembly Files properties require you to type in the exact name of the workspace, assembly, and business rule. If incorrect, you will receive a compile error.

## Source Code Types

When you [create an assembly file](#), you must choose a source code type. This populates the Business Rule Type property inside the rule. You can choose from two source code types: Business Rule types and Service types.

There are several duplications between Service types and Business Rule types, which provides a choice on how you write rules in assemblies. For example, the XFBR String Service and the Dashboard String Function business rule both function the same. In some cases, you might realize slight performance gains when using Service types as they offer more flexibility.

### Business Rule Types

There are two ways to create the various business rule types (cube view extenders, XFBR strings, dashboard extenders, spreadsheets, or data sets):

1. Choosing a business rule type when the assembly is created.
2. [Configuring a Service Factory](#).

In this section, you will learn more about the first option as this is more familiar to developers who have written business rules in OneStream. However, the recommended action for solution developers is to use the second option due to performance impacts.

Regardless of the option you choose, it is important to be aware of the slight differences when calling rules created as assemblies versus calling business rules.

## Dashboard String Function Business Rules in Assembly Files

The following examples of an XFBRString were created as files within assemblies. These types of rules are often referenced throughout the application to more finely tune formatting and display options. These business rules are usually called using the following syntax:

### **XFBR(RuleName, FunctionName)**

However, if this XFBR was created as an assembly, it would look like this:

### **XFBR(Workspace.WorkspaceName.AssemblyName.FileName, FunctionName)**

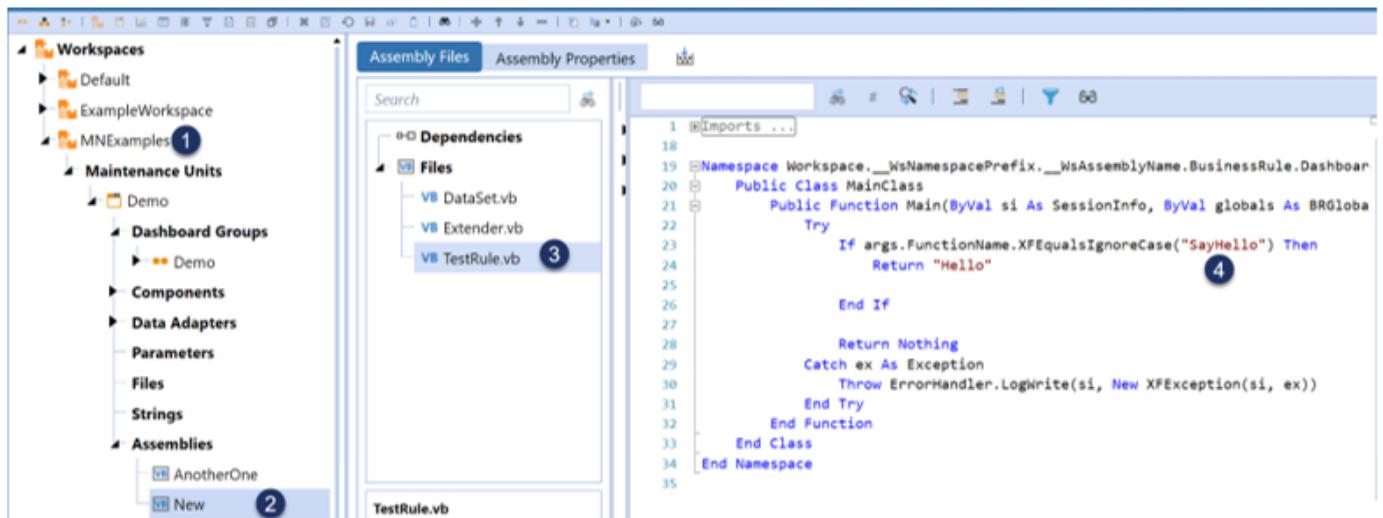
More information is required to call this rule to ensure correctness, such as:

### **XFBR(Workspace.MNExamples.New.TestRule, SayHello) or XFBR (Workspace.Current.New.TestRule, SayHello)**

## Presenting Data With Books, Cube Views and Dashboards

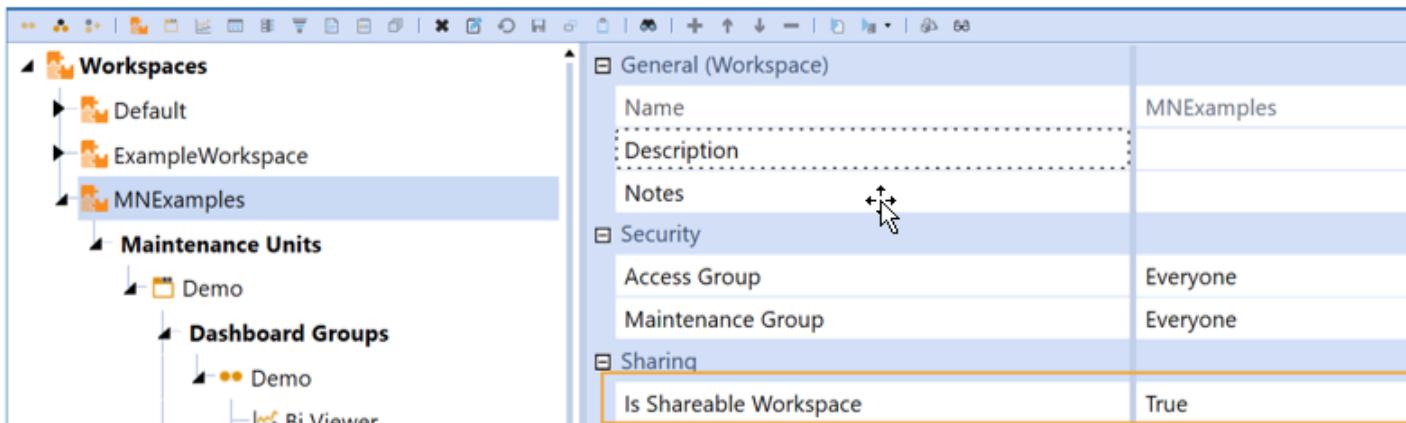
---

For workspace name, you can use the word "Current" to refer to the workspace you are currently in. This is useful when you are referencing rules within objects created in each workspace.



1. Name of the workspace
2. Name of the assembly
3. Name of the assembly file
4. Source code

**NOTE:** If you are not using “Current” as the workspace name you must have the [Is Shareable Workspace](#) property set to True, even if you are referencing the rule within the same workspace. If you do not do this, your rule will not run.



## Dashboard Extender Business Rules in Assembly Files

In the following example, when a button is clicked, a dashboard extender rule completes the selected workflow profile.

**Component Properties**

General (Component)	
Formatting	
Image	
Button	
Action	
Bound Parameter	
Parameter Value For Button Click	
Apply Selected Value To Current Dashboard	True
Save Action	
Selection Changed Save Action	No Action
Selection Changed Save Arguments	
POV Action	
Selection Changed POV Action	No Action
Selection Changed POV Arguments	
Server Task	
Selection Changed Server Task	Execute Dashboard Extender Business Rule (General Server)
Selection Changed Server Task Argument:	(Workspace.Current.New.Extender){CompleteWorkflow}{Profile=[ WFProfile ], Scenario=[ WFSchedule ], Time=[ WFTime ]}

Typical syntax for this rule written in the business rules page would be:

**{MyDashboardExtenderBRName}{MyFunction}{Param1=[MyValue1], Param2=[MyValue2]}**

Because this is an assembly, however, the rule is modified to provide more specific syntax:

{Workspace.WorkspaceName.AssemblyName.FileName}{MyFunction}{Param1=[MyValue1], Param2=[MyValue2]}

## Dashboard Data Set Business Rules in Assembly Files

In this example, a Method Query Data Adapter is referencing a business rule. However, because this was created as an assembly file, appropriate syntax must be used. This type of business rule is typically called with the following syntax:

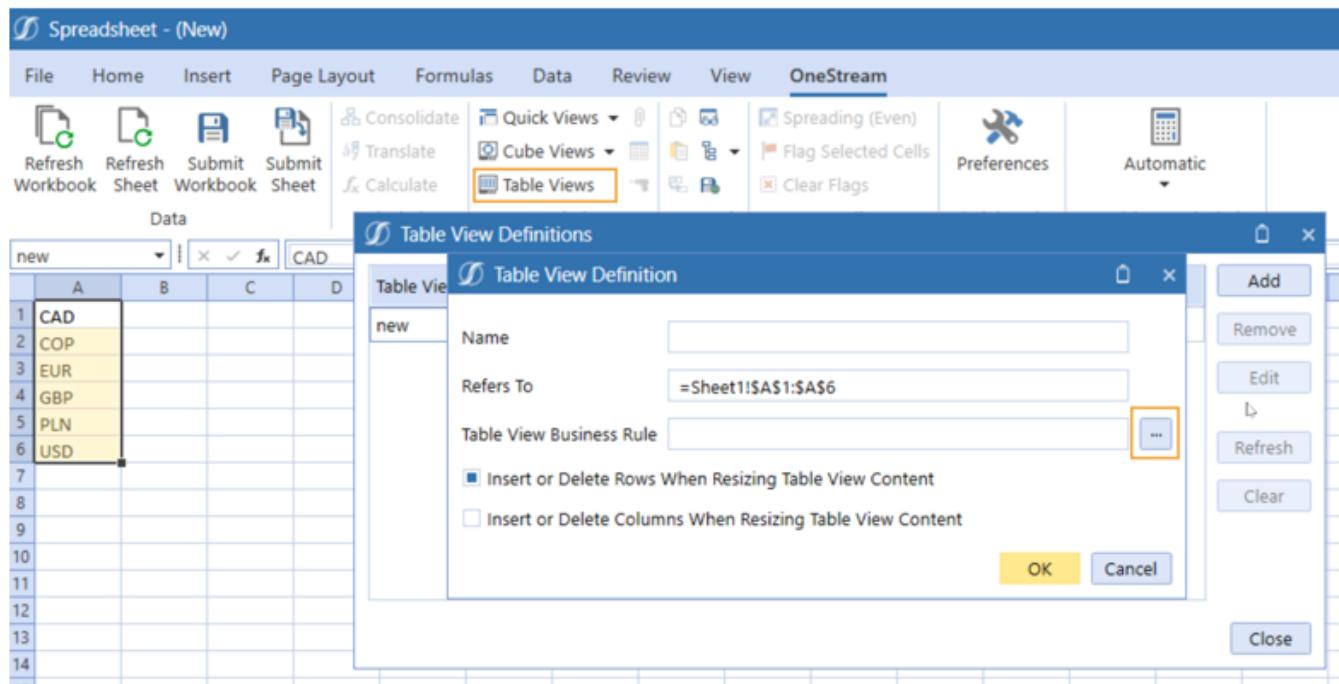
{MyDataSetBRName}{DataSetName}{Param1=[MyValue1], Param2=[MyValue2]}

Because this is an assembly, it has to be modified:

{Workspace.WorkspaceName.AssemblyName.FileName}{DataSetName}{Param1=[MyValue1], Param2=[MyValue2]}

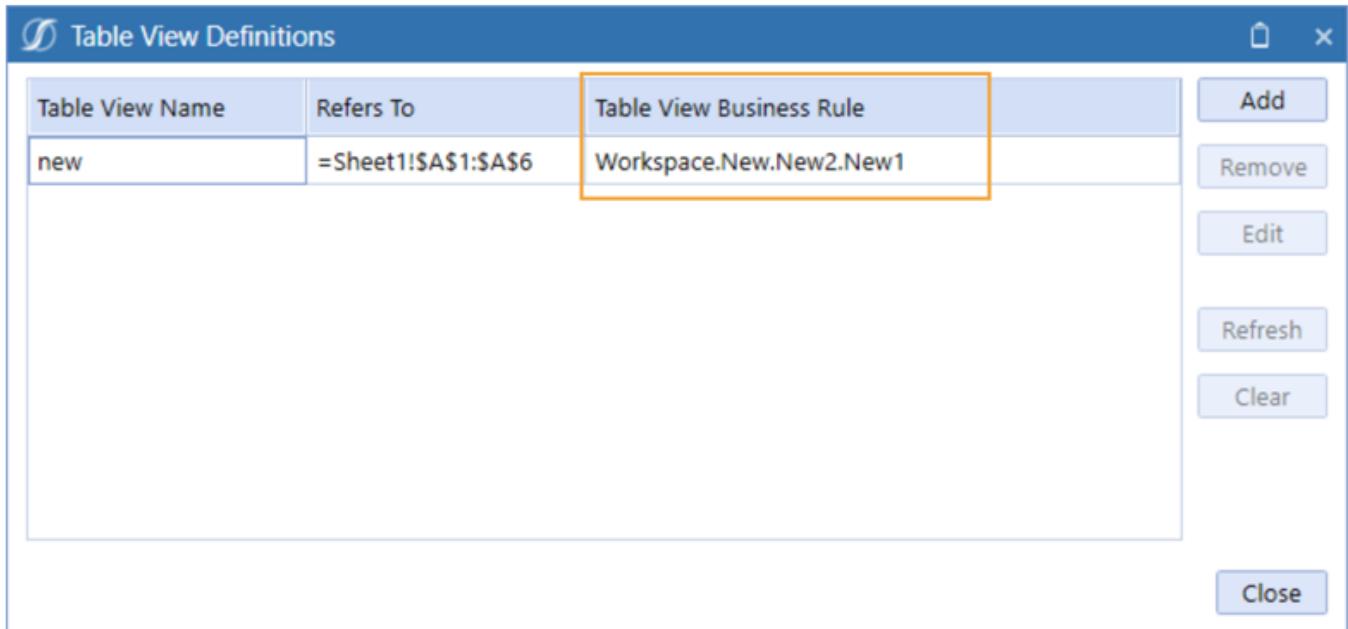
## Spreadsheet Business Rules in Assembly Files

Business rule names are typically chosen by clicking the ellipsis icon in the spreadsheet:



However, if you have created the spreadsheet rule as an assembly file, you will notice that the rule cannot be found by clicking the ellipsis icon. The spreadsheet assembly can still be referenced using the business rules syntax:

**Workspace.WorkspaceName.AssemblyName.FileName**

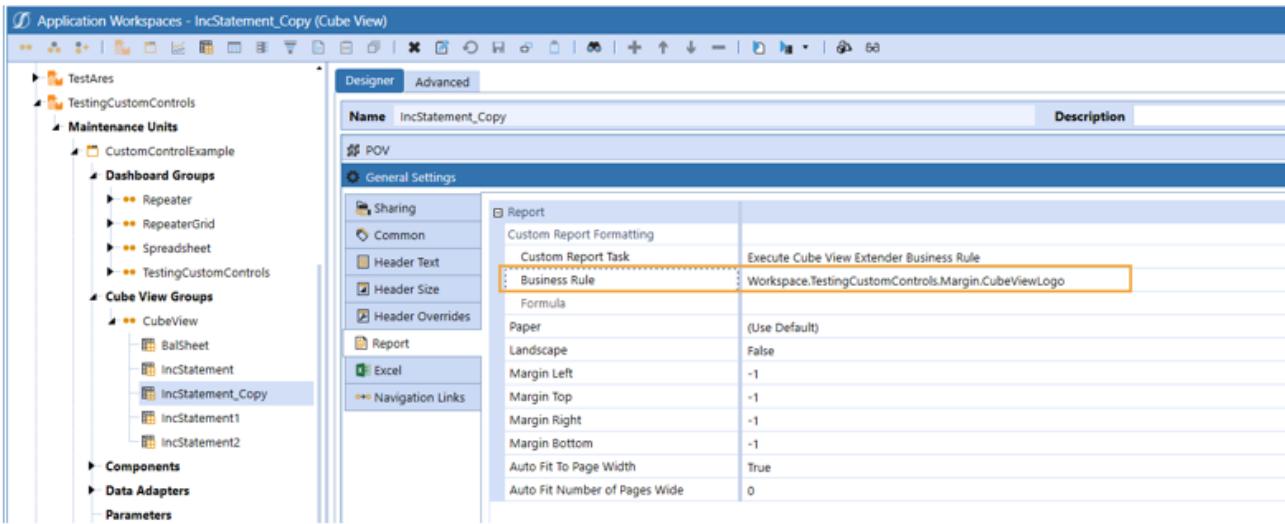


## Cube View Extender Business Rules in Assembly Files

Cube view extender rules are commonly used to fine-tune formatting of cube views. You can write these rules directly on a cube view, from the Business Rules page, or through assemblies. These rules run only on the PDF version of the cube view.

## Presenting Data With Books, Cube Views and Dashboards

---



To reference a cube view extender rule that was created as an assembly, you should set the cube view custom report task to "Execute Cube View Extender Business Rule." Refer to the rule using the following syntax:

**Workspace.WorkspaceName.AssemblyName.FileName**

Do not use the ellipsis icon to select a business rule in this case, as this only provides rules present in the Business Rules page.

## XF Project

XF Project extract is for application project designers who are building solutions that span many artifacts, such as workspaces, dashboard maintenance units, business rules, cubes, dimensions, cube views, and other artifacts. The application Extract and Load option collects defined objects, such as dashboards and business rules, as a single file export package that can be reloaded as a package.

XF Project is a convenient way to organize workspaces, data maintenance units, or similar solutions into a folder structure that can be integrated with a version control system, such as Git. Developers must create an XML file that is the definition for the contents of the project export.

**NOTE:** When working with cube views that are in the non-default workspace, the workspace must be shareable.

### Project File

Application designers must first manually define an XML file to support the export of objects as a project file. The file is then saved with the .xfProj file extension to a local project folder that also supports a version control system.

```
1 <xfProject topFolderPath="" defaultZipFileName="">
2   <projectItems>
3     <projectItem itemType="DashboardWorkspace" folderPath="" workspace="Corporate" name="Corporate" includeDescendants="false" />
4     <projectItem itemType="DashboardMaintenanceUnit" folderPath="" workspace="Corporate" name="Corporate Reports" includeDescendants="true" />
5     <projectItem itemType="DashboardMaintenanceUnit" folderPath="" workspace="Corporate" name="Corporate Templates" includeDescendants="true" />
6     <projectItem itemType="DashboardProfile" folderPath="" workspace="Corporate" name="Corporate Solution" includeDescendants="true" />
7   </projectItems>
8 </xfProject>
```

The root node of xfProject contains two attributes:

- **TopFolderPath:** Creates and defines the starting folder location where the specified files are extracted.
- **DefaultZipFileName:** Creates a standard default file name for .zip file extracts.

**ProjectItems** is a list structure that contains the project items to extract. No attributes needed.

**ProjectItem** reflects what is needed to extract from OneStream or to load from the file system. It has five attributes:

- **ProjectItemType:** The type of the project item. Includes the following types:
  - BusinessRule
  - Cube
  - CubeViewGroup
  - CubeView
  - CubeViewProfile
  - DashboardWorkspace
  - DashboardMaintenanceUnit
  - DashboardFile
  - DashboardString

- DashboardParameter
  - DashboardGroup
  - DashboardAdapter
  - DashboardComponent
  - Dashboard
  - WorkspaceAssembly
  - DashboardProfile
  - DataManagementGroup
  - DataManagementStep
  - DataManagementSequence
  - DataManagementProfile
  - DataSource
  - Dimension
  - TransformationRuleGroup
  - TransformationRuleProfile
- **FolderPath:** The name of the subfolder where the project item type is extracted.
  - **Workspace:** The name of the workspace for the project item.
  - **Name:** The name of the project item.
  - **IncludeDescendants:** The default is True and only affects the following project item types:
    - CubeViewGroup
    - DashboardWorkspace
    - DashboardGroup

- DashboardMaintenanceUnit
- DataManagementGroup

### File Extract

You can place the .xfProj file into a local folder, such as your desktop. The defined folder path folders will be generated there as the target location for application exports and loads. There are two file extract options available:

- **.zip**: The export option will collect all objects defined in the .xfProj file as a zip file to the location of the .xfproj file.
- **File**: Exports all objects defined in the .xfProj file to the folder path locations defined in the .xfproj file.

To use file extract:

1. Navigate to **Application > Tools > Load/Extract**.
2. Click the **Extract** tab.
3. Under File Type, select XF Project.
4. Click the ellipses (...) and navigate to the .xfProj file.
5. (Optional) Select the **Extract to Zip** checkbox to create an application zip file that contains all defined objects.
6. Click **Extract** on the toolbar.

**NOTE:** For XML extract, if you select the file type of “Application Workspaces,” the system will update the third line of code from <applicationDashboardsRoot> to <applicationWorkspacesRoot> on the extract.

For XML load, the system properly loads <applicationWorkspacesRoot>. To support backward compatibility, if you have an XML extract that contains <applicationDashboardsRoot>, the system still loads the older version.

### File Load .xfProj

File loading using the defined .xfProj file provides a seamless link to the project files. When you load an .xfProj file you will see the option to merge or replace the target files. The only files affected are those defined by the .xfproj file.

If you select replace, only files that differ for CubeViewGroups, DashboardWorkspaces, DataManagementGroups, DashboardMaintenanceUnits, and DashboardGroups are removed. For all other items, such as business rules or extensibility rules, replace acts as a merge. You can remove files that differ with Assembly file folders as well.

To load an .xfProj file:

1. Navigate to **Application > Tools > Load/Extract**.
2. Click the **Load** tab.
3. Under File Name, browse to the .xfProj file and select it.
4. Click **Load** on the toolbar.
5. Select the load method, either Merge or Replace.

**NOTE:** For XF Project, when extracting a file, the folder structure is updated from “Application Dashboards” to “Application Workspaces.”

### Zip Load

Zip file load functions like any other application file load. The contents of the file are merged into the application. Zip file load is not supported by alternative merge or replace file load options.

### Build Cube Views in Workspaces

Cube views are used for reporting, analysis, and data entry. You can create cube views on the Cube Views page under the Application tab and also build and use them within the Workspaces page.

Depending on your security roles, cube views created on the Cube View Page can be used through the Default Maintenance Unit in the Default Workspace. You can create additional cube views in other workspaces but they cannot be accessed by the Cube View page.

## Presenting Data With Books, Cube Views and Dashboards

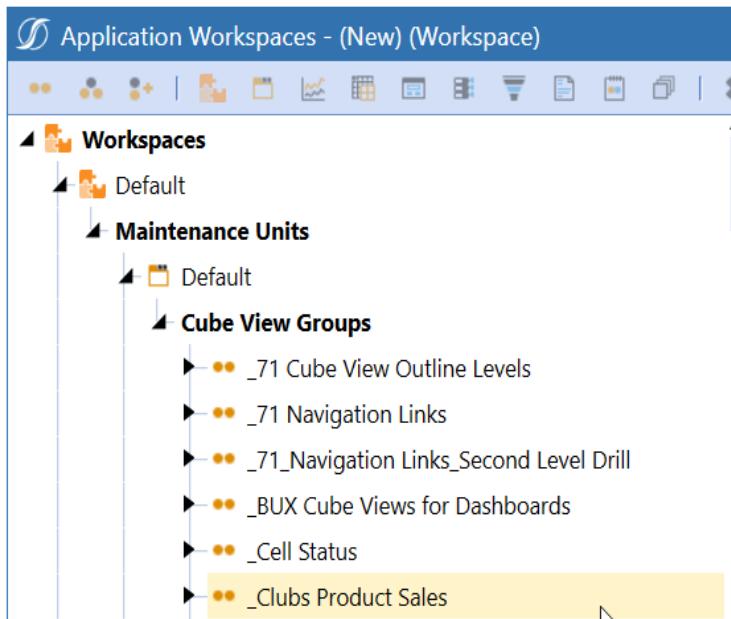
---

You can load and extract cube views, cube view groups, and cube view profiles using the traditional XML process. The file type of Application Dashboards is renamed to Application Workspaces.

### Display Cube Views Page and Groups

There is a new Default folder under Maintenance Units within the Workspaces Default Maintenance Units section. This is where the cube views page and cube view groups reside.

1. On the **Application** tab, under **Presentation**, click **Workspaces**.
2. In the **Application Workspaces** pane, under **Workspaces**, expand **Default > Maintenance Unit > Default > Cube View Groups**.

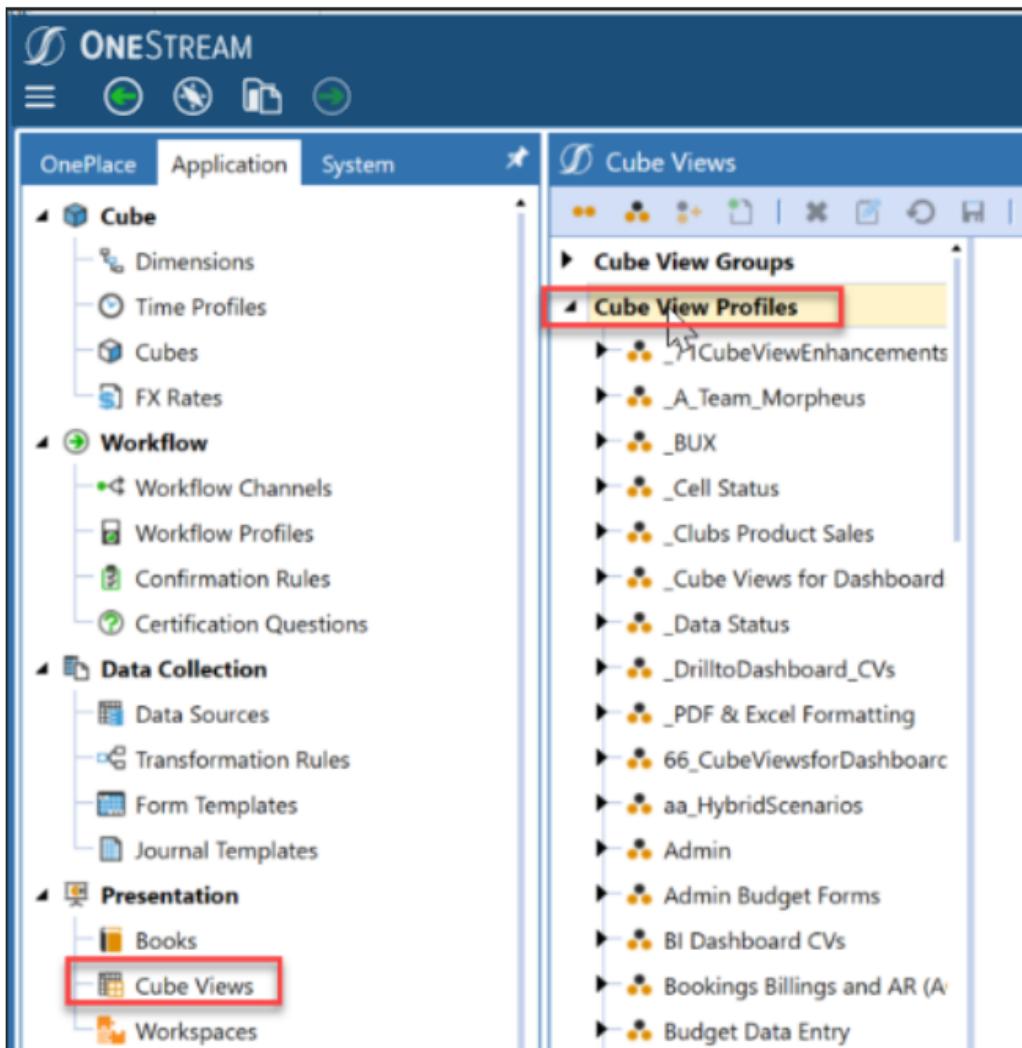


Role	View	Location
Application Administrator	Cube View Profiles	<b>Application &gt; Presentation &gt; Cube Views</b>
	Workspaces	<b>Application &gt; Presentation &gt; Workspaces</b>
	Cube View Groups	<b>Application &gt; Presentation &gt;</b>

Role	View	Location
		<b>Workspaces</b>
	Designer Mode	<b>Application &gt; Presentation &gt; Cube Views &gt; Cube View Groups</b> > select a cube view
	Advanced Mode	<b>Application &gt; Presentation &gt; Cube Views &gt; Cube View Groups</b> > select a cube view > Advanced
Cube View Administrator	Cube Views	<b>Application -&gt; Presentation -&gt; Cube Views</b>
	Cube View Groups	<b>Administration &gt; Presentation &gt; Cube Views</b>
	Designer Mode	<b>Application &gt; Presentation &gt; Cube Views &gt; Cube View Groups</b> > select a cube view
	Advanced Mode	<b>Application &gt; Presentation &gt; Cube Views &gt; Cube View Groups</b> > select a cube view > Advanced

### Display Cube View Profiles and Groups

1. On the **Application** tab, under **Presentation**, click **Cube Views**.
2. In the **Cube Views** pane, expand **Cube View Groups** or **Cube View Profiles**.



## Configure the Service Factory

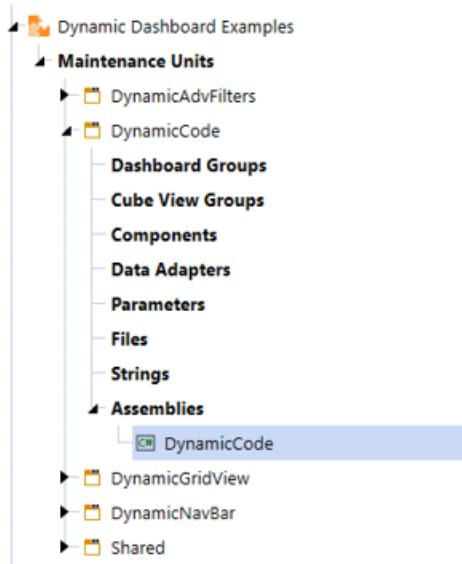
You should write all assemblies inside workspaces. This allows for greater flexibility and provides the ability to create dynamic dashboards. Though several ways exist to create a dynamic dashboard, the following method is recommended. You can complete the setup in three steps, which are defined below.

## Presenting Data With Books, Cube Views and Dashboards

---

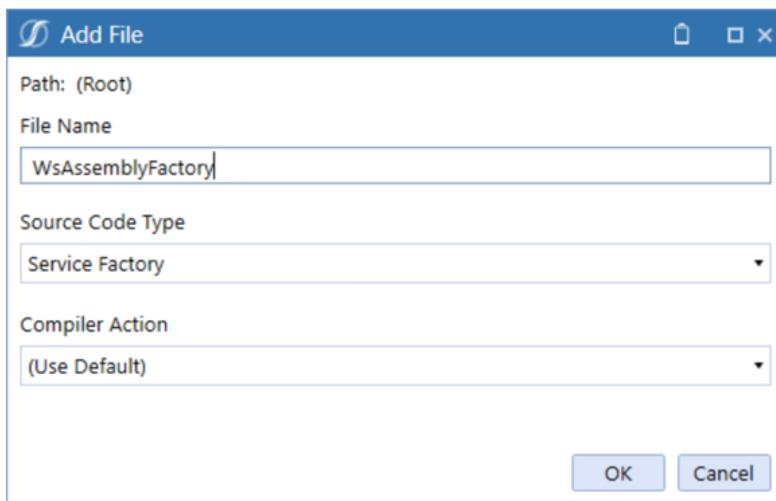
### Step 1: Create a maintenance unit for all assemblies within a workspace.

The first step is organizational, designed to keep all rules within a single location. Create a maintenance unit within the workspace to hold all necessary assemblies. The following image displays multiple maintenance units in each workspace, but only DynamicCode contains assemblies. This will be the central location for all rules.



### Step 2: Create an assembly file for the Service Factory.

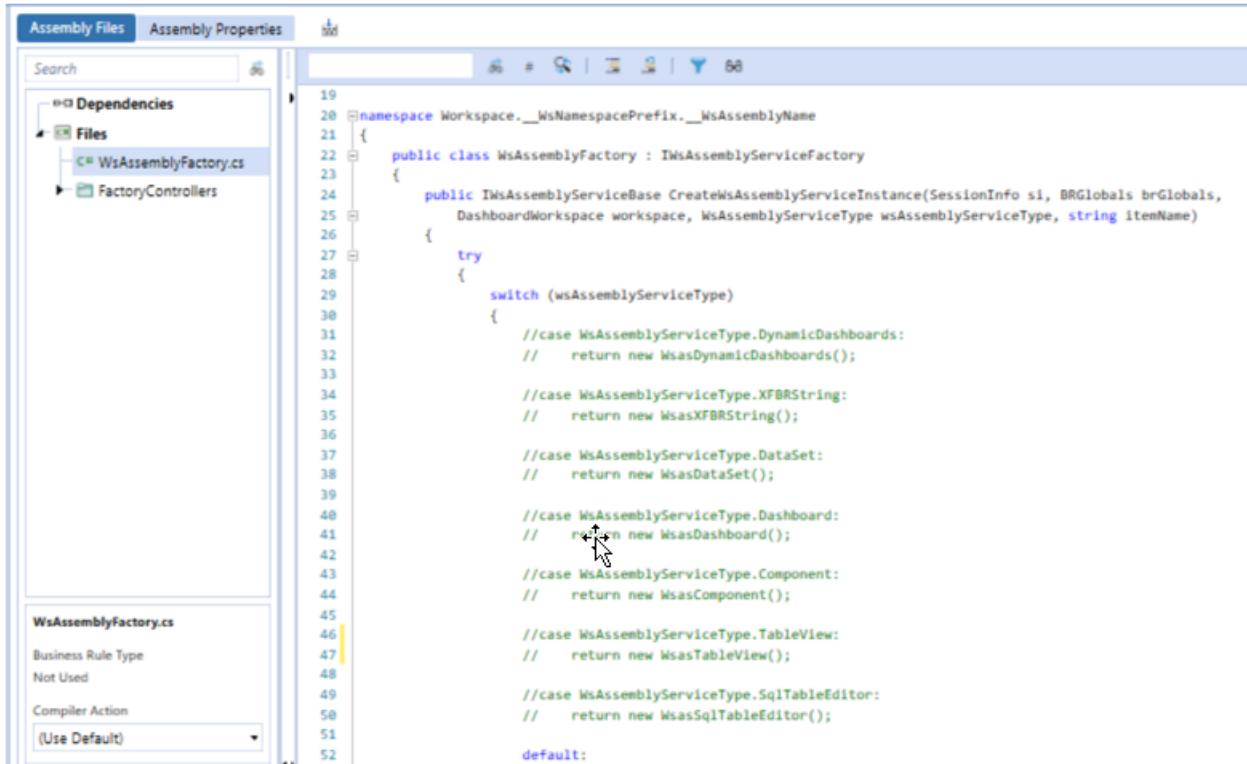
The second step is to [create an assembly file](#) with the source code type of Service Factory.



## Presenting Data With Books, Cube Views and Dashboards

---

The Service Factory assembly is needed to take advantage of all items used within assemblies. It should look similar to the following image (the example uses C#):



The screenshot shows a code editor interface for a C# file named `WsAssemblyFactory.cs`. The file contains a class definition for `WsAssemblyFactory` which implements `IWsAssemblyServiceFactory`. The code uses a switch statement to handle different service types, returning various objects like `WsasDynamicDashboards`, `WsasXFBRString`, `WsasDataSet`, `WsasDashboard`, `WsasComponent`, `WsasTableView`, and `WsasSqlTableEditor`. The code editor has tabs for "Assembly Files" and "Assembly Properties". The properties panel on the left shows the file is a "Business Rule Type" and "Not Used". The compiler action is set to "(Use Default)".

```
19
20 namespace Workspace.___WsNamespacePrefix.___WsAssemblyName
21 {
22     public class WsAssemblyFactory : IWsAssemblyServiceFactory
23     {
24         public IWsAssemblyServiceBase CreateWsAssemblyServiceInstance(SessionInfo si, BRGlobals brGlobals,
25             DashboardWorkspace workspace, WsAssemblyServiceType wsAssemblyServiceType, string itemName)
26         {
27             try
28             {
29                 switch (wsAssemblyServiceType)
30                 {
31                     //case WsAssemblyServiceType.DynamicDashboards:
32                     //    return new WsasDynamicDashboards();
33
34                     //case WsAssemblyServiceType.XFBRString:
35                     //    return new WsasXFBRString();
36
37                     //case WsAssemblyServiceType.DataSet:
38                     //    return new WsasDataSet();
39
40                     //case WsAssemblyServiceType.Dashboard:
41                     //    return new WsasDashboard();
42                     //case WsAssemblyServiceType.Component:
43                     //    return new WsasComponent();
44
45                     //case WsAssemblyServiceType.TableView:
46                     //    return new WsasTableView();
47
48                     //case WsAssemblyServiceType.SqlTableEditor:
49                     //    return new WsasSqlTableEditor();
50
51                 default:
52             }
53         }
54     }
55 }
```

The code allows you to create the following objects:

- Dynamic Dashboard
- XFBRString
- DataSet
- Dashboards
- Components
- TableViews
- SqlTableEditor

## Presenting Data With Books, Cube Views and Dashboards

---

If you are not using any of the objects listed, keep them commented out in the code. Otherwise, you will get a syntax error when compiling. For example, if you only need the DynamicDashboard call, the Service Factory would look similar to the following image:

```
20  namespace Workspace._WsNamespacePrefix._WsAssemblyName
21  {
22      public class WsAssemblyFactory : IWsAssemblyServiceFactory
23      {
24          public IWsAssemblyServiceBase CreateWsAssemblyServiceInstance(SessionInfo si, BRGlobals brGlobals,
25              DashboardWorkspace workspace, WsAssemblyServiceType wsAssemblyServiceType, string itemName)
26          {
27              try
28              {
29                  switch (wsAssemblyServiceType)
30                  {
31                      case WsAssemblyServiceType.DynamicDashboards:
32                          return new WsDynamicDashboardsService();           ↴
33
34                      //case WsAssemblyServiceType.XFBRString:
35                      //    return new WsasXFBRString();
36
37                      //case WsAssemblyServiceType.DataSet:
38                      //    return new WsasDataSet();
39
40      }
```

Keep the Unknown (line 26) uncommented.

**Step 3: Configure the workspace assembly service property.**

This last step allows the entire workspace to use the new Service Factory file so your rules run properly. Assembly files will not run correctly if you skip this step.

Go to the workspace and update the Workspace Assembly Service to {WsAssemblyName}. {ServiceFactoryName}. For this example, that would be DynamicCode.WsAssemblyFactory.

## Presenting Data With Books, Cube Views and Dashboards

---

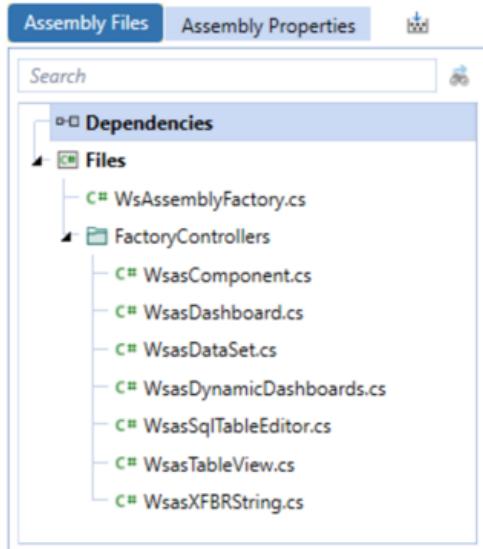
The screenshot shows the Service Factory interface for managing workspaces and maintenance units. On the left, a tree view lists several workspaces under 'Workspaces'. One workspace, 'Dynamic Dashboard Examples', is selected and expanded to show its internal components: Maintenance Units, Dashboard Groups, Cube View Groups, Components, Data Adapters, Parameters, Files, Strings, Assemblies, and Shared files. On the right, a detailed configuration pane for the selected workspace is displayed. The 'General (Workspace)' section includes fields for Name (Dynamic Dashboard Examples), Description, Notes, Access Group (Everyone), Maintenance Group (Everyone), and Sharing (Is Shareable Workspace set to False). The 'Assemblies' section lists multiple assembly imports, with the 'Workspace Assembly Service' entry highlighted by a yellow box. The 'Text' section contains the assembly service name: DynamicCode.WsAssemblyFactory.

You can do this on any given maintenance unit. You should configure the workspace to keep all assembly files within a single maintenance unit. You can designate the workspace and workspace maintenance unit level in all actions.

This screenshot shows the 'Application Workspaces - Advanced Filters Transaction Matching (Maintenance Unit)' interface. The left sidebar lists various workspaces and maintenance units. A specific maintenance unit, 'Advanced Filters Transaction Matching', is selected and expanded to show its components: Advanced Filters Transaction Matching, Code, Dynamic Column Definitions, Navigation Menu Bar, and Shared. On the right, a configuration pane for this maintenance unit is shown. The 'General (Maintenance Unit)' section includes fields for Name (Advanced Filters Transaction Matching), Workspace (\_Dynamic Dashboard Examples 2), Description, Is Mobile (False), and Assemblies (Workspace Assembly Service set to DynamicCode.WsAssemblyFactory). The 'Security' section shows Access Group (Everyone) and Maintenance Group (Everyone).

## Organize Service Factory Assembly Files

With the Service Factory set up, you can build any necessary assembly files to bring the workspace to life. The image below shows a sample organization structure of assembly files.



Notice that the Service Factory assembly file is separate, and a folder holds all necessary factory controller files. The workspaces use one of each calls from the Service Factory assembly file.

While you can refer to business rules created as assemblies in other workspaces, you cannot refer to service types from other workspaces. This means a lighter syntax is required when calling business rules throughout the workspace. The following sections highlight the syntax used to reference the service types.

### XFBRString Service Type

This service type lets you create an XFBR String within assemblies. You can do this by choosing this service type or by choosing the Dashboard String Function business rule. These rules are generally used to create effects similar to parameters and substitution variables but allow greater flexibility by using code to return string values in dashboards, cube views, and extensible documents.

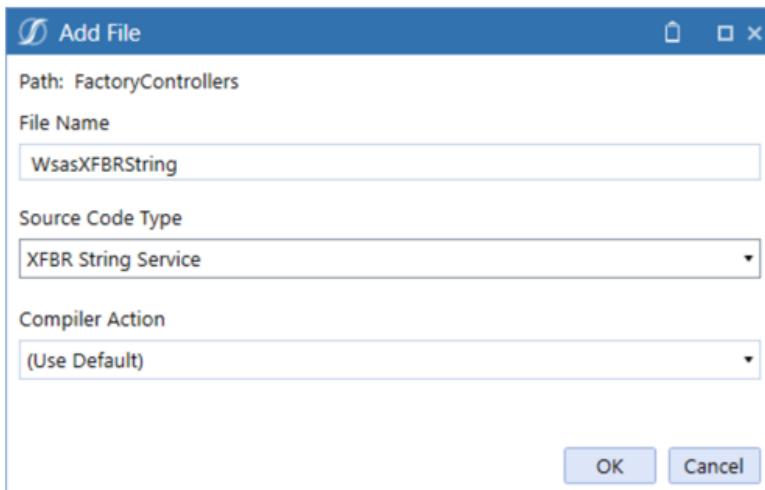
You must create the XFBR service type file after creating the service factory file and after uncommenting the appropriate lines of code. The following is an example of what this looks like in C#:

## Presenting Data With Books, Cube Views and Dashboards

---

```
29         switch (wsAssemblyServiceType)
30         {
31             //case WsAssemblyServiceType.DynamicDashboards:
32             //    return new WsasDynamicDashboards();
33
34             case WsAssemblyServiceType.XFBRString:
35                 return new WsasXFBRString();
36
37             //case WsAssemblyServiceType.DataSet:
38             //    return new WsasDataSet();
39
40             //case WsAssemblyServiceType.Dashboard:
41             //    return new WsasDashboard();
42
43             //case WsAssemblyServiceType.Component:
44             //    return new WsasComponent();
45
46             //case WsAssemblyServiceType.TableView:
47             //    return new WsasTableView();
48
49             //case WsAssemblyServiceType.SqlTableEditor:
50             //    return new WsasSqlTableEditor();
51
52         default:
53             return null;
54     }
55 }
```

Given the changes in the code, you would configure the assembly file in the following way:

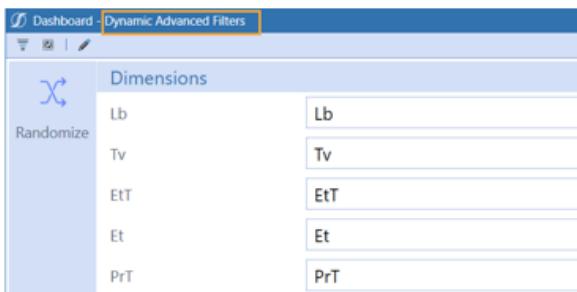


**NOTE:** The file name should match the return function in the Service Factory file.

The image below shows an example of an assembly file returning a page caption in a dashboard. This is commonly used to query an object or to create a dynamic page caption.

## Presenting Data With Books, Cube Views and Dashboards

---



This assembly is then referenced in the Page Caption of the dashboard.

A screenshot showing the Assembly Explorer on the left and the Dashboard Properties dialog on the right. The Assembly Explorer tree view shows a node 'DynamicAdvFilters' expanded, revealing 'Dashboard Groups' which further expand into 'Distinct Values Dialog' and 'Main' groups, each containing several sub-components like '0000\_DistinctDialog' and '0000\_AdvFilters\_Main'. The 'Main' group's '0000\_AdvFilters\_Main' component is selected. The 'Dashboard Properties' dialog is open, showing the 'General (Dashboard)' tab with the following details:

Name	0000_AdvFilters_Main
Workspace	_Dynamic Dashboard Examples 2 DO NOT EDIT
Maintenance Unit	DynamicAdvFilters
Dashboard Group	Main
Description	(empty)
Page Caption	XFBR(WS, GetDashboardPageCaption)
Notes	(empty)

The 'Behavior' tab is also visible, showing 'Dashboard Type' set to 'Top Level'.

The syntax for referencing this object is XFBR(WS,FunctionName).

The syntax for using the Assembly Service on the Maintenance level is XFBR(WSMU,GetRandomText).

### Dashboard Service Type

This service type lets you create a LoadDashboard function type in a dashboard extender rule within assemblies. You can do this by choosing this service type or by choosing the Dashboard Extender business rule type. These rules are usually used to perform custom tasks within workspaces. Developers often use the Dashboard service type with selection components, such as combo boxes to set parameters to a default value or the last selected value.

Create this file after the Service Factory file has been created and the appropriate lines of code have been uncommented. Here is an example of a dashboard service type in C#:

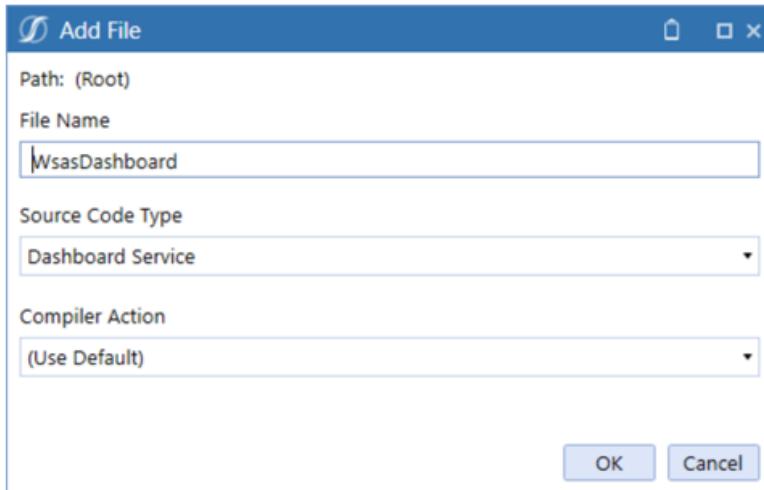
## Presenting Data With Books, Cube Views and Dashboards

---

```
29         switch (wsAssemblyServiceType)
30     {
31         //case WsAssemblyServiceType.DynamicDashboards:
32         //    return new WsasDynamicDashboards();
33
34         //case WsAssemblyServiceType.XFBRString:
35         //    return new WsasXFBRString();
36
37         //case WsAssemblyServiceType.DataSet:
38         //    return new WsasDataSet();
39
40         case WsAssemblyServiceType.Dashboard:
41             return new WsasDashboard();
42
43         //case WsAssemblyServiceType.Component:
44         //    return new WsasComponent();
45
46         //case WsAssemblyServiceType.TableView:
47         //    return new WsasTableView();
48
49         //case WsAssemblyServiceType.SqlTableEditor:
50         //    return new WsasSqlTableEditor();
51
52         default:
53             return null;

```

Given the changes in the code, you would configure the dashboard service file in the following way:



**NOTE:** The file name should match the return function in the Service Factory file.

The following is an example of an assembly file that is populating default values inside labels:

## Presenting Data With Books, Cube Views and Dashboards

---

Attributes	
A1	A1
A2	A2
A3	A3
Ib	Ib
Tv	Tv
EtT	EtT
Et	Et
PrT	PrT
Pr	Pr

The labels should be populated at runtime, so the assembly file is referenced in the following manner:

The screenshot shows the BusinessObjects Administration interface. On the left, there is a tree view under 'Workspaces' containing several workspace entries. On the right, there are two tabs: 'Dashboard Properties' (selected) and 'Dashboard Components'. Under 'Dashboard Properties', there are sections for General (Dashboard), Behavior, Processing, Formatting, and Action (Primary Dashboard Only). Under 'Action', there is a 'Server Task' section with two items: 'Load Dashboard Server Task' and 'Load Dashboard Server Task Arguments'. The 'Arguments' item has a value of '{WS}{LoadDefaultValues}()' highlighted with an orange box. Below this, there is a 'Grid Layout Type' section with settings for Number Of Rows (1), Number Of Columns (2), Row 1 Type (Component), and Row 1 Height (e.g., 150, \*, 2\*, Auto).

The syntax for referencing this object is {WS}{FunctionName}{Parameter1=Value1}.

The syntax for using the Assembly Service on the Maintenance level is {WSMU}{FunctionName}{Parameter1=Value1}.

### Component Service Type

This service type lets you create a ComponentSelectionChanged function type in a dashboard extender rule within assemblies. You can do this by choosing this service type or by choosing the Dashboard Extender business rule type. These rules are usually used to perform custom tasks within workspaces. Developers often use this Dashboard extender rule as it handles all actions occurring due to interactions with dashboard components, such as grids or combo boxes. These rules are referenced in the action properties of a given component.

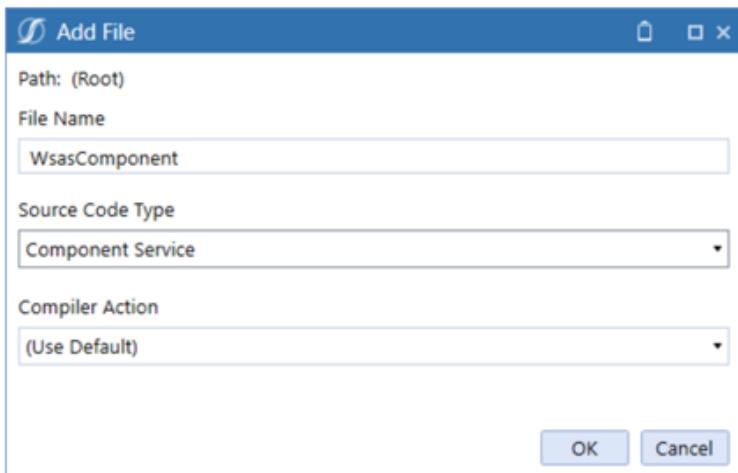
Create this file after the Service Factory file has been created and the appropriate lines of code have been uncommented. Here is an example of a dashboard service type in C#:

```
29         switch (wsAssemblyServiceType)
30     {
31         //case WsAssemblyServiceType.DynamicDashboards:
32         //    return new WsasDynamicDashboards();
33
34         //case WsAssemblyServiceType.XFBRString:
35         //    return new WsasXFBRString();
36
37         //case WsAssemblyServiceType.DataSet:
38         //    return new WsasDataSet();
39
40         //case WsAssemblyServiceType.Dashboard:
41         //    return new WsasDashboard();
42
43         case WsAssemblyServiceType.Component:
44             return new WsasComponent();
45
46         //case WsAssemblyServiceType.TableView:
47         //    return new WsasTableView();
48
49         //case WsAssemblyServiceType.SqlTableEditor:
50         //    return new WsasSqlTableEditor();
51
52         default:
53             return null;
54     }
55 }
```

Given the changes in the code, you would configure the dashboard service file in the following way:

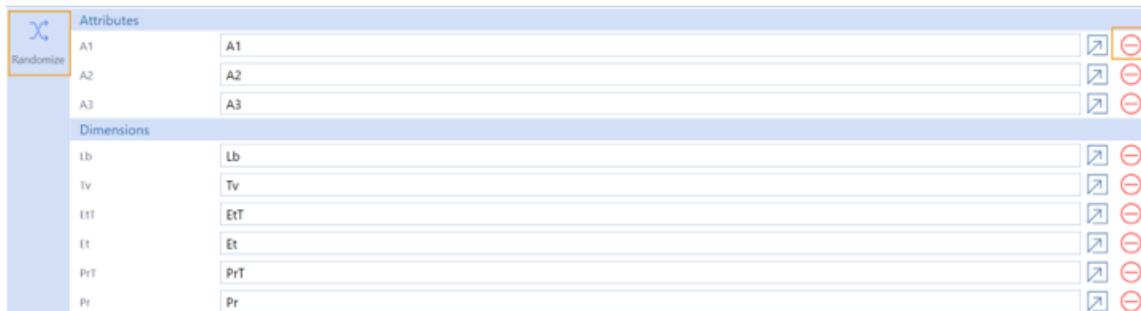
## Presenting Data With Books, Cube Views and Dashboards

---



**NOTE:** The file name should match the return function in the Service Factory file.

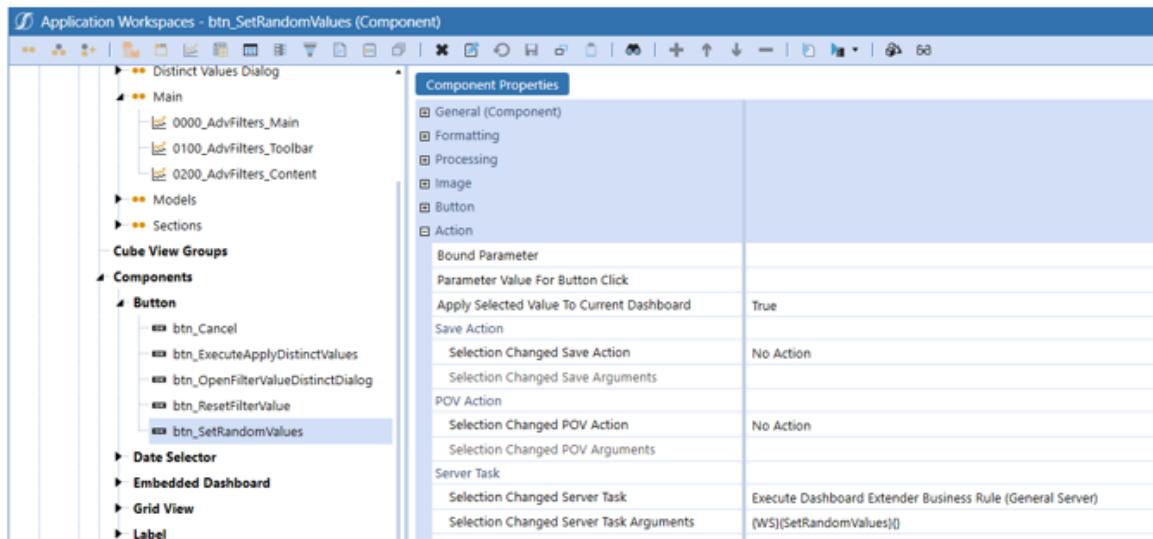
The following is an example of an assembly file that is operating the two highlighted icons in the dashboard:



The Randomize icon fills the labels with random values, while the second icon resets the values to their original state. The logic, populated within the assembly file, runs when you click a button.

## Presenting Data With Books, Cube Views and Dashboards

---



The syntax for referencing this object is {WS}{FunctionName}{Parameter1=Value1}.

The syntax for using the Assembly Service on the Maintenance level is {WSMU}{FunctionName}{Parameter1=Value1}.

### Data Set Service Type

This service type lets you create Data Set logic within assemblies. You can do this by choosing this service type or by choosing the Dashboard Data Set business rule type. These rules are usually queries and return data to data adapters and parameters using a method query.

Create this file after creating the Service Factory file and after the appropriate lines of code have been uncommented. Here is an example of a data set service type in C#:

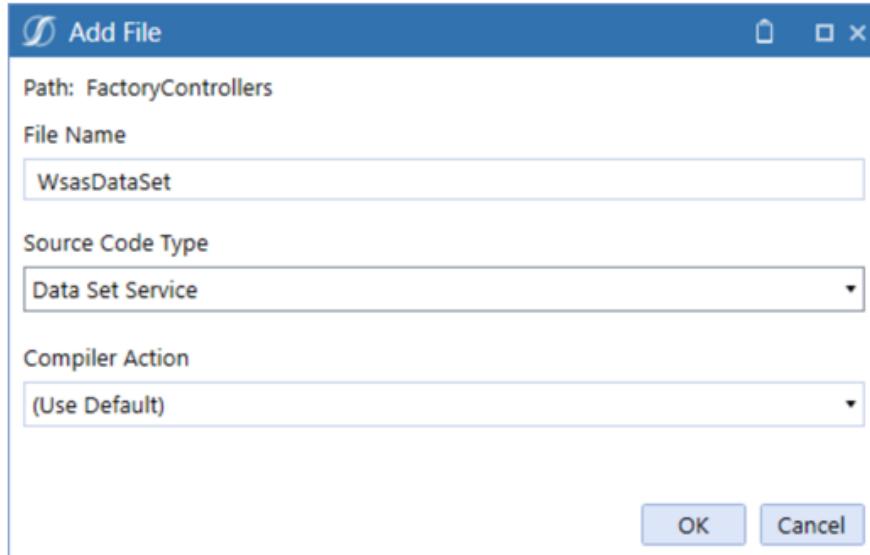
## Presenting Data With Books, Cube Views and Dashboards

---

```
29         switch (wsAssemblyServiceType)
30     {
31         //case WsAssemblyServiceType.DynamicDashboards:
32         //    return new WsasDynamicDashboards();
33
34         //case WsAssemblyServiceType.XFBRString:
35         //    return new WsasXFBRString();
36
37         case WsAssemblyServiceType.DataSet:
38             return new WsasDataSet();
39
40         //case WsAssemblyServiceType.Dashboard:
41         //    return new WsasDashboard();
42
43         //case WsAssemblyServiceType.Component:
44         //    return new WsasComponent();
45
46         //case WsAssemblyServiceType.TableView:
47         //    return new WsasTableView();
48
49         //case WsAssemblyServiceType.SqlTableEditor:
50         //    return new WsasSqlTableEditor();
51
52         default:
53             return null;

```

Given the changes in the code, you would configure the dashboard service file in the following way:



**NOTE:** The file name should match the return function in the Service Factory file.

## Presenting Data With Books, Cube Views and Dashboards

---

The following is an example of an assembly file that creates a grid of potential values you might see based on a selection. The assembly file activates when you click on an icon.

This screenshot shows a software interface for managing data dimensions. On the left, there's a sidebar with buttons for 'Randomize' and 'X'. The main area is divided into 'Attributes' and 'Dimensions' sections. Under 'Attributes', there are two rows: A1 (containing A1) and A2 (containing A2). Under 'Dimensions', there are 15 rows: Lb, Tv, EtT, Et, PrT, Pr, CnT, Cn, SnT, Sn, TmT, Tm, VwT, and Vw. Each dimension row contains a dropdown menu with a list of items. To the right of each dimension row is a column of icons: a blue square with a white arrow pointing up-right, a red circle with a minus sign, and another blue square with a white arrow pointing up-right. The first blue square in the Lb column is highlighted with a yellow box.

This runs through a method query data adapter that populates the grid component.

This screenshot shows the Power BI Model Editor. On the left, there's a navigation pane with sections for 'DynamicAdvFilters', 'Dashboard Groups', 'Cube View Groups', 'Components', and 'Data Adapters'. Under 'Data Adapters', a node for 'grd\_GetDistinctValuesForField' is selected and highlighted with a blue box. On the right, there's a configuration pane titled 'General (Data Adapter)' with the following settings:

Name	grd_GetDistinctValuesForField
Workspace	_Dynamic Dashboard Examples 2 DO NOT EDIT
Maintenance Unit	DynamicAdvFilters
Description	(Optional)
Processing	Template Parameter Values (e.g., Param1=Value1, ...)
Text 1	(Optional)
Text 2	(Optional)
Data Source	
Command Type	Method
Method Type	BusinessRule
Method Query	{WS}({GetDistinctValuesForField}{FieldName={  SelectedFieldName  }})
Results Table Name	DistinctValues

The syntax for referencing this object is {WS}{DataSet}{Parameter1=Value1}.

The syntax for using the Assembly Service on the Maintenance level is {WSMU}{DataSet}{Parameter1=Value1}.

### Dynamic Dashboards Service Type

Dynamic dashboards are the most unique of the service types. They let you modify components within the dashboard as well as modifying the dashboard itself.

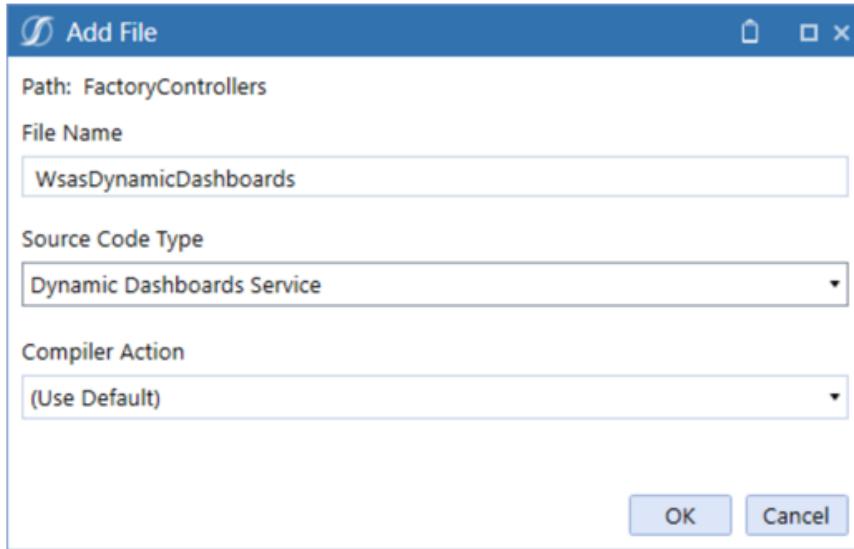
Create this file after the Service Factory file has been created and after the appropriate lines of code have been uncommented. Here is an example of a data set service type in C#:

```
29         switch (wsAssemblyServiceType)
30     {
31         case WsAssemblyServiceType.DynamicDashboards:
32             return new WsasDynamicDashboards();
33
34         //case WsAssemblyServiceType.XFBRString:
35         //    return new WsasXFBRString();
36
37         //case WsAssemblyServiceType.DataSet:
38         //    return new WsasDataSet();
39
40         //case WsAssemblyServiceType.Dashboard:
41         //    return new WsasDashboard();
42
43         //case WsAssemblyServiceType.Component:
44         //    return new WsasComponent();
45
46         //case WsAssemblyServiceType.TableView:
47         //    return new WsasTableView();
48
49         //case WsAssemblyServiceType.SqlTableEditor:
50         //    return new WsasSqlTableEditor();
51
52         default:
53             return null;
```

Given the changes in the code, you would configure the dashboard service file in the following way:

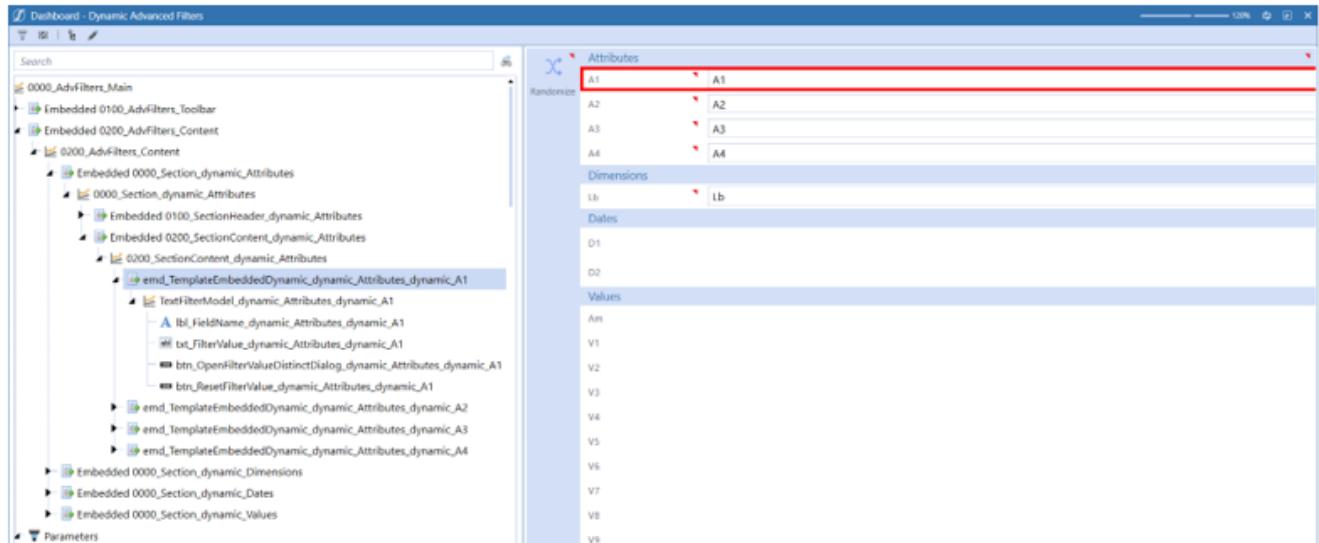
## Presenting Data With Books, Cube Views and Dashboards

---



**NOTE:** The file name should match the return function in the Service Factory file.

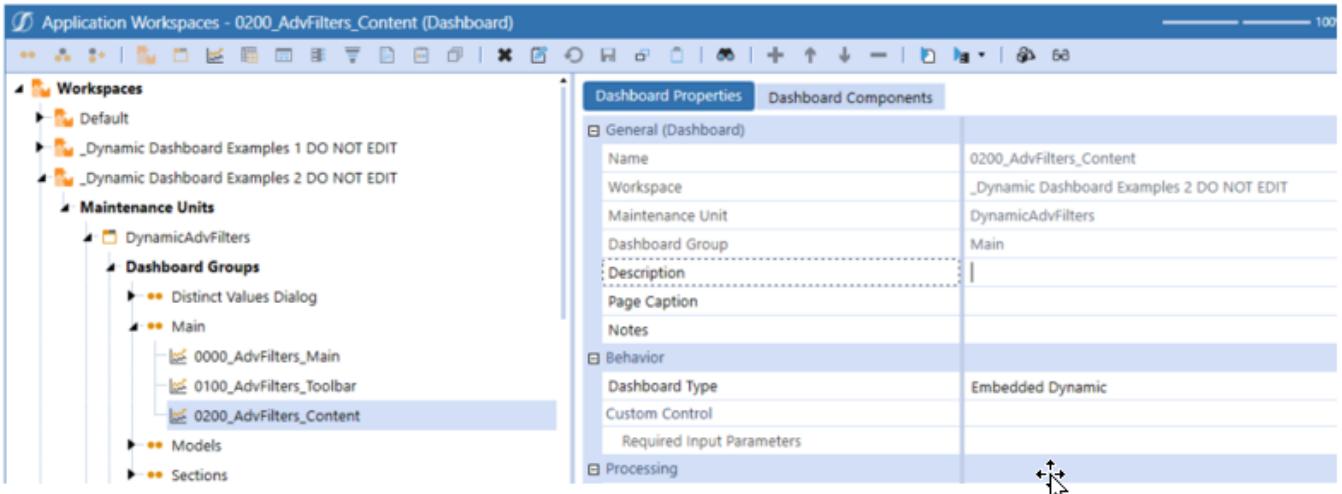
The example below shows embedded components that have not been created manually in the workspaces interface but that are generated by the assembly file.



Before writing the assembly, make sure that the Dashboard Type property is set to **Embedded Dynamic**. The rule will not run if you forget this step. This is also where you should make sure that the Workspace Assembly Service property is set to the Service Factory assembly file.

## Presenting Data With Books, Cube Views and Dashboards

---



There is no special syntax required for referencing this service type.

### SQL Table Editor Service Type

This service type lets you create `SqlTableEditorSaveData` function type in a dashboard extender rule within assemblies. You can do this either by choosing this service type or by choosing the Dashboard Extender business rule type. These rules are generally used to perform custom tasks in workspaces, such as saving actions within the SQL Table Editor component.

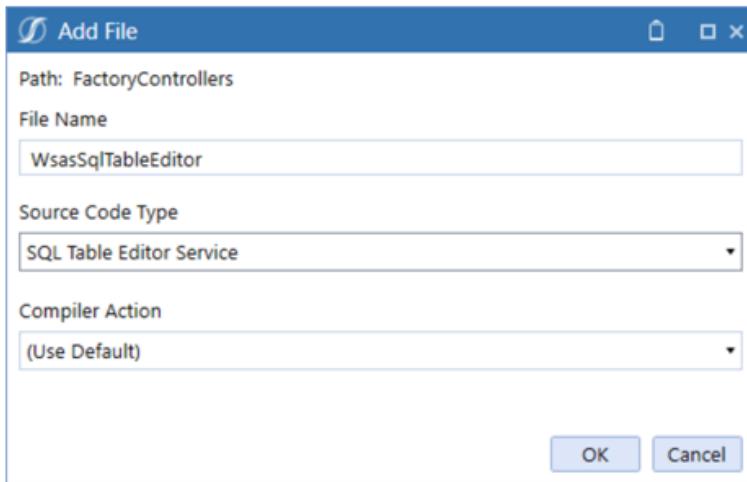
Create this file after the Service Factory file has been created and after the appropriate lines of code have been uncommented. Here is an example of a data set service type in C#:

## Presenting Data With Books, Cube Views and Dashboards

---

```
29         switch (wsAssemblyServiceType)
30     {
31         //case WsAssemblyServiceType.DynamicDashboards:
32         //    return new WsasDynamicDashboards();
33
34         //case WsAssemblyServiceType.XFBRString:
35         //    return new WsasXFBRString();
36
37         //case WsAssemblyServiceType.DataSet:
38         //    return new WsasDataSet();
39
40         //case WsAssemblyServiceType.Dashboard:
41         //    return new WsasDashboard();
42
43         //case WsAssemblyServiceType.Component:
44         //    return new WsasComponent();
45
46         //case WsAssemblyServiceType.TableView:
47         //    return new WsasTableView();
48
49     case WsAssemblyServiceType.SqlTableEditor:
50         return new WsasSqlTableEditor();
51
52     default:
53         return null;
54 }
```

Given the changes in the code, you would configure the assembly file in the following way:



**NOTE:** The file name should match the return function in the Service Factory file.

The syntax for referencing this object is {WS}{FunctionName}{Parameter1=Value1}, similar to the Dashboard service type and the Component service type.

## Presenting Data With Books, Cube Views and Dashboards

---

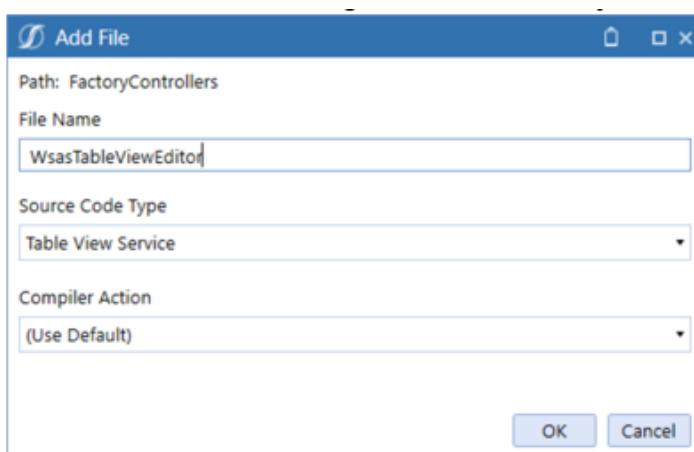
### Table View Service Type

This service type lets you create SqlTableEditorSaveData function type in a dashboard extender rule within assemblies. You can do this either by choosing this service type or by choosing the Spreadsheet business rule type. These rules query and edit data that does not live within a cube in the spreadsheet or the Excel Add-In.

Create this file after the Service Factory file has been created and after the appropriate lines of code have been uncommented. Here is an example of a data set service type in C#:

```
29         switch (wsAssemblyServiceType)
30     {
31         //case WsAssemblyServiceType.DynamicDashboards:
32         //    return new WsasDynamicDashboards();
33
34         //case WsAssemblyServiceType.XFBRString:
35         //    return new WsasXFBRString();
36
37         //case WsAssemblyServiceType.DataSet:
38         //    return new WsasDataSet();
39
40         //case WsAssemblyServiceType.Dashboard:
41         //    return new WsasDashboard();
42
43         //case WsAssemblyServiceType.Component:
44         //    return new WsasComponent();
45
46         case WsAssemblyServiceType.TableView:
47             |    return new WsasTableView();
48
49         //case WsAssemblyServiceType.SqlTableEditor:
50         //    return new WsasSqlTableEditor();
51
52         default:
```

Given the changes in the code, you would configure the assembly file in the following way:



**NOTE:** The file name should match the return function in the Service Factory file.

The syntax for referencing this object is Workspace.WorkspaceName.AssemblyName.FileName.

## Dynamic Dashboards

Dynamic dashboards give developers the ability to build and modify dashboards through coding. You use workspace assemblies, which enhances the flexibility of your dashboards and reduces the time spent in the user interface. This allows you to build more of your Solutions within workspaces.

**NOTE:** Dynamic dashboards should be created by developers.

The following procedure demonstrates how to create a new workspace for an assembly.

1. Create a new workspace. See [Workspaces Setup](#).
2. Create a maintenance unit under the new workspace.
3. Create an assembly under the new maintenance unit.  
For the new assembly, you need to [create the Service Factory](#).
4. Create a file under the new assembly.
  - a. In the **Add File** dialog box, type a name for the file in the **File Name** field. For example, WsAssemblyFactory.
  - b. Set the Source Code Type to **Service Factory**.
5. Create another file. This will be used in the Service Factory.
  - a. In the **Add File** dialog box, type a name for the file in the **File Name** field. For example, WsDynamicDashboardsService.
  - b. Set the Source Code Type to **Dynamic Dashboards Service**.
6. Open the Service Factory file that you created in Step 4 above.
  - a. Uncomment lines #31 and #32 for DynamicDashboards.
  - b. Replace the returned class with the name of the file created in Step 5 above.

```
20  namespace Workspace._WsNamespacePrefix._WsAssemblyName
21  {
22      public class WsAssemblyFactory : IWsAssemblyServiceFactory
23      {
24          public IWsAssemblyServiceBase CreateWsAssemblyServiceInstance(SessionInfo si, BRGlobals brGlobals,
25          DashboardWorkspace workspace, WsAssemblyServiceType wsAssemblyServiceType, string itemName)
26          {
27              try
28              {
29                  switch (wsAssemblyServiceType)
30                  {
31                      case WsAssemblyServiceType.DynamicDashboards:
32                          return new WsDynamicDashboardsService();
33
34                      //case WsAssemblyServiceType.XFBRString:
35                      //    return new WsasXFBRString();
36
37                      //case WsAssemblyServiceType.DataSet:
38                      //    return new WsasDataSet();
39
40      }
```

7. Navigate back to the workspace.
8. Update the Workspace Assembly Service to {WsAssemblyName}.{ServiceFactoryName}.  
For example, DynamicDashboardSBSCode.WsAssemblyFactory

## Considerations

- At the workspace level under Assemblies and Maintenance Units, there is a new setting called Workspace Assembly Service. This is the name of the assembly that you'd like to set the Service Factory to.
- In Dashboard Properties under Dashboard Types, there are two new values: Embedded Dynamic and Embedded Dynamic Repeater. Select the type of dashboard appropriate for your situation.
- In Assembly Files, when you are adding a new file, you can select Standard Class as the Source Code Type. Select Standard Class if you do not want to use Visual Basic or C# classes.
- There is a new settings section on Dashboard components under Component Properties called Processing. Here you can enter template parameter values and any text needed for processing.

## Embedded Dynamic Repeater Dashboard

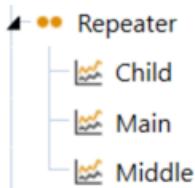
The Embedded Dynamic Repeater dashboard is a type of dashboard used to add multiple instances of the same component without having to recreate each component individually. Unlike custom controls, you are not required to make additional embedded components. An example is a dashboard that requires the following:

- Multiple labels that have differing text
- Many buttons that perform different actions
- Multiple reports and charts with slight variations in data

**NOTE:** Embedded Dynamic Repeater dashboards are not visible outside of the Workspaces page.

## Recommended Setup

To maintain organization, OneStream recommends you create your first dashboards using the Embedded Dynamic Repeater component, similar to the following image.



The Repeater group contains three dashboards:

- **Child:** This dashboard contains the components needed to create a dashboard.
- **Main:** This dashboard contains the Embedded Component of the Middle dashboard. Make sure to set the Dashboard Type property to **Top Level, Use Default, or Top Level No Parameters**. This ensures that the dashboard is visible outside the Workspaces page.
- **Middle:** This dashboard has its Dashboard Type property set to Embedded Dynamic Repeater. You will also configure other properties as well. This is where you assign the Embedded Component for the Child dashboard.

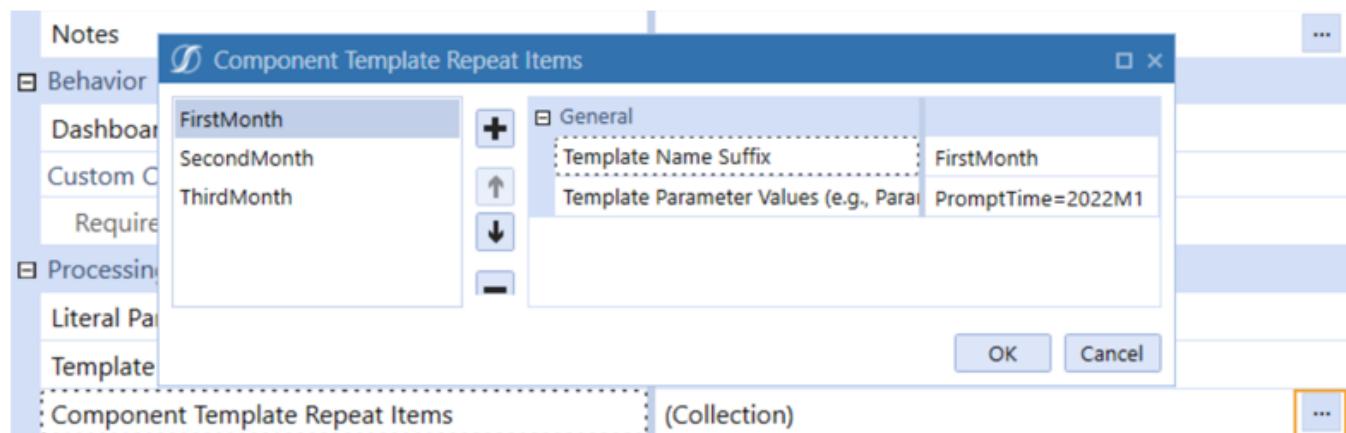
## Presenting Data With Books, Cube Views and Dashboards

---

### Component Template Repeat Items

Whichever dashboard you designate with the Embedded Dynamic Repeater dashboard type, must have a few additional properties configured. Setting the dashboard type to Embedded Dynamic Repeater will activate the Component Template Repeat Items property.

Click the **ellipses** at the right end of the Component Template Repeat Items property to repeat the component added to this dashboard and to display a dialog box where you can configure additional properties.



In the example above, three versions of the same component have been repeated: FirstMonth, SecondMonth, and ThirdMonth. You must set the following two General settings for each:

- **Template Name Suffix:** Added to the name of the dashboard when run.
- **Template Parameter Values:** Resolve any parameters in the component or data adapter.

## Presenting Data With Books, Cube Views and Dashboards

---

In the following image, the Time parameter is resolved differently in each repeated item.

The image displays three separate dashboard windows, each titled "RepeaterGrid\_dynamic\_{Month}" where {Month} is either "FirstMonth", "SecondMonth", or "ThirdMonth". Each window contains a table with columns: UniqueID, Cube, Entity, Parent, Cons, Scenario, Time, Account, Flow, Origin, IC, and UD. The "Time" column is highlighted with a yellow box in all three windows.

**RepeaterGrid\_dynamic\_FirstMonth**

UniqueID	Cube	Entity	Parent	Cons	Scenario	Time	Account	Flow	Origin	IC	UD
48b37628-578b-4dbb-a908-0021b9731dd7	Houston	Houston Heights		USD	BudgetV2	2022M1	2000_100	None	BeforeAdj	None	None
7f2df701-41eb-48f8-b315-2d5f5b6bfc2b	Houston	Houston Heights		USD	BudgetV2	2022M1	FICA %	None	BeforeAdj	None	None

**RepeaterGrid\_dynamic\_SecondMonth**

UniqueID	Cube	Entity	Parent	Cons	Scenario	Time	Account	Flow	Origin	IC	UD
3ca536e4-3115-4c6c-a611-073c3c4414b1	Houston	South Houston		USD	Actual	2022M2	10100	None	Top	Top	Top
47cd5951-a190-472f-af1e-2c12025e2af1	Houston	Houston Heights		USD	Actual	2022M2	10100	None	Top	Top	Top

**RepeaterGrid\_dynamic\_ThirdMonth**

UniqueID	Cube	Entity	Parent	Cons	Scenario	Time	Account	Flow	Origin	IC	UD
8ae105e4-2f2c-4171-b8f7-02435351ceb4	Houston	Houston Heights		USD	Variance	2022M3	EBITDAVar	Price		Form	
76cc9fc4-12aa-4854-a0b5-09ada24f4404	Houston	Houston Heights		USD	Variance	2022M3	EBITDAVar	Material Usage & Scrap		Form	

## Template Parameters

Template parameters are slightly different than the typical parameters used within OneStream. Template parameters in Embedded Dynamic Repeater dashboards use the following syntax:  
~!ParameterName!~.

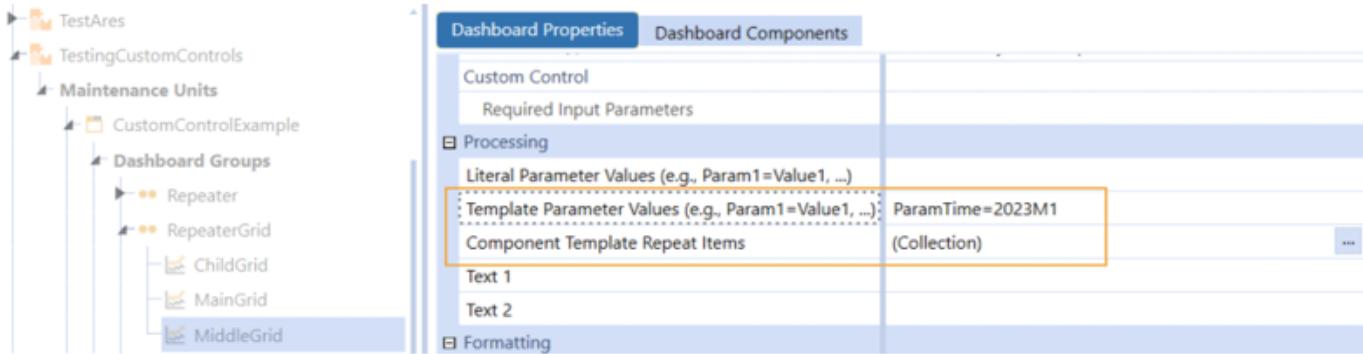
The following is an example of a template parameter used in an SQL Data Adapter:

```
SQL Query
68

Select
*
From DataAttachment
Where Time='~!PromptTime!~'
```

This parameter type does not prompt you for information, and can only be resolved through the Template Parameters property.

## Presenting Data With Books, Cube Views and Dashboards



The Component Template Repeat Items property is where you could repeat the component and vary the parameter setting each time. However, the first highlighted property in the above image is where you would resolve the parameter without repeating it.

**NOTE:** You cannot use this parameter type in cube views.

## Embedded Dynamic Repeater in Design Mode

When you are using the Embedded Dynamic Repeater dashboard in design mode, your changes to the dashboard are viewable so that you can see variations. For example, in the following image, it looks as though three separate dashboards, grid components, and data adapters have been added to the Middle Grid dashboard.

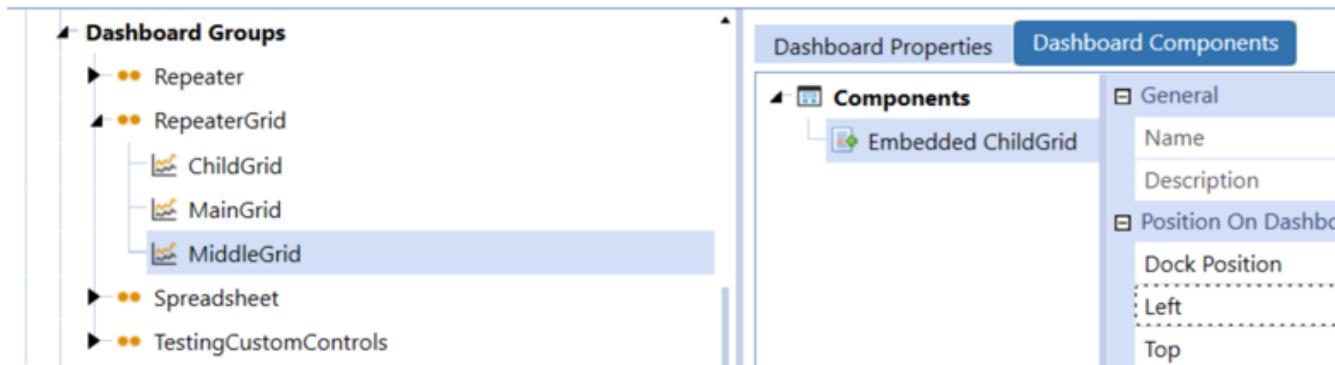
The screenshot shows the SAP BusinessObjects Designer interface with the title 'Dashboard - MainGrid'. On the left, the object browser displays a hierarchical structure of components: 'MainGrid' containing 'Embedded MiddleGrid', which in turn contains 'MiddleGrid' components for 'dynamic\_FirstMonth', 'dynamic\_SecondMonth', and 'dynamic\_ThirdMonth'. Each 'MiddleGrid' component has its own 'RepeaterGrid' and 'RepeaterDataAdapter' sub-components. On the right, three separate data grids are shown, each representing a different month: 'RepeaterGrid\_dynamic\_FirstMonth', 'RepeaterGrid\_dynamic\_SecondMonth', and 'RepeaterGrid\_dynamic\_ThirdMonth'. Each grid displays data for Houston entities across various scenarios and time periods.

RepeaterGrid_dynamic_FirstMonth							
UniqueId	Cube	Entity	Parent	Cons	Scenario	Time	Account
48b37628-578b-4dbb-a908-0021b9731dd7	Houston	Houston Heights		USD	BudgetV2	2022M1	2000_100
7f2df701-41eb-48f8-b315-2d5f5b6bfc2b	Houston	Houston Heights		USD	BudgetV2	2022M1	FICA %

RepeaterGrid_dynamic_SecondMonth							
UniqueId	Cube	Entity	Parent	Cons	Scenario	Time	Account
3ca536e4-3115-4c6c-a611-073c3c4414b1	Houston	South Houston		USD	Actual	2022M2	10100
47cd5951-a190-472f-af1e-2c12025e2af1	Houston	Houston Heights		USD	Actual	2022M2	10100

RepeaterGrid_dynamic_ThirdMonth							
UniqueId	Cube	Entity	Parent	Cons	Scenario	Time	Account
8ae105e4-2f2c-4171-b8f7-02435351ceb4	Houston	Houston Heights		USD	Variance	2022M3	EBITDAVar
76cc9fc4-12aa-4854-a0b5-09ada24f4404	Houston	Houston Heights		USD	Variance	2022M3	EBITDAVar

But if you look at the dashboard, only one embedded component was added.



This means that the view in design mode shows different variations configured through the Embedded Dynamic Repeater dashboard components.

## Find and Assign Values Using Object Lookup

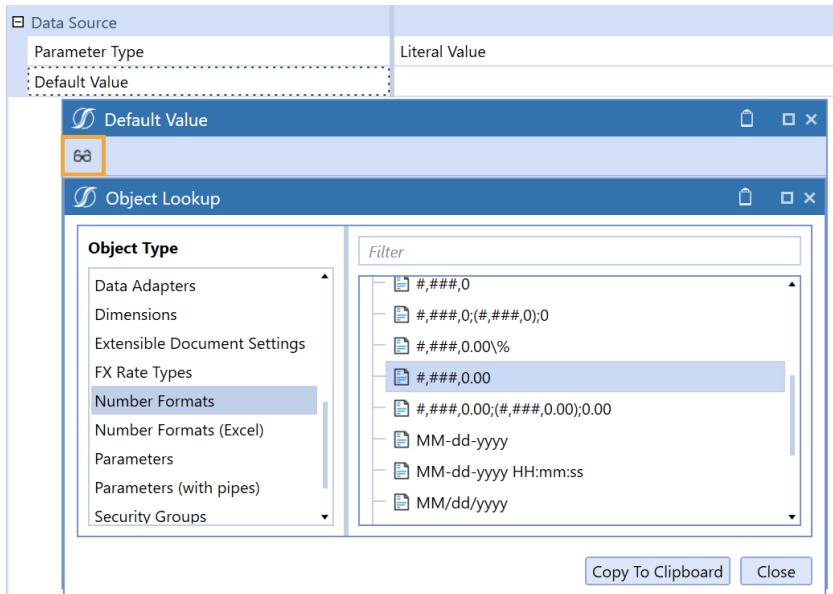
Use the **Object Lookup** dialog box throughout OneStream to search for object to assign rather than having to remember a specific object name or format. The **Object Lookup** dialog box is useful when you do not know the value or object to apply with a parameter or to display as the default value the first time a parameter runs.

In an Input Value parameter, the default value is the object or value the parameter applies when assigned to an object. For the other parameter types, the default value is the object or value to initially display when a parameter first runs. For example, to display a specific member at the top of a list of six members in a member list parameter, specify that member name as the default value.

Object Lookup enables you to browse application values and objects, such as business rules, colors, cube views, dimensions, number formats, and then copy and paste them where needed. Instead of entering a default value, use Object Lookup as follows to find and assign a value or object.

For example, to use Object Lookup to a default value to a parameter, first create the parameter, then do the following:

1. In the parameter's **Default Value**, click **Edit** and then **Object Lookup**.



2. In the **Object Lookup** dialog box, select an object type to the left to browse available objects or values. To refine the objects displayed, filter by name.
3. Select the object and click **Copy To Clipboard**.
4. In the **Default Value** dialog box, right-click, select **Paste**, and then click **OK**.

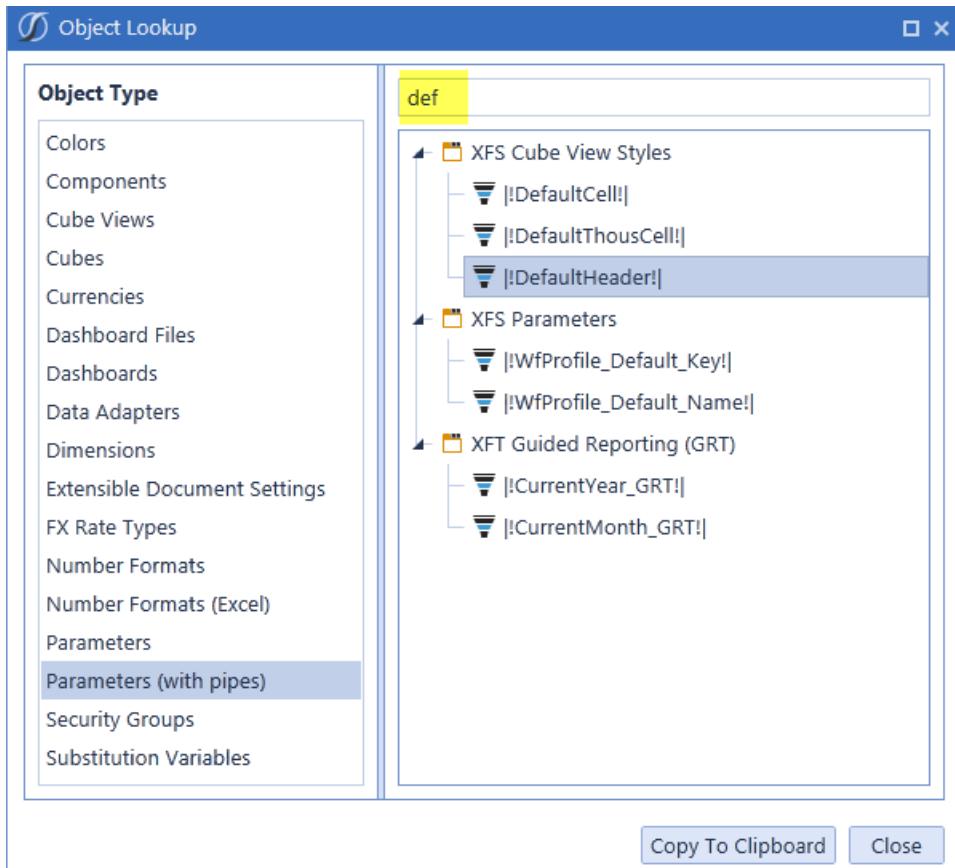
To remove or use another value or object, delete the contents in the **Default Value** dialog box and click **Object Lookup**.

## Parameters (with Pipes) Example

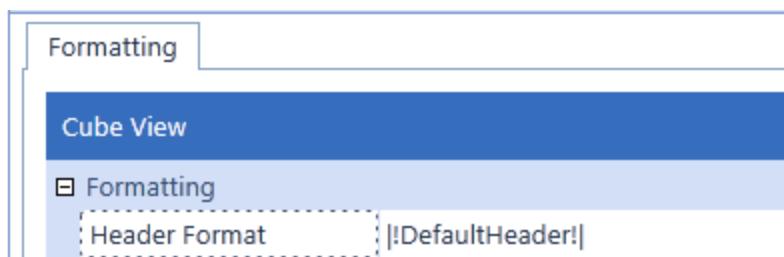
When designing a cube view header, click the **Object Lookup** icon and select **Parameters (with Pipes)** from the Object Type menu. If the name of the specific parameter is known, begin typing it in the filter at the top of the dialog box. Select the desired cube view styles parameter, such as **DefaultHeader**, and click **Copy to Clipboard** or click CTRL/double-click to copy.

## Presenting Data With Books, Cube Views and Dashboards

---



Next, click into the empty field where the copied parameter needs to go, and click CTRL+V. This pastes the parameter into the Cube View field.

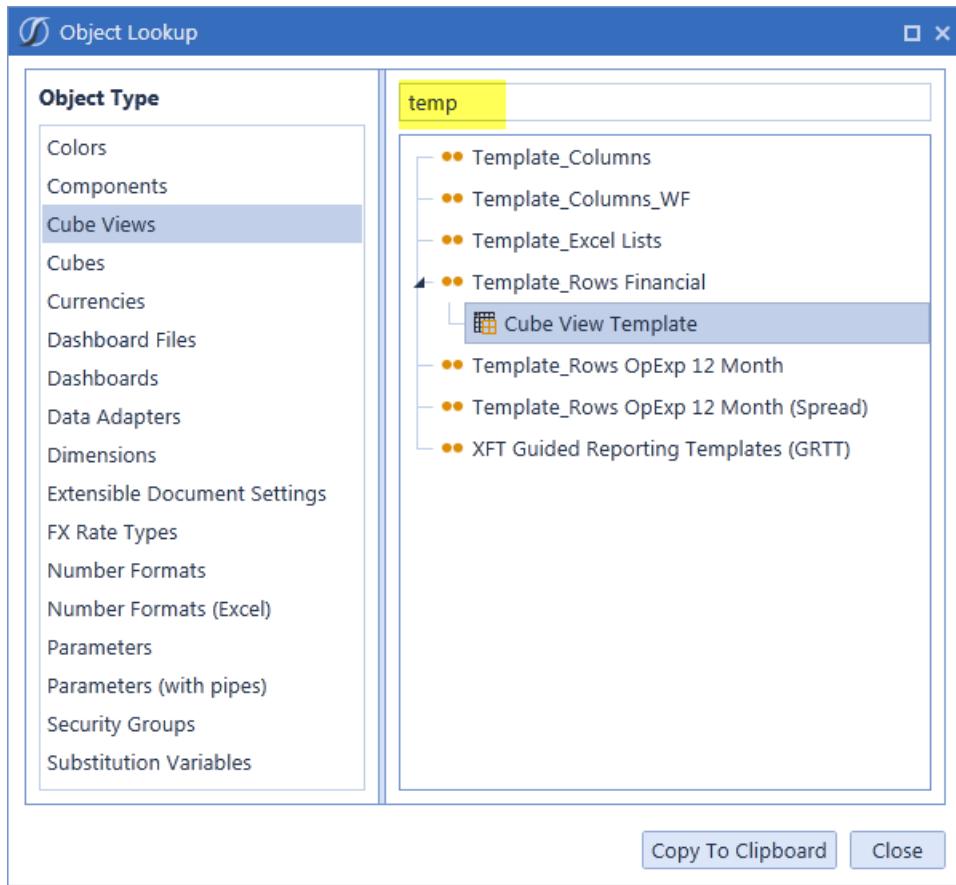


## Cube Views Example

Another example of how this feature can be used is when using Cube View Templates to create Cube View Rows or Columns. Open the Object Lookup, and begin typing the desired Cube View Template name, select the Cube View Template, and click Copy to Clipboard.

## Presenting Data With Books, Cube Views and Dashboards

---



Then paste it into the Cube View Name For Sharing All Rows property.



## Extensible Document Settings

There is a variety of item types that can be used in Extensible Documents, however each one needs to be configured a specific way for the image to process correctly at run-time. An example of each item type's configuration is provided below. For more details on how to utilize and configure these report item types in an Extensible Document, see Extensible Document Framework in "Presenting Data With Extensible Documents" on page 222.

### Insert Content in Office Image

#### Chart/Chart Report

To insert a Chart or Chart Report Dashboard Component into an Extensible Document, the following string needs to be updated and pasted into the Title field when configuring the image:

{XF}{ItemLocation}{ItemType}{ChartComponentName}

#### Example

{XF}{Application}{Chart}{Waterfall}

{XF}{Application}{ChartReport}{Waterfall}

**NOTE:**

Use this for all Chart (Advanced) Dashboard Components.

#### Cube View Report

To insert a Cube View Report into an Extensible Document, the following string needs to be updated and pasted into the Title field when configuring the image:

{XF}{Application}{ItemType}{CubeViewName}

#### Example

{XF}{Application}CubeViewReport}{BalanceSheetSummary}

#### Excel Sheet/Excel Named Range

To insert an Excel Sheet or Excel Named Range into an Extensible Document, the following string needs to be updated and pasted into the Title field when configuring the image:

{XF}{Application}{ItemType}{FilePath}

#### Example

{XF}{Application}{ExcelFile}{Documents/Users/jsmith/Favorites/VarianceReport.xfDoc.xlsx}

**NOTE:** For Excel Named Range Item Types, the Excel Named Range Name is configured in the formatting string. (e.g., ExcelNamedRange=TotalAssets)

### PDF

To insert a PDF into an Extensible Document, the following string needs to be updated and pasted into the Title field when configuring the image:

{XF}{Application}{ItemType}{FilePath}

### Example

{XF}{Application}{FileViaPDF}{Documents/Users/jsmith/Favorites/IS.pdf}

### Report

To insert a Report Dashboard Component into an Extensible Document, the following string needs to be updated and pasted into the Title field when configuring the image:

{XF}{ItemLocation}{ItemType}{ReportComponentName}

### Example

{XF}{Application}{Report}{UserTaskActivity}

## Options

The following options are available when formatting Extensible Document Image Content. Each Image Type has a formatting string which is located in the Object Lookup Dialog under Extensible Document Settings|Insert Content in Office Image. The user can copy and paste the desired formatting string into the Description field when configuring the image. If changes need to be made to the string, delete the current option and replace it with the correct one. The list below covers all the formatting options for every Image Type.

### Item Location

Application, System

### Item Type

Chart, ChartReport, CubeViewReport, ExcelFile, FileViaPDF, Report

### FillMode Options

Width, Height, LargestSide, SmallestSide

### Anchor Options

BottomCenter/Left/Right, MiddleCenter/Left/Right, TopCenter/Left/Right

### Cropping Options

This allows a user to narrow in on a portion of the image before other settings are applied. The default setting is 0 which means cropping is not being used.

CropLeft, CropTop, CropWidth, CropHeight

## XFCCell

XFCCell is a retrieve function used mainly in text documents such as Microsoft Word or PowerPoint. The specific Dimension details provided in the function obtains a single cell of data from OneStream and displays the updated value on an Extensible Document at run-time. Examples of common XFCCell formulas are provided below. See Extensible Document Framework in "Presenting Data With Extensible Documents" on page 222 for more details on how to configure XFCCells in an Extensible Document.

The following example pulls data for a specific Entity and Account:

XFCCell(E#US:A#Sales)

Additional Dimension Members can be added for more specific detail:

XFCCell(E#US:A#Sales:T#2014)

**NOTE:** Any Dimensions not specified in the formula will come from the user's POV.

Number formatting and scaling can also be added to an XFCCell.

The example below is using the number format N3 with a scaling of 6

XFCCell(E#US:A#Sales, NumberFormat=N3, Scale=6)

Culture Invariant is a default culture not associated with a specific country. It is typically used when users need to convert a number to a string, but do not want the result to be different if one user is running the string on a French PC and another user is using an English PC.

XFCCell(E#US:A#Sales, Culture=Invariant)

Include Member Scripts in XFCCells in order to retrieve data for specific Dimension Members.

Culture User is based on the computer's Windows settings and the User settings.

XFCCell(Memberscript, Culture=User, NumberFormat=N3,DisplayNoDataAsZero=True, Scale=3, FlipSign=True, ShowPercentageSign=False)

## Use Business Rules in Extensible Documents

Business Rules can also be referenced in an XFCCell function. The following details need to be updated for this to work correctly.

BR#[BRName, FunctionName=yourFunctionName, Name1=Value1, AnotherName=[AnotherValue]]

### Example

BR#[BRName=FMK\_Helper, FunctionName= GetUserSetting, SolutionID=MBL, FieldName=[Period]]

# Application Tools

Use application tools to manage how you work with the OneStream application, configuring applications to best suit your needs. Application tools include setting application security roles, setting application properties, and scheduling data management sequences. In this section you will learn how to use these tools and others to manage the application environment.

## Application Security Roles

Below are the specific Application-level security roles and what they control:

### **Administer Application**

This role allows a user to administer the application and load zip files. This is useful when multiple applications exist in one environment and different groups of administrators/users need to administer separate applications.

### **Administrator Database**

This application-level role is intended for a few people that are allowed to mass delete metadata and data, primarily using the database page.

This roleType is unlike most other roleTypes because Administrators are not automatically given access to operations that require this role.

### **Open Application**

This allows the user to see and open the application.

### **Modify Data**

This allows the user to modify data. The user is basically a read-only user throughout this application if he/she does not have this role.

### **View All Data**

This allows a group of users to view all data in the application.

### **Create Audit Attachments**

This allows the user to create data attachments for supporting documentation.

### **Create Footnote Attachments**

This allows the user to create a footnote attachment for supporting documentation.

### **Certify and Lock Descendants**

This allows the user to certify and lock descendants from the Workflow. This is typically an administrator function.

## **Application Tools**

---

### **Unlock and UnCertify Ancestors**

This allows the user to uncertify and unlock ancestors from the Workflow. This is typically an administrator function.

### **Preserve Import Data**

The administrator will lock the Workflows and then preserve imported data when changes need to be made. The Workflow can then be unlocked so changes can be made.

### **Restore Import Data**

This allows the administrator to restore imported data to the original state from the Preserve Import process.

### **Unlock Workflow Unit**

This allows a user to unlock a Workflow Unit, however, the user must also have Workflow Execution access in order to lock a Workflow Unit.

### **View Source Data Audit**

This allows a user to view the Source Data Audit Report within the Import Workflow.

### **Encrypt Business Rules**

This allows a user to Encrypt and Decrypt a rule from the Business Rule screen in the Application tab, if the user is in the role.

### **Manage Application Properties**

This allows a user to update this application's properties.

### **Manage Metadata**

This allows a user to edit metadata under the Dimension Library for this application.

### **Manage FX Rates**

This allows a user to update FX Rates.

### **Manage Data**

This allows users to manage data in all aspects included, but not limited to exporting data and clearing data completed through Data Management. This is typically an administrator function.

### **Manage Cube Views**

This allows a user to create new Cube Views and manage Cube View Groups and Profiles.

### **Manage Data Sources**

This allows a user to create new Data Sources.

### **Manage Transformation Rules**

This allows a user to create new Transformation Rules and manage Transformation Rules Groups and Profiles.

## **Application Tools**

---

### **Manage Confirmation Rules**

This allows a user to create new Confirmation Rules and manage Confirmation Rules Groups and Profiles.

### **Manage Certification Questions**

This allows a user to create new Certification Questions and manage Certification Question Groups and Profiles.

### **Manage Workflow Channels**

This allows a user to create new Workflow Channels.

### **Manage Workflow Profiles**

This allows a user to create new Workflow Profiles.

### **Manage Journal Templates**

This allows a user to create new Journal Templates and manage Journal Groups and Profiles.

### **Manage Form Templates**

This allows a user to create new Form Templates and manage Forms Groups and Profiles.

### **Manage Application Dashboards**

This allows a user to create new Application Dashboards and manage Dashboard Groups and Profiles.

### **Manage Application Database Files**

Two file systems which are stored in the Framework database (i.e., the System database) and each Application database. Users in the security roles for ManageSystemDatabaseFiles and ManageApplicationDatabaseFiles have full read and write access to his/her user folders in those two database file systems, respectively. These folders are private to the user and access is intentionally restricted to just the user and managers. Security cannot be edited for a user folder. Users can be given read and/or write access to specific folders in the database file systems using the individual folders' security settings, however this excludes access to User folders and sub-folders.

Below are the specific Application-Level User Interface Roles and what they control:

#### **Application Load Extract Page**

This gives access to the Load/Extract screen located in |Application |Tools|. This is typically restricted to administrators.

#### **Application Properties Page**

This gives access to the Application Properties screen located in |Application |Tools|. This is typically restricted to administrators.

#### **Application Security Roles Page**

This gives access to the Application Security screen located in |Application |Tools|. This is typically restricted to administrators.

## Application Tools

---

### **BookAdminPage**

This gives access to the Book Designer screen located in |Application|Presentation|. This is typically restricted to administrators, or any users who create Report Books.

### **Business Rules Page**

This gives access to the Business Rules screen located in |Application|Tools|. This is typically restricted to administrators.

### **Certification Questions Page**

This gives access to the Certification Questions screen located in |Application| Workflow|. This is typically restricted to administrators.

### **Confirmation Rules Page**

This gives access to the Confirmation Rules screen located in |Application|Workflow|. This is typically restricted to administrators.

### **Cube Admin Page**

This gives access to the Cube Admin screen located in |Application|Cube|. This is typically restricted to administrators.

### **Cube Views Page**

This gives access to the Cube Views screen located in |Application|Presentation|. This is typically restricted to administrators, or any users who create Cube Views.

### **Dashboard Admin Page**

This gives access to the Dashboard Admin screen located in |Application|Presentation|. This is typically restricted to administrators.

### **Data Management Admin Page**

This gives access to the Data Management Admin screen located in |Application|Tools|. This is typically restricted to administrators.

### **Data Sources Page**

This gives access to the Data Sources screen located in |Application|Data Collection|. This is typically restricted to administrators.

### **Dimension Library Page**

This gives access to the Dimension Library screen located in |Application|Cube|. This is typically restricted to administrators.

### **FX Rates Page**

This gives access to the FX Rates screen located in |Application|Cube|. This is typically restricted to administrators.

### **Form Templates Page**

This gives access to the Form Templates screen located in |Application|Data Collection|. This is typically restricted to administrators.

### Journal Templates Page

This gives access to the Journal Templates screen located in |Application |Data Collection|. This is typically restricted to administrators.

### Transformation Rules Page

This gives access to the Transformation Rules screen located in |Application |Data Collection|. This is typically restricted to administrators.

### Workflow Channels Page

This gives access to the Workflow Channels screen located in |Application |Workflow|. This is typically restricted to administrators.

### Workflow Profiles Page

This gives access to the Workflow Profiles screen located in |Application |Workflow|. This is typically restricted to administrators.

**NOTE:** Click  and begin typing the name of the Security Group in the blank field. As the first few letters are typed, the Groups are filtered making it easier to find and select the desired Group. Once the Group is selected, click CTRL and Double Click. This will enter the correct name into the appropriate field.

# Application Properties

This is where default properties are set for the application and for properties that differ by Scenario Type.

## General Properties

### Global Point of View

These are enabled when forcing Global Scenario and Global Time through Transformation settings. An initial value should be configured even if the Transformation setting is not being used.

#### Global Scenario

This is the default Scenario users will see when looking at Workflow.

#### Global Time

This is the default Time users will see when looking at Workflow.

# Company Information

### Company Name

Place company name here for it to appear on automatically generated reports from Cube Views.

### Logo File (png, height ~ 50 pixels)

Attach a logo file in order for it to appear on Cube Views and Reports. An image format of PNG is required approximately 50 pixels high.

# Workflow Channels

### UD Dimension Type for Workflow Channels

The Origin Dimension controls data load, but in some cases other User Defined Dimensions require their own layer of locking. For example, a company plans by Entity by Product. One Entity can have five products done by different people. Each channel can be locked separately to protect that layer of data instead of locking the entire Entity.

# Formatting

### Number Format

This shows the format for numeric values displayed throughout the application. Configure to show additional degrees of precision to the right of the decimal point. This setting can be overridden through Cube View formatting.

#### N0

This setting will not show any decimals or zeroes

#### N1-N6

These settings will show X amount of decimals. If N2 is chosen, two decimals are displayed, N5 will display five decimals, etc.

**NOTE:** The N in the above settings indicates that these settings are international.

The following formats use 10000.001 as an example.

#,###,0\%

This returns 10,000% and -10,000%

#,###,0

This returns 10,000 and -10,000

The three sections in a number format, separated by semi-colons, represent the format for positive numbers, negative numbers and zeros.

## Application Tools

---

#,###,0;(#,##,0);0

This returns 10,000 and (10,000)

#,###,0.00%

This returns 10,000.00% and -10,000.00%

#,###,0.00

This returns 10,000.00 and -10,000.00

#,###,0.00;(#,##,0.00);0.00

This returns 10,000.00 and (10,000.00)

**NOTE:** In order to vertically align positive and negative numbers in reports where parenthesis are used for negative values, include trailing spaces in the positive number format. This will account for the trailing parenthesis used by negative numbers.

Example: #,###,0.00 ;(#,##,0.00);0.00

**NOTE:** Click  , select the desired format, and click CTRL and Double Click. This will enter the correct format into the appropriate field.

## Currencies

All currencies used in the application must be listed here in order to be used on the Entity, to do any translation of currency, or enter any rates. The list of currencies will include any available currencies that are pre-Euro, or phased out currencies for historical data loading purposes. If a now defunct or new currency is not listed and is needed for the application, call OneStream Support.

## Transformation

### Enforce Global POV

If set to True, this will enforce the current Global Scenario and Time setting for all users, so they cannot change their Workflow View. If the Global POV is enforced,  will display on the Import task during the Workflow.

## Application Tools

---

### Allow Loads Before Workflow View Year

If set to True, this will enforce the current Workflow View setting to allow data loading to time periods prior to the current Workflow year. If this is set to False, data cannot be imported to time

periods prior to the current Workflow year and  will display on the Import task during the Workflow.

### Allows Loads After Workflow View Year

If set to True, this will enforce the current Workflow setting to allow data loading to time periods after the current Workflow year. If this is set to False, data cannot be imported to time periods

after the current Workflow year and  will display on the Import task during the Workflow.

## Certification

### Lock after Certify

If set to True, this will auto lock after certification in the Workflow.

## Dimension Properties

General	Dimensions	Standard Reports
<input type="checkbox"/> Time Dimension		
Start Year		2009
End Year		2020
<input type="checkbox"/> User Defined Dimensions		
UD1 Description		Cost Centers
UD2 Description		Product
UD3 Description		Sales Channel
UD4 Description		Customers
UD5 Description		
UD6 Description		
UD7 Description		
UD8 Description		

## Time Dimension

### Start Year

The starting year of the application.

### End Year

The ending year of the application.

## User Defined Dimensions (Descriptions)

Applies a custom description to the eight User Defined dimensions. The description applies to the dimension type, not to each dimension and will be visible to the user in various interfaces in the OneStream Application. The descriptions display in the hover/tool-tips, pop-up windows and other dimension interfaces. These are viewable in:

- Point of View
- Dimension Library
- Cube View Member Filters
- Drill Down Dimension headers
- Excel Add-in / Spreadsheet
- Journals

### UD1-8 Description

Enter a generic description that best describes the purpose of each User Defined dimension. See example above.

## Standard Reports

These settings will be applied with auto-generating a report from a Cube View.

## Logo

### Height

Enter a numerical value to determine the Height of the report. (e.g. 105 pixels)

### Bottom Margin

Enter a numerical value to determine the Bottom Margin size.

### Title

**Top Margin**

Enter a numerical value to determine the Top Margin size.

**Font Family**

The font displayed in the Title of the report.

**Font Size**

Enter a numerical value to determine the size of the font.

**Bold**

If set to True, the Title will be bold in the report.

**Italic**

If set to True, the Title will be in italics in the report.

**Text Color**

Use the ellipsis icon to choose a text color for the report Title.

### Header Labels

This is where the default Header Label properties are defined for all the reports in the application.

**Top/Bottom Margin**

Enter a numerical value to determine the Top/Bottom Margin size.

**Font Family**

The font displayed in the Header Labels of the report.

**Font Size**

Enter a numerical value to determine the size of the font.

**Bold**

If set to True, the Header Labels will be bold in the report.

**Italic**

If set to True, the Header Labels will be in italics in the Report.

**Text Color**

Use the ellipsis icon to choose a text color for the Header Labels.

### Header Bar

**Background Color**

Use the ellipsis icon to choose a Header Background color.

### Line Color

Use the ellipsis icon to choose a Header Line color.

## Footer

### Text

An open field to enter footer text.

### Font Family

The font displayed in the Footer.

### Font Size

Enter a numerical value to determine the size of the font in the Footer.

### Show Line

If set to True, the report will show a line in the Footer.

### Show Date

If set to True, the report will show the date in the Footer.

### Show Page Numbers

If set to True, the report will show page numbers in the Footer.

### Line Color

Use the ellipsis icon to choose the color of the line in the Footer.

### Text Color

Use the ellipsis icon to choose a text color in the Footer.

## Business Rules

Business Rules contain calculation logic configured to run against different parts of an application. These rules are compiled with VB.NET or C# code and are created within a rich integrated development environment, or IDE.

Download the OneStream API Overview Guide and OneStream API Details and Database Documentation from MarketPlace for detailed Business Rule engine background, an API guide and information on each database related to OneStream.

There are several areas in the product using the exact same rule syntax and applying it to how data is imported, how the Cubes are calculated, and other operations. Once this syntax is understood, logic can be written.

There are three ways to write this calculation logic:

### Business Rules

Business Rules are found under the Application Tab|Tools. There are nine types of Business Rules as shown below. These can be stored, secured, and then assigned to multiple areas of the product with the ability to re-use them.

### Complex Expression

This logic can be created as a Business Rule or as a Complex Expression from within an application artifact such as a Data Source. The syntax is the same with the only difference being that a Business Rule can be shared across multiple application artifacts where a Complex Expression is contained within the artifact.

### Member Formula

The same Business Rule syntax can be applied to Member Formulas as well. This logic stays with the Member and cannot be shared.

There are also three utility groups available when writing Business Rules:

#### BR API

BR API provides application programming interface to commonly used functions involving all areas of the product where a Business Rule can be used.

#### API

The more specific API provided as a Parameter to a Business Rule provides functions specific to the type of Business Rule being written. For example, when implementing a Business Rule for a finance-oriented task, API refers to the functions used for processing calculation logic and other capabilities related to processing a Cube's data and metadata.

#### ARGS

An argument represents the value passed to a Business Rule when the procedure is called and the calling code supplies the arguments. For example, if a Parser Business Rule is assigned to the Account Dimension, args will supply the Account Dimension data as well as a set of functions available to use against that data. Different args will be provided depending on the type of Business Rule used.

## Business Rule Encryption (Decryption)

There is also functionality to Encrypt and Decrypt Business Rules with password protection when writing and saving Business Rules.

The authorized user can Encrypt Business Rule by clicking the encrypt button.

- KO\_SecUsers\_GetStatusByID
- KO\_SecUsers\_GetStatusByName
- KO\_SecUsers\_LogonInactivityThresholdForAdmin

## Application Tools

Once clicked, an Encrypt Business Rule dialog will display prompting the user for a password. The user will create and enter the password and then Re-Enter the password in the box below and click OK. The system encrypts the Business Rule, displays “Business Rule Is Encrypted” message text in the editor and the editor is in read only mode.

**NOTE:** It is important to remember and record the password being used for each Business Rule being encrypted or the Business Rule will not be able to be decrypted for further changes without the assistance of OneStream Support.

To Decrypt a Business Rule, an authorized user can Decrypt Business Rule by clicking the

Decrypt button  . At this time, the Decrypt Business Rule dialog box will display prompting the user for a password. User will enter the password and click OK. The system will decrypt the Business Rule, display the Business Rule text editor and then return to read/write mode again.

Set the proper access to Encrypt Business Rules to allow an authorized user to Encrypt and Decrypt a rule from the Business Rule screen in the Application tab, if the user is in the role.

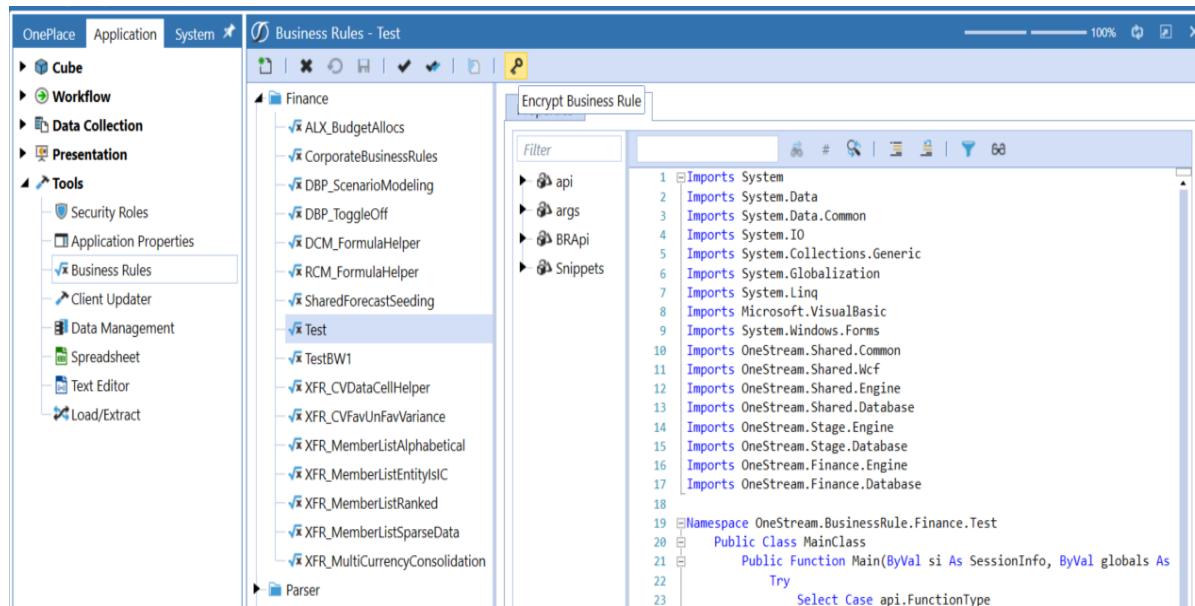
OnePlace	Application	System	
<b>Cube</b>			
Dimensions			Administrators
Time Profiles			Nobody
Cubes			Everyone
FX Rates			Everyone
<b>Workflow</b>			
Workflow Channels			Administrators
Workflow Profiles			Administrators
Confirmation Rules			Administrators
Certification Questions			Administrators
<b>Data Collection</b>			
Data Sources			Administrators
Transformation Rules			Administrators
Form Templates			Administrators
Journal Templates			Administrators
<b>Presentation</b>			
Books			Everyone
Cube Views			Administrators
Dashboards			Administrators
<b>Tools</b>			
Security Roles			Administrators
Application Properties			Everyone
Business Rules			Everyone
Client Updater			Everyone
Data Management			Everyone
Spreadsheet			Everyone
Text Editor			Everyone
Load/Extract			Everyone

Next advance to the Business Rules section (Application Tab>>Tools>>Business Rules) and Refresh the screen and the Encrypt Business Rule button will appear in the menu.

- ✓ KO\_SecUsers\_GetStatusById
  - ✓ KO\_SecUsers\_GetStatusByName
  - ✓ KO\_SecUsers\_LogonInactivityThresholdForAdmin

## Application Tools

Select the Business Rule to be encrypted, then select the Encrypt Business Rule button.



At this time, an Encrypt Business Rule dialog will display prompting the user for a password.



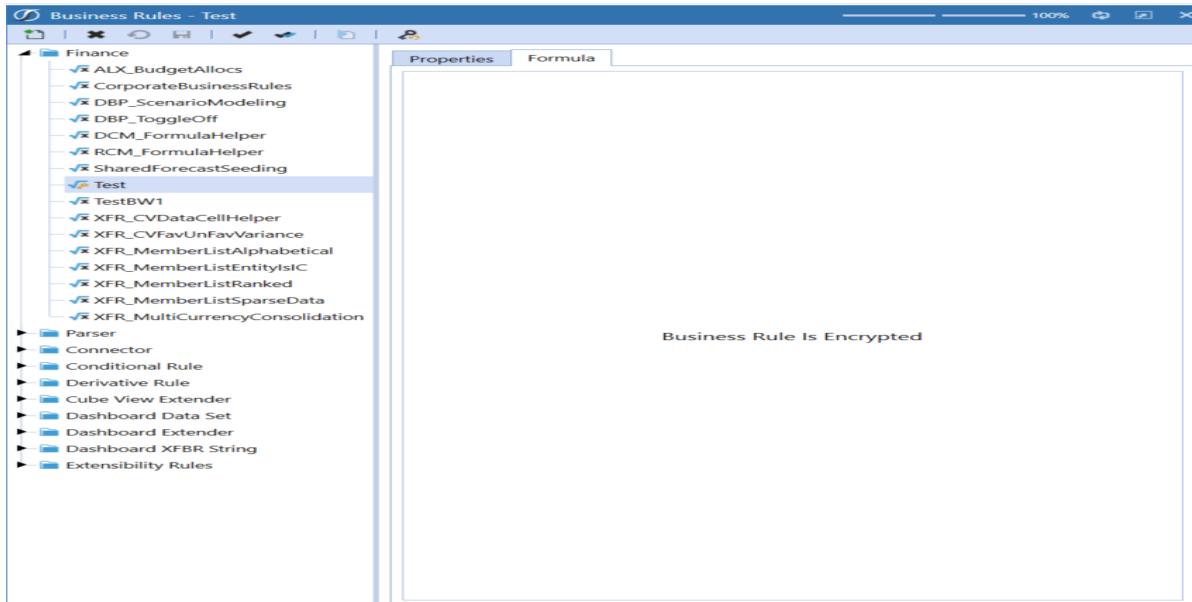
The user will create and enter the password, then Re-Enter the password in the box below and click OK.

## Application Tools

---



The system encrypts the Business Rule, displays “Business Rule Is Encrypted” message text in the editor and the editor is in read only mode and the Decrypt Business Rule button is now displayed in the menu bar.

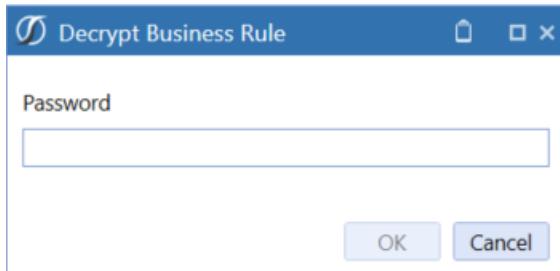


## Application Tools

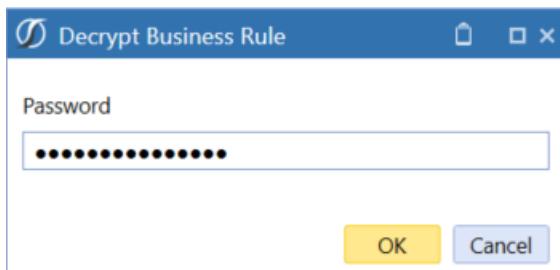
---

To Decrypt a Business Rule, an authorized user can Decrypt Business Rule by clicking the

Decrypt button  . At this time, the Decrypt Business Rule dialog box will display prompting the user for a password.

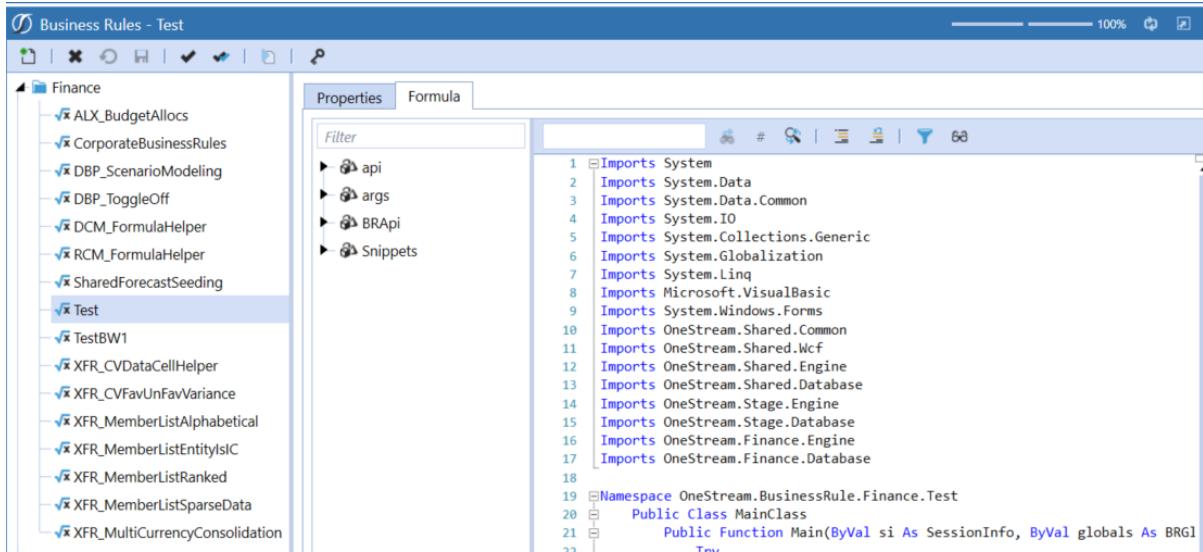


User will enter the password and click OK.



The system Decrypts the Business Rule displays the Business Rule text editor and the Business Rule is in read/write mode again.

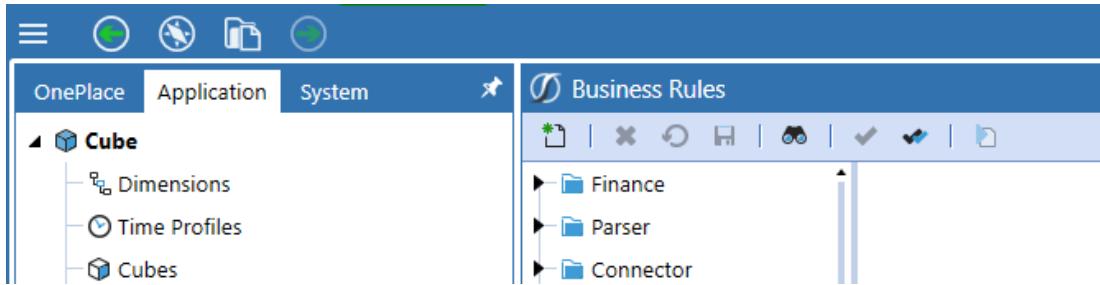
## Application Tools



## Business Rule Search

Find a business rule quickly by performing a search instead of scrolling through business rules.

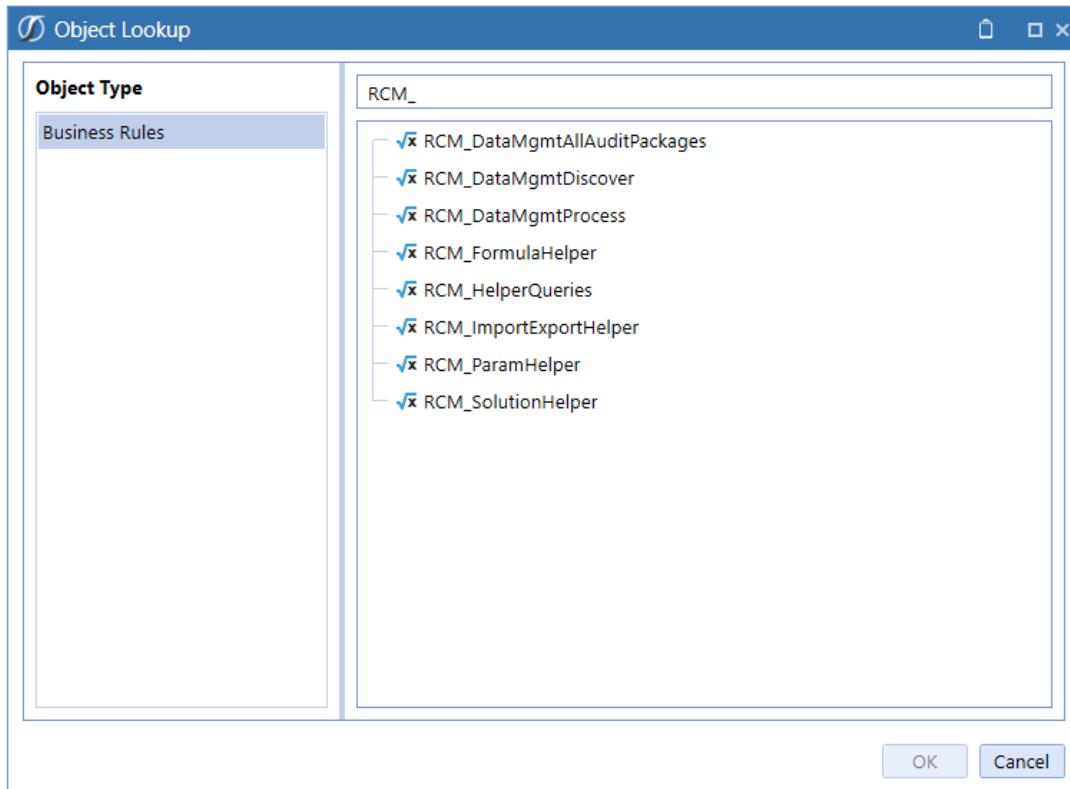
1. Select the Application tab.
2. From Tools, select Business Rules.



3. Click Business Rule Search. The Object Lookup dialog box opens.
4. Start typing the start of the business rule name. All business rules that match your text display in the list.

## Application Tools

---



5. Select the business rule from the list and click **OK**.

## Business Rule Toolbar and Hotkeys

### **Compile Business Rule to Check Syntax**

Use this to compile a selected Business Rule in order to check its syntax.

### **Compile All Business Rules and Formulas to Check Syntax**

Use this to compile all Business Rules and Formulas in order to check syntax. This is most commonly used after installing a software upgrade where some Rule or Formula syntax may have changed. This feature is only available for application administrators.

### **Execute Extender**

Use this to run the selected Extender Business Rule

## Application Tools

---

```
└─✓x KO_SecUsers_GetStatusByID  
└─✓x KO_SecUsers_GetStatusByName  
└─✓x KO_SecUsers_LogonInactivityThresholdForAdmin
```

### Encrypt Business Rule

Use this to Encrypt the Business Rule formula. Clicking this button will be prompt the user to enter a password in the Encrypt Business Rule dialog box to complete the encryption process.



### Decrypt Business Rule

Use this to Decrypt the Business Rule formula. Clicking this button will be prompt the user to enter a password in the Decrypt Business Rule dialog box to complete the encryption process.

### Ctrl+M

Expand /Collapse all regions and methods. Click in the script after selecting the Business Rule in order to use the hotkey.

## Business Rule Client Image Types

Client images are specified via a XFClientImageType Class in a Business Rule. These images can display on certain Dashboard Parameter Components such as Buttons, Maps, SQL Table Editors, Grid Views, etc.

The Image Types are as follows:

- StatusGrayBall
- StatusWhiteBall
- StatusOrangeBall
- StatusBlueBall
- StatusRedBall
- StatusLightGreenBall
- StatusGreenBall
- StatusGrayCheckMark
- StatusGreenCheckMark

- StatusLockedWithCheckMark
- StatusLockedWithFolder

# Business Rule Properties

## General

### Name

The name of the Business Rule

### Type

The type of Business Rule (see below for a detailed description of each Business Rule Type)

### Contains Global Functions for Formulas

Set this to **True** to write a function in a **Business** Rule and then call the function from Member Formulas or other Business Rules. This is helpful when the same code must be copied to multiple Member Formulas and instead of using the same complicated code, a Public function with two lines of code written in a Business Rule can be called. Only use this setting for Business Rules referenced from Member Formulas because it does incur some overhead when the system needs to compile Member Formulas.

For more details on this feature, see "About the Financial Model" on page 2.

### Referenced Assemblies

Enter a list of referenced assembly names separated by semi-colons.

### Reference Another Business Rule

To reference another Business Rule, enter BR\ followed by the other Business Rule name.

**NOTE:** A rule may reference either a VB.NET or C# rule.

### BR\SharedFunctionsBR

Example:

```
Dim sharedFinanceBR As New  
OneStream.BusinessRule.Finance.SharedFinanceFunctions.MainClass  
Dim myResult As String = sharedFinanceBR.Test(si, api, args)
```

For more details on this feature, see Referencing a Business Rule from a Member Formula or Business Rule in "About the Financial Model" on page 2.

### Reference a DLL File

DLL files can be stored in the File Share's Business Rule Assembly Folder, an Application Server Configuration File, or in a common Network Share Folder to which numerous Application Servers can reference. To reference a DLL File, enter XF\ followed by the DLL file name.

### XF\ThirdPartyFunctions.dll

Otherwise, use no prefix and enter the full path and file name of any DLL file stored on the application server(s) file system.

### Is Encrypted

This will be set to True if the Business Rule has been encrypted or False if it has not been encrypted.

## Security

### Access

Members of this group will have access to the Business Rule

### Maintenance

Members of this group have the authority to maintain the Business Rule

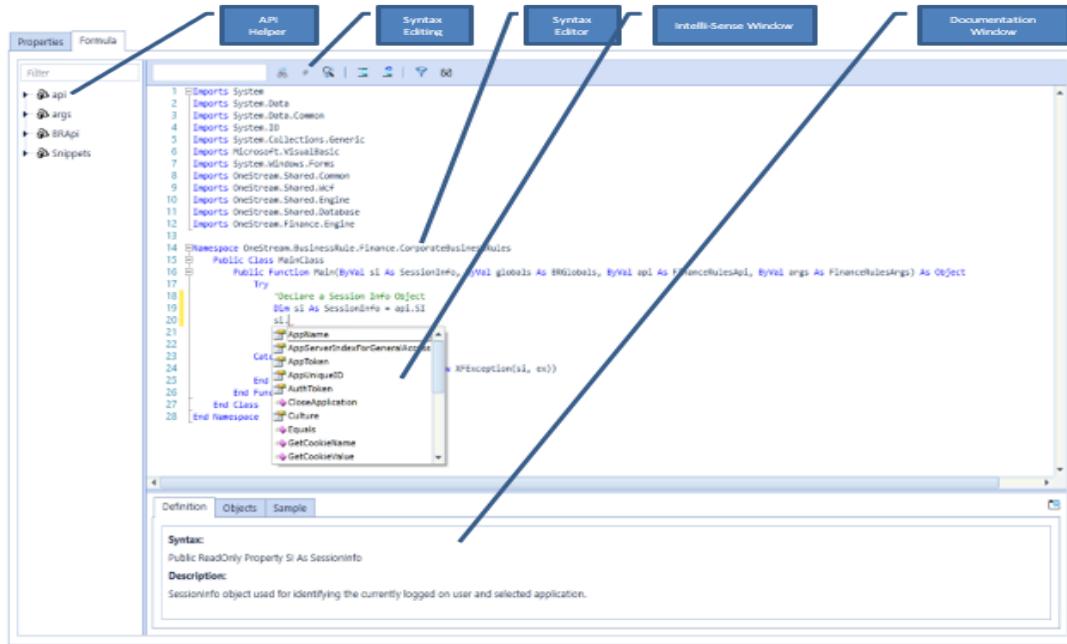
**NOTE:** Click  and begin typing the name of the Security Group in the blank field. As the first few letters are typed, the Groups are filtered making it easier to find and select the desired Group. Once the Group is selected, click CTRL and Double Click. This will enter the correct name into the appropriate field.

## Business Rules Editor Overview

The OneStream Business Rule editor is a powerful in-solution screen that provides integrated API context help, syntax editing with intelli-sense, and full outlining capabilities. The actual syntax content and Business Rule structure will be discussed at length in subsequent sections of this document.

The image below explains the major regions and elements of the Business Rule editor.

## Application Tools



## Business Rule Types

### Finance Rule

Finance Business Rules are used to generate multi-Dimensional calculations. These Business Rules are written as Shared Business Rules and applied to a Cube or Member Formulas. Eight of these can be assigned to run at the Cube level during consolidations.

### Example APIs

#### api.FunctionType

The expression used when special logic needs to be run and a certain process needs to be isolated.

### Finance Function Type

#### Calculate

Additional logic during calculation of Entity, Consolidation Scenario and Time. This sets the value of one or more values (left side of Formula) equal to another (right side). It then executes a calculation for a specifically qualified Point of View. This is the most common function used.

## Application Tools

---

### Translate

Additional logic that uses custom translation.

### FXRate

Custom logic used to determine Foreign Exchange rates for any intersection.

### Consolidate Share

Additional logic used during the custom calculation of the Share Member.

### Consolidate Elimination

Additional logic used during the custom calculation of the Elimination Member.

### Calculation Drill Down Member Formula

Provides custom drill down results.

### Conditional Input Rule

Conditional Input Rules make data cells read-only. While the settings for this can be done directly on the Cube, using a Conditional Input Business Rule offers more flexibility and still allows the use of the Cube settings. This rule can return the following: ConditionalInputResultType.Default, ConditionalInputResultType.NoInput, ConditionalInputResultType.NoInputAllowCellDetail, and ConditionalInputResultType.NoCellDetailAllowInput.

The following Business Rule example will make all cells for the Account 6000 read-only. This should be added to a Business Rule attached to a Cube.

```
Case Is = FinanceFunctionType.ConditionalInput
  If api.Pov.Account.Name.XFEqualsIgnoreCase("6000") Then
    Return ConditionalInputResultType.NoInput
  End If
  Return ConditionalInputResultType.Default
```

### Confirmation Rule

Special logic that runs with Confirmation Rules.

### Data Cell

Named GetDataCell calculations that can be reused such as a Better/Worse calculation in Cube Views.

### Dynamic Calc Account

Special logic to use in Dynamic Calc members.

### Member List and Member List Headers

A custom list of members for use in Cube Views and other areas. See Commonly Used Member Filter Functions in "Cubes" on page 328 for more details on using custom lists in a Cube View.

# Finance Business Rule Example

```
Namespace OneStream.BusinessRule.Finance.CorporateBusinessRules
Public Class MainClass
    Public Function Main(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As FinanceRulesApi, ByVal args As FinanceRulesArgs) As Object
        Try
            Select Case api.FunctionType
                Case Is = FinanceFunctionType.Calculate
                    'Populate Headcount Flow
                    If api.pov.Scenario.name = "Actual" Then
                        api.data.calculate("A#Headcount:F#BegBalHC=A#Headcount:F#None:T#POVPrior1")
                        api.data.calculate("A#Headcount:F#EndBalHC=A#Headcount:F#None")
                        api.data.calculate("A#Headcount:F#CheckHC=A#Headcount:F#BegBalHC+A#Headcount:F#TotalMovementHC-A#Headcount:F#EndBalHC")
                    End If
                    If api.pov.Scenario.name = "Variance" Then
                        api.data.Calculate("F#BudgetEBITDA:A#[EBITDAVar] = A#EBITDA:S#Budget:F#None")
                        api.data.Calculate("F#ActualEBITDA:A#[EBITDAVar] = A#EBITDA:S#Actual:F#None")
                        api.data.Calculate("F#[Bad Debt]:A#[EBITDAVar] = (A#54100:S#Budget:F#None-A#54100:S#Actual:F#None)")
                        api.data.Calculate("F#VarianceCheckSum:A#[EBITDAVar]=F#BudgetEBITDA:A#[EBITDAVar]+F#VarianceEBITDA:A#[EBITDAVar]-F#ActualEBITDA:A#[EBITDAVar]")
                    End If
                End Select
            Return Nothing
        Catch ex As Exception
            Throw ErrorHandler.LogError(si, New XFEException(si, ex))
        End Try
    End Function
End Class
End Namespace
```

# Parser Rule

Parser Business Rules are used to evaluate and/or modify field values being processed by the Stage Parser Engine as it reads source data. These Business Rules are written as Shared Business Rules or Logical Expressions and applied to a Data Source Dimension.

## Example API

### **args.Line**

This will return the entire record being processed from the Data Source.

### **args.Value**

This will return the Dimension the Business Rule is assigned to in the Data Source.

## Common Usage

- Custom parsing logic
- Field value concatenation
- Field value bypassing
- Evaluate field other than current field being parsed

## Parser Business Rule Example

```
Namespace OneStream.BusinessRule.Parser.XFR_ParseDelimitedTextAndGetField
    Public Class MainClass
        '
        'Reference Code:      XFR_ParseDelimitedTextAndGetField
        'Description:        Parse a delimited string and return a selected field from the delimited list.
        'Usage:              Parser Business Rule intended to be used on any field within a Data Source.
        '
        Public Function Main(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As ParserDimension, ByVal args As ParserArgs) As Object
            Try
                'Parse the line
                Dim fields as List(Of String) = StringHelper.SplitString(args.Line, " ", StageConstants.ParserDefaults.DefaultQuoteCharacter)
                If fields.Count >= 2 Then
                    'Return the 2nd field (Zero based index)
                    Return fields(1)
                Else
                    'Line invalid, just return incoming value and let parser engine handle the evaluation
                    Return args.Value
                End If
            Catch ex As Exception
                Throw ErrorHandler.LogWrite(si, New XFEException(si, ex))
            End Try
        End Function
    End Class
End Namespace
```

## Connector Rule

Connector Business Rules are used to communicate with, collect data from, and drill back to external systems. These Business Rules are written as Shared Business Rules and applied to a Data Source.

See Connectors in "Collecting Data" on page 132 for more information on using Connectors.

### Example API

#### **args.ActionType**

This will return one of the four available Connector action types.

#### **ConnectorActionTypes.GetFieldList**

This will run the SQL query to return the available field list to the Data Source for Dimension assignment.

#### **ConnectorActionTypes.GetData**

This will run the SQL query to retrieve the source data and return it to the Stage based on the Data Source Dimension assignment.

#### **ConnectorActionTypes.GetDrillBackTypes**

This will return a list of the specified drill back types. (e.g., File or Data Grid)

#### **ConnectorActionTypes.GetDrillBack**

This will run the required SQL query against the source system and will provide greater detail than what was originally imported.

## Connector Business Rule Example

Namespace OneStream.BusinessRule.Connector.RevenueMgmtHouston

## Application Tools

---

```
Public Class MainClass
    Public Function Main(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Transformer, ByVal args As ConnectorArgs) As Object
        Try
            'Get the query information
            Dim connectionString As String = GetConnectionString(si, globals, api)

            'Get the Field name list or load the data
            Select Case args.ActionType
                Case Is = ConnectorActionTypes.GetFieldList
                    'Return Field Name List
                    Dim fieldListSQL As String = GetFieldListSQL(si, globals, api)
                    Return api.Parser.GetFieldNameListForSQLQuery(si, DbProviderType.OLEDB,
                        connectionString, true, fieldListSQL, false)

                Case Is = ConnectorActionTypes.GetData
                    'Process Data
                    Dim sourceDataSQL As String = GetSourceDataSQL(si, globals, api)
                    api.Parser.ProcessSQLQuery(si, DbProviderType.OLEDB, connectionString,
                        true, sourceDataSQL, false, api.ProcessInfo)
                    Return Nothing

                Case Is = ConnectorActionTypes.GetDrillBackTypes
                    'Return the list of Drill Types (Options) to present to the end user
                    Return Me.GetDrillBackTypeList(si, globals, api, args)

                Case Is = ConnectorActionTypes.GetDrillBack
                    'Process the specific Drill-Back type
                    Return Me.GetDrillBack(si, globals, api, args, args.DrillBackType.DisplayType,
                        connectionString)
            End Select

            Catch ex As Exception
                Throw ErrorHandler.LogWrite(si, New XFEException(si, ex))
            End Try
        End Function

        'Create a Connection string to the External Database
        Private Function GetConnectionString(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Transformer) As String
            Try
                'Named External Connection In App Server Config
                '_____
                Return "RevenueMgmtSystem"

                Catch ex As Exception
                    Throw ErrorHandler.LogWrite(si, New XFEException(si, ex))
                End Try
            End Function

            'Create the field list SQL Statement
            Private Function GetFieldListSQL(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Transformer) As String
                Try
                    'Create the SQL Statement
                    Dim sql As New Text.StringBuilder

                    sql.Append("SELECT Top(1)")
                    sql.Append("TransID, PlantCode, CustId, CustName, InvNo, InvYear, InvMonth, InvDesc, GLAccount,
                        WorkDay, ProdModel, BomCode, UnitPrice, Units, Amount, DestinationCode ")

                    sql.Append("FROM InvoiceDocumentDetail ")


```

## Application Tools

---

```
Return sql.ToString

Catch ex As Exception
    Throw ErrorHandler.LogError(si, New XFEException(si, ex))
End Try
End Function

'Create the data load SQL Statement
Private Function GetSourceDataSQL(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Transformer) As
String
    Try
        'Create the SQL Statement
        Dim statement As New Text.StringBuilder
        Dim selectClause As New Text.StringBuilder
        Dim fromClause As New Text.StringBuilder
        Dim whereClause As New Text.StringBuilder
        Dim orderByClause As New Text.StringBuilder

        selectClause.Append("SELECT ")
        selectClause.Append("TransID, PlantCode, CustId, CustName, InvNo, InvYear, InvMonth, InvDesc,
GLAccount, WorkDay, ProdModel, BomCode, UnitPrice, Units, Amount, DestinationCode ")

        fromClause.Append("FROM InvoiceDocumentDetail ")

        whereClause.Append("WHERE ")
        'Get the YEAR from the current XF Workflow Unit TimeKey
        whereClause.Append("(")
        whereClause.Append("InvYear = " &
TimeDimHelper.GetYearFromId(api.WorkflowUnitPk.TimeKey).ToString)
        whereClause.Append(")")

        'Get the MONTH from the current XF Workflow Unit TimeKey
        whereClause.Append(" And ")
        whereClause.Append("(")
        whereClause.Append("InvMonth = 'M' &
TimeDimHelper.GetSubComponentsFromId(api.WorkflowUnitPk.TimeKey).Month.ToString & "")")
        whereClause.Append(")")

        'Select Houston Plant Codes
        whereClause.Append(" And ")
        whereClause.Append("(")
        whereClause.Append("PlantCode IN('H200','H210')") )
        whereClause.Append(")")

        orderByClause.Append("ORDER BY ")
        orderByClause.Append("PlantCode, CustId, WorkDay, ProdModel, DestinationCode")

        'Create the full SQL Statement
        statement.Append(selectClause.ToString)
        statement.Append(fromClause.ToString)
        statement.Append(whereClause.ToString)
        statement.Append(orderByClause.ToString)

        Return statement.ToString

    Catch ex As Exception
        Throw ErrorHandler.LogError(si, New XFEException(si, ex))
    End Try
End Function

'Create the drill back options list
Private Function GetDrillBackTypeList(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Transformer, ByVal
args As ConnectorArgs) As List(Of DrillBackTypeInfo)
    Try
        'Create the SQL Statement
```

## Conditional Rule

Conditional Rules (mapping) are used to conditionally evaluate mapping criteria during the data transformation process. These Business Rules are written as Shared Business Rules or Logical Expressions and applied to a Transformation Rule definition.

They are only applicable to Transformation Rules with the type of Composite, Range, List, or Mask either as a Business Rule or Complex Expression.

## Example API's

### **args.GetSource**

This will return the source value for the specified Dimension.

### **args.GetTarget**

This will return the mapped or transformed value for the specified Dimension.

### **args.OutputValue**

This will return the originally mapped target value from the Transformation Rules.

## Conditional Business Rule Example 1

Conditional Rule for Dimension Member mapping based on source values. Note that if the Business Rule does not call args.OutputValue, the target field can be empty and will not be considered.

```
Namespace OneStream.BusinessRule.ConditionalRule.XFR_Conditional_Source
    Public Class MainClass
        '
        'Reference Code:      XFR_Conditional_Source
        'Description:        Demonstrates how to map a flow Dimension based on the source account.
        'Usage:              Used for conditional flow Member mapping based on source account.
        '
        Public Function Main(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Transformer, ByVal args As
        ConditionalRuleArgs) As Object
            Try
                'Grab the source account value
                Dim account As String = args.GetSource("A#")

                'Assign the flow Member based on the source account
                If account.StartsWith("1190") Then
                    Return "Additions"
                Else
                    'Just return value assigned to the Transformation Rule
                    Return args.OutputValue
                End If

                Return Nothing
            Catch ex As Exception
                Throw ErrorHandler.LogError(si, New XFEException(si, ex))
            End Try
        End Function
    End Class
End Namespace
```

# Conditional Business Rule Example 2

Conditional Rule for Dimension Member mapping based on target values.

```
Namespace OneStream.BusinessRule.ConditionalRule.XFR_Conditional_Target
    Public Class MainClass
        '
        'Reference Code:      XFR_Conditional_Target
        'Description:        Demonstrates how to map a flow Dimension based on the target account.
        'Usage:              Used for conditional flow Member mapping based on target account.
        '

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

                'Grab the target account value
                Dim account As String = args.GetTarget("A#")

                'Assign the flow Member based on the first character of the target account
                Select Case Left(account,1)
                    Case "1","2","3"
                        Return "End_Inp"
                    Case Else
                        Return "None"
                End Select

                Catch ex As Exception
                    Throw ErrorHandler.LogError(si, New XFFException(si, ex))
                End Try
            End Function
        End Class
    End Namespace
```

# Derivative Rule

Derivative Rules (derive data prior to mapping) are used to evaluate and/or calculate values during the data derivation process. These Business Rules are written as Shared Business Rules or Logical Expressions and applied to a Derivative Rule definition.

They are only applicable to Transformation Rules with the type of Derivative either as a Business Rule or Complex Expression.

# Common Usage

- Calculate mathematical expressions
- Lookup value from transformation cache for use in calculations

## Application Tools

---

- Lookup value from Cube for use in calculations
- Source system check rule logic (validation rules on source data)

## Derivative Business Rule Example

```
Namespace OneStream.BusinessRule.DerivativeRule.XFR_ExecuteStageCacheQuery
    Public Class MainClass
        '
        'Reference Code:      XFR_ExecuteStageCacheQuery
        '
        'Description:          Retrieves a value from the transformation cache calculated by another derivative rule
                             and uses the value as an input to calculate the ratio of one account to another.
        '
        'Usage:                Derivative Business Rule intended to demonstrate how to retrieve a value from the
                             transformation cache and use it as input to a calculation.
        '
        Public Function Main(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Transformer, ByVal args As
DerivativeRuleArgs) As Object
            Try
                'Get Cash Value by querying the cache
                Dim valueCash As Decimal = Decimal.Zero
                Using dtCash As DataTable = api.QueryDataCache(si, "Ac = 'Cash'", False, True)
                    If Not dtCash Is Nothing Then
                        For Each row As DataRow In dtCash.Rows
                            valueCash = row("Am")
                        Exit For
                    Next
                End If
            End Using

            'The current value is AR|
            Dim valueAR As Decimal = args.ColumnValue

            'Return the ratio of AR/Cash
            If (valueAR <> Decimal.Zero) And (valueCash <> Decimal.Zero) Then
                Return (valueAR / valueCash)
            Else
                'Just return the value from the Stage
                Return args.ColumnValue
            End If
            Return Nothing
        Catch ex As Exception
            Throw ErrorHandler.LogError(si, New XFEException(si, ex))
        End Try
    End Function
End Class
End Namespace
```

# Cube View Extender

Cube View Extender Rules are used to apply advanced Cube View formatting to any Cube View while in Report view.

**NOTE:** These rules do not apply to how a Cube View looks like in the Data Explorer Grid view.

The Extender Rule is used in conjunction with the Execute Cube View Extender Business Rule setting on the Cube View. See Cube View Extender: Advanced Cube View Formatting in "Implementing Security" on page 252 for examples on how to use this rule. See the OneStream API Overview Guide as well as the OneStream API Details and Database Documentation Guide for more details on this Business Rule.

## Common Usage

The following are key uses for Cube View Extender Business Rules in formatting Reports.

- Alter Headers and Footers
  - Display different logos on select reports based on conditional logic or security and manage their placement and size
  - Customize the page number in the header or footer  
Page numbers can be on the top or bottom row of a report and the horizontal position can be specified for rows. This only applies to the top or bottom rows.
  - Format individual header and footer fields
  - Control the Left, Right, Center Subtitle widths
  - Control the font size of Title and Subtitles
  - Customize the date display
- Alter Column and Row Headers
  - Customize bottom text alignment
  - Control Word Wrap
- Apply Conditional Formatting to Data Cells

- Format cells based on their contents
- Change the text color of a value in order to effectively hide the result

## Dashboard DataSet

DashboardDataSet Rules are used to create programmatic query results. This rule type combines multiple types of data into a single result set using the full syntax capability of VB.Net. These Business Rules are written as Shared Business Rules and applied to Dashboard Data Adapters or Dashboard Parameters.

### Common Usage

- Combine different types of data for a report
- Build programmatic data queries (e.g., analytic plus sql)
- Conditionally build data query reports
- Conditionally build data query for Parameters
- Create geographical data to display via a Map Parameter Component in a Dashboard
- Create a data series to display via a Chart (Advanced) Parameter Component in a Dashboard

## Dashboard Extender

DashboardExtender Rules are used to perform a variety of tasks associated with custom Dashboards and MarketPlace Solutions. These Business Rules can be thought of as multi-purpose rules and make up the majority of the code written in a MarketPlace Solution. In addition, they are written as Shared Business Rules and applied to Application Dashboard Parameter Components (Buttons, Combo Boxes, etc.).

### Common Usage

- Execute task when the user clicks a button
- Perform a task and show a message to the user
- Perform a custom calculation

- Upload a file from the end user's machine
- Automate a Workflow
- Build a custom Workflow
- Create custom data tables
- Include Page State to store parameters and values about a specific Dashboard page instance

## Dashboard XFBRString

Dashboard XFBRString Rules are used to process conditional Dashboard Parameters. These rules inspect and alter a Dashboard Parameter value using the full syntax capabilities of VB.Net or C#. Dashboard String Functions are written as Shared Business Rules and called by using a XFBR (BusinessRuleName, FunctionName, UserParam=[UserValue]) specification anywhere a standard Dashboard Parameter is used. After a Dashboard BRString rule is created, create a Dashboard Component to call the BRString using the following script.

```
BRString(brRuleName, funcNameWithinBRRule, optionalName1 = var1, optionalName2 = var2)
```

Enter the Business Rule as a Parameter using `||!BRString!|`

The return value from the Business Rule will be used in the Dashboard Component.

**NOTE:** This Business Rule can be applied to any Dashboard or Cube View property where a Parameter is used.

## Extensibility Rule

Extensibility Rules have these two types: Extender and Event Handlers. Extender Rules are the most generalized type of Business Rule in the OneStream platform. Use these to write a simple utility function or a specific helper function called as part of a Data Management Job. Event Handlers are exclusively called before or after a certain operation occurs within the system.

## Extensibility Business Rule Example

Event Handler that sends an email notification after a ProcessCube event.

## Application Tools

---

```
Namespace OneStream.BusinessRule.DataQualityEventHandler.DataQualityEventHandler
Public Class MainClass
    'Reference Code: DataQualityEventHandler
    'Description: Event handler method that provides an opportunity to supplement a normal data
    '             quality action with your own custom functionality.
    '             (Example: email after ProcessCube or publish report to SharePoint after
    '             failed confirmation).
    'Usage: Executes when a Data Quality action is run and fires this Business Rule. If you have
    '       written code in that handles the specified event operation the code will be executed.
    '-----
    Public Function Main(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Object, ByVal args As
    DataQualityEventArgs) As Object
        Try
            'Define a switch to control event processing, since many of these are reference examples we do not
            'want them to run all the time
            Dim processEvents As Boolean = False

            'Set the default return values
            Dim returnValue As Object = args.DefaultReturnValue
            args.UseReturnValueFromBusinessRule = False
            args.Cancel = False

            'Evaluate the operation type in order to determine which subroutine to process
            Select Case args.OperationName
                Case Is = BREventOperationType.DataQuality.ProcessCube.NoCalculate
                    'Execute a Data Management job after process Cube runs
                    Me.XFR_HandleProcessCubeNoCalculate(si, globals, api, args)

                Case Is = BREventOperationType.DataQuality.Certify.FinalizeSetCertifyState
                    'Send an email after a Workflow Profile executes its certification
                    Me.XFR_HandleFinalizeSetCertifyState(si, globals, api, args)
            End Select
        Return returnValue
        Catch ex As Exception
            Throw ErrorHandler.LogError(si, New XFEException(si, ex))
        End Try
    End Function

    #Region "ProcessCube.NoCalculate Helpers"

    Private Sub XFR_HandleProcessCubeNoCalculate(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Object,
    ByVal args As DataQualityEventArgs)
        'Reference Code: XFR_HandleProcessCubeNoCalculate
        'Description: Run a Data Management Sequence after the Workflow process Cube task is
        'run.
        'Note: the Data Management Sequence name is assigned to a
        'Workflow Profile CalcDef filter field
        'so this event does not have to be modified, the user can simply
        'edit the CalcDef grid
        'for a Workflow Profile and this Business Rule will execute the
        'specified sequence.
        'Usage: Used to supplement the standard "ProcessCube" functionality associated
```

## Application Tools

```

with a Workflow Profile by allowing a Data Management Sequence to be
executed for the Workflow Profile as well.
-----
```

---

```

Try
    'Get the DataUnitInfo from the Event arguments so that we can get the name of the
    DataManagement sequence to process.
    Dim calcinfo As DataUnitInfo = DirectCast(args.Inputs(2), DataUnitInfo)
    If Not calcinfo Is Nothing Then
        'Make sure that a Sequence name was assigned to the filter value of the Calc Definition of
        the executing Workflow Profile
        If calcinfo.FilterValue <> String.Empty Then
            'Now, execute the DataMgmt Sequence that was specified in the FilterValue
            '(In a background thread)
            BRApi.Utilities.StartDataMgmtSequence(si, calcinfo.FilterValue, Nothing)
        End If
    End If

    Catch ex As Exception
        Throw ErrorHandler.LogError(si, New XException(si, ex))
    End Try
End Sub

#End Region

#Region "FinalizeSetCertifyState Helpers"
    'Apply tag this to Text1 of users that should be notified when a child Workflow certifies
    Public const NotifyCertifyCompleted as String = "#NCC"

    Private Sub XFR_HandleFinalizeSetCertifyState(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Object,
        ByVal args As DataQualityEventArgs)
        '
        'Reference Code:      XFR_HandleFinalizeSetCertifyState
        '
        'Description:          Send an email to users assigned the Workflow execution group of the Parent
        Workflow Profile of the Workflow Profile that just certified.
        '
        'Usage:                Used to provide email notification when a dependent Workflow Profile
        certifies.
        '
    Try
        'check the before / after flag, we want to handle the AFTER event
        If (args.IsBeforeEvent = False) Then
            'Check to see if the profile was certified, add send an email all Parents of the certifying
            profile
            Dim certifyInstance As CertifyInstanceStateInfo = DirectCast(args.Inputs(2),
                CertifyInstanceStateInfo)
            If Not certifyInstance Is Nothing Then
                If certifyInstance.IsCertified(si) Then
                    'Get the list of users assigned to the Parent of the Workflow
                    profile so that we know who should get an email
                    Dim notificationUsers As List(Of UserInfo) =
                        BRApi.Workflow.General.GetUsersInWorkflowGroupForParents(
                            si, certifyInstance.Workflow.WfUnitPk.CreateWorkflowUnit(
                                ClusterPk()), SharedConstants.WorkflowProfileAttributeIndexes,
                                WorkflowExecutionGroup, True, True, NotifyCertifyCompleted)
                    For Each userSummary As UserInfo In notificationUsers
                        'Send the email
                        me.CreateMessageAndSendMail(si, certifyInstance,
                            userSummary)
                    Next
                End If
            End If
        End If
    End Sub

```

## Application Tools

---

```
End if

End if

Catch ex As Exception
    Throw ErrorHandler.LogError(si, New XFEException(si, ex))
End Try
End Sub

Private Sub CreateMessageAndSendMail(ByVal si As SessionInfo, certinstance as CertifyInstanceInfo, userSummary as
UserInfo)
    'Reference Code: CreateMessageAndSendMail
    '
    'Description: Create the email message body and execute the "Send Mail" function.
    '
Try
    Using dbConnApp as DbConnInfo = BRApi.Database.CreateApplicationDbConnInfo(si)

        Dim emailInfo as new Text.StringBuilder
        emailInfo.AppendLine("To: " & userSummary.User.Email)

        'Create the message title
        Dim messageTitle as new Text.StringBuilder
        messageTitle.Append("OneStream Workflow [")
        messageTitle.Append(certinstance.WorkflowProfileName)
        messageTitle.Append(", ")
        messageTitle.Append(ScenarioDimHelper.GetNameFromId(dbConnApp,
certinstance.Workflow.WfUnitPk.ScenarioKey))
        messageTitle.Append(", ")

        messageTitle.Append(TimeDimHelper.GetNameFromId
(certinstance.Workflow.WfUnitPk.TimeKey))
        messageTitle.Append(") is ready for you to review.")
        messageTitle.AppendLine("")

        'Create the message body
        Dim messageBody as new Text.StringBuilder
        messageBody.AppendLine("Certification Information:")
        messageBody.AppendLine("-----")
        messageBody.AppendLine("Profile Name..... " & certinstance.WorkflowProfileName)

        Dim signOff as CertifySignOffInstanceInfo = Nothing
        If certinstance.SignOffsInDescendingTimestampOrder.Count > 0 Then
            signOff = certinstance.SignOffsInDescendingTimestampOrder(0)
            messageBody.AppendLine("User Name..... " & signOff.UserName)

            messageBody.AppendLine("Time Completed..... " &
signOff.TimeStampToLocalTime & " Local Time")

            messageBody.AppendLine("Comments..... " & signOff.Comments)
        End If
        messageBody.AppendLine("")

        messageBody.AppendLine("Dependent Information:")
        messageBody.AppendLine("-----")

        For Each relativeInfo as CertifyRelativeInfo in certinstance.Dependants
            messageBody.AppendLine(relativeInfo.WorkflowProfile.Name & " (" &
relativeInfo.Workflow.GetOverallStatusText("Not Started", "Completed",
"Locked", "Running", "", & ")")
        Next
        messageBody.AppendLine("")
    End Using
End Sub
```

## Application Tools

---

```
'Test the email by writing a log message
ErrorHandler.LogMessage(si, emailInfo.ToString & vbCrLf & messageTitle.ToString &
vbcrlf & messageBody.ToString)

'Send an email message
'Me.SendMail(si, userSummary.User.Email, messageTitle.ToString,
messageBody.ToString)

End Using

Catch ex As Exception
    Throw ErrorHandler.LogError(si, New XFEException(si, ex))
End Try
End Sub

Private Sub SendMail(ByVal si As SessionInfo, ByVal userEmailAddress As String, ByVal subject As String, ByVal
messageBody As String)
    'Reference Code:      SendMail
    '
    'Description:          Create the mail connection and send.
    '

Try
    'Prepare the message
    Dim emailConnectionName As String = "Office365Mail"
    Dim toEmail As New List(Of String)
    toEmail.Add("userEmailAddress")
    Dim subject As String = "Subject"
    Dim body As String = "Body"
    Dim attachments As New List(Of String)
    attachments.Add ("FileLocationString")

    'Send the Message
    BRApi.Utilities.SendMail(si, emailConnectionName, toEmail, subject, body, attachments)

    Catch ex As Exception
        Throw ErrorHandler.LogError(si, New XFEException(si, ex))
    End Try
End Sub

#End Region
|
End Class
End Namespace
```

## Extender

This can be used to automate custom tasks like running external scripts, backups and report publishing.

### WCF Event Handler

This allows direct interaction with the Microsoft Windows Communication Foundation which means it listens to communication between the client and the web server. The rule will intercept the communication, analyze it, and if certain criteria is met, it will run its logic. This is quite flexible and has a variety of uses such as creating, reading, deleting, and updating different types of objects in the system for users in a group or Transformation Rule changes. For example, a rule can be created to e-mail an auditor about every metadata change as it happens.

## Available operations

### Transformation Event Handler

This can be run at various points from Import through Load. Available operations.

## Application Tools

---

- StartParseAndTransForm
- InitializeTransformer
- ParseSourceData
- LoadDataCacheFromDB
- ProcessDerivativeRules
- ProcessTransformationRules
- DeleteData
- DeleteRuleHistory
- WriteTransformedData
- SummarizeTransformedData
- CreateRuleHistory
- EndParseAndTransForm
- FinalizeParseAndTransForm
- StartRetransForm
- EndRetransForm
- FinalizeRetransForm
- StartClearData
- EndClearData
- FinalizeClearData
- StartValidateTransForm
- ValidateDimension
- EndValidateTransForm
- FinalizeValidateTransForm

## Application Tools

---

- StartValidateIntersect
- EndValidateIntersect
- FinalizeValidateIntersect
- LoadIntersect
- StartLoadIntersect
- EndLoadIntersect
- FinalizeLoadIntersect

## Journals Event Handler

This can be run before, during, or after a Journal operation such as Submission, Approval, or Post. Available operations:

- SubmitJournal
- ApproveJournal
- RejectJournal
- PostJournal
- UnpostJournal
- StartUpdateJournalWorkflow
- EndUpdateJournalWorkflow
- FinalizeUpdateJournalWorkflow

## Save Data Event Handler

This is run in order to track all save events in an application.

## Forms Event Handler

This can be run before, during, or after an operation such as Form Save. Available operations:

- SaveForm
- CompleteForm
- RevertForm
- StartUpdateFormWorkflow
- EndUpdateFormWorkflow
- FinalizeUpdateFormWorkflow

## Data Quality Event Handler

This can be run before, during, or after data quality events like Confirmation and Certification.  
Available operations:

- StartProcessCube
- Calculate
- Translate
- Consolidate
- EndProcessCube
- FinalizeProcessCube
- PrepareICMatch
- StartICMatch
- PrepareICMatchData
- EndICMatch
- StartConfirm
- EndConfirm
- FinalizeConfirm
- SaveQuestionResponse

## Application Tools

---

- StartSetQuestionairreState
- SaveQuestionairreState
- EndSetQuestionairreState
- StartSetCertifyState
- SaveCertifyState
- EndSetCertifyState
- FinalizeSetCertifyState

## Data Management Event Handler

This can be run before or after a Data Management Sequence or Step runs. Available operations:

- StartSequence
- ExecuteStep
- EndSequence

## Workflow Event Handler

This can be run before or after a Workflow execution step. Available operations:

- UpdateWorkflowStatus
- WorkflowLock
- WorkflowUnlock

## Client Updater

The Client Updater retrieves updated software from the OneStream server for the Excel Add-In client program when the version being used does not match the version found on the server being connected. To update, the user needs to be able to write to the installation folder. From this page, the user can first save work and then restart using elevated Windows privileges using the Restart OneStream as Windows Administrator button.

## Application Tools

---

Information about the current version displays at the top of the window, as well as the module being reviewed, the location of the installation folder and the version status all appear within this window.

Example of versions that match:

Current XF Version: **4.2.0.8203**

Client Module

OneStream Excel AddIn

Installation Folder: C:\Program Files (x86)\OneStream Software\OneStreamExcelAddIn

OneStream Excel AddIn XF Version: **4.2.0.8203**

**OneStream Excel AddIn is up to date.**

Example of versions that do not match:

The screenshot shows the 'Client Updater' application interface. At the top, it displays 'Current XF Version: 4.2.0.8203'. Below this, under 'Client Module', it lists 'OneStream Excel AddIn'. It also shows the 'Installation Folder' as 'C:\Program Files (x86)\OneStream Software\OneStreamExcelAddIn'. The 'OneStream Excel AddIn XF Version' is shown as '4.2.0.8204'. A red warning message at the bottom states: 'OneStream Excel AddIn is not up to date. Please press the Update button to download new XF installation files.' A blue 'Update' button is visible at the bottom left.

### Client Module

Use this field to select which application (OneStream Excel Add-In) to compare to the current version of each that are currently installed on the user's desktop. Click the selection arrow and select the appropriate application. If the versions do not match, click **Update** and then **OK**.

**NOTE:** Close any open versions of Excel before clicking OK to proceed.

**NOTE:** A backup folder with files for the outdated version is automatically created and saved as part of the update process. It can be found in the same location as the newly updated version folder.

If this functionality has been disabled, the following message appears when trying to update those applications. "The Client Updater has been disabled by your System Administrator. Please use OneStream's full client installation program, or see your System Administrator."

## Data Management

The Data Management module allows you to copy data or clear data for a Cube, Scenario, Entity, and Time. In order to do this, each Data Management Group must contain a Sequence and a Step. The Steps are created and specifically defined by Cube, Scenario, Entity, and Time. Once the Step is defined and saved, it can be assigned to a Sequence. Once these Groups are created, they are organized into Data Management Profiles.

### Search

1. Click the binoculars to open the Select Data Management Object dialog box.
2. Select the Object type from the drop-down: All Items, Sequence, or Step.
3. Enter as much information as you have and click Search.
4. The search results show.
5. If you want to view the results by hierarchy, click View in Hierarchy.

## Data Management

A Data Management Group allows you to create different groups each containing Sequences and Steps. A Data Management Group can be assigned to multiple Data Management Profiles.

Click  to create a new Data Management Group.

### General

**Name**

The name of the new data group.

**Description**

A description of the new data group.

### Rename, Copy, and Paste

Existing Sequences and Steps can be renamed, copied, and pasted using the toolbar. Data Management Groups and Profiles cannot be copy or pasted.

Rename a Step or Sequence by selecting it from the Data Management Group then clicking  on the tool bar. Rename the Step or Sequence then click **OK**.

**NOTE:** A confirmation dialogue appears when changing the name of a Sequence.  
Click **OK** to rename or **Cancel** to revert changes.

Copy a Sequence or Step by right-clicking it then selecting **Copy** or clicking  on the toolbar.

Paste a Sequence or Step by right-clicking the area of the Data Management Group pane in which to paste then click **Paste** or clicking  on the toolbar. The steps of a sequence

**NOTE:** Pasting a Sequence or Step in the same hierarchy results in "Name\_copy".

### Sequences

A Sequence is an ordered series of one or more Data Management Steps which will execute in the order in which the Steps are organized. Click  to create a new Data Management Sequence.

### General

**Name**

The name given to the Data Management Sequence.

**Description (Optional)**

A description of the Data Management Sequence.

### Data Management Group

This indicates the Data Management Group under which the particular Sequence was created.

### Application Server (Optional)

This allows the specification of a particular application server in the event the Sequence has an abnormally long run-time or one application server is preferred over another.

## Task Activity

The Data Management sequence monitors the server's CPU and evaluates other queued tasks in order to make sure that they are processed in order as resources become available. If the server resource utilization is greater than the limit, the job status will be set to Queued and the task progress bar will stay on the queued task step until enough CPU resources are available to start the task. This can be monitored in Task Activity.

### Use Queueing

Set this to True to use queueing for this task in order to have better control of the application server's CPU utilization. This task will not start running until CPU utilization is below the specified value, or until all previously queued tasks have been completed. The default is set to True.

### Maximum % CPU Utilization To Run

Enter the maximum % CPU utilization allowed on the application server before this task is allowed to transition from queued to running. Enter a number between 1 and 100 or leave this setting blank to use the default. The default is 70%, however this can be modified in the Application Server Configuration.

**NOTE:** Do not set a value less than 10 or the task may never start.

### Maximum Queued Time Before Canceling (minutes)

Enter the number of minutes that this task is allowed to be queued before canceling it automatically. Enter a number, or leave this setting blank to use the default. The default is 20 minutes; however this can be modified in the Application Server Configuration.

When a task gets queued, the task progress dialog will stay on Task Queued. Open Task Activity to monitor the queued task and the current server CPU utilization.

**NOTE:** Batch processing queue overrides other Workflow batch queue settings.

The batch processing queue does not apply to batch script, only the Batch screen in the Workflow.

## Parameter Substitutions 1-8

### Parameter Name (Optional)

The name of a Parameter being used as a variable to be passed into the Sequence.

### **Value**

The value passed into the related Parameter variable.

## Sequence Steps Tab

This is where Steps are assigned and ordered to the Sequence.

## Steps

There are six built in Data Management Step types.

### Calculate

A Step can be created to use one of the built-in consolidation/calculation options available.

#### **Name**

The name of the Data Management Step.

#### **Description (Optional)**

The description of the Data Management Step.

#### **Data Management Group**

The Data Management Group to which the Step belongs.

#### **Step Type**

The type of Data Management Step chosen.

## Calculation Type

### **Calculate**

This executes a calculation when Parameter Components are changed. It runs calculations at the Entity level within the Local Member of the Consolidation Dimension without translating or consolidating.

### **Translate**

This executes Translate when Parameter Components are changed. It runs the Calculate step above at the Entity level and then translates data within the Translated Member of the Consolidation Dimension for each applicable relationship.

### **Consolidate**

This executes a Consolidate when Parameter Components are changed. This runs the Calculate and Translate steps and then completes the calculations required all the way up the Consolidation Dimension.

### Force Calculate, Translate, or Consolidate

The Force menu (e.g. Force Consolidation) items will run as if every cell included is marked as needing to be calculated, translated or consolidated.

### Calculate, Translate, or Consolidate with Logging

The Logging items (e.g. Force Translation with Logging) trigger additional detailed logging which can be viewed in the Task Activity area. Drill into a log to see the length of time and details about every calculation.

### Force Calculate, Translate, or Consolidate with Logging

This executes a Force Calculation, Translation, or Consolidation with Logging.

#### Cube

Specify the Cube where the consolidation/calculation will run.

#### Entity Filter

Specify the Entity, list of Entities or any combination of Entity hierarchies to be included in the consolidation/calculation.

#### Parent Filter

If alternate hierarchies are used, a Parent may be specified in order to be included in the consolidation/calculation.

#### Consolidated Filter

Specify the Consolidation Member or Members to be included in the consolidation/calculation.

#### Scenario Filter

Specify the Scenario Member or Members to be included in the consolidation/calculation.

#### Time Filter

Specify the Time Member or Members to be included in the consolidation/calculation.

## Execute Scenario Hybrid Source Data Copy

Copies base-level Cube data from a specified source Scenario (using the Hybrid Source Data settings) to a specified target Scenario. Data copy occurs when a standard calculation is run on the target Scenario and, by default, follows the standard OneStream calculation sequence.

## Clear Data

#### Name

The name of the Data Management Step.

#### Description (Optional)

A description of the Data Management Step.

## Application Tools

---

### Data Management Group

The Data Management Group to which the Step belongs.

### Step Type

The type of Data Management Step chosen.

### Use Detailed Logging

If set to True, a detailed log found in the Task Activity Log will provide additional detail about the process.

### Cube

Select the Cube where the consolidation/calculation will run.

### Entity Filter

Select the Entity, list of Entities or any combination of Entity hierarchies to be included in the consolidation/calculation.

### Scenario Filter

Select the Scenario Member or Members to be included in the consolidation/calculation.

### Time Filter

Select the Time Member or Members to be included in the consolidation/calculation.

### Clear Imported Data

This indicates whether the Import Member of the Origin Dimension should be included in the Clear Step. Settings are True or False.

### Clear Forms Data

This indicates whether the Forms Member of the Origin Dimension should be included in the Clear Step. Settings are True or False.

### Clear Adjustment Data and Delete Journals

This indicates whether the Adjustment Members of the Origin Dimension should be included in the Clear Step. Settings are True or False.

## Copy Data

### Name

The name of the Data Management Step.

### Description (Optional)

A description of the Data Management Step.

### Data Management Group

The Data Management Group to which the Step belongs.

### Step Type

The type of Data Management Step chosen.

### **Use Detailed Logging**

If set to True, a detailed log found in the Task Activity Log will provide additional detail about the process.

### **Source Cube**

Select the source Cube for the data copy.

### **Source Entity Filter**

Select the source Entity, list of Entities or any combination of Entity hierarchies to be included in the data copy.

### **Source Scenario**

Select the source Scenario Member or Members to be included in the data copy.

### **Source Time Filter**

Select the source Time Member or Members to be included in the data copy.

### **Source View**

Select the View Member to be included in the data copy. This selection allows users to copy YTD data into a Periodic Member and vice versa.

### **Destination Cube**

Select the destination Cube for the data copy.

### **Destination Entity Filter**

Select the destination Entity, list of Entities or any combination of Entity hierarchies to be included in the data copy.

### **Destination Scenario**

Select the destination Scenario Member or Members to be included in the data copy.

### **Destination Time Filter**

Select the destination Time Member or Members to be included in the data copy.

### **Destination View**

Select the destination View Member to be included in the data copy.

### **Copy Imported Data**

Choose True or False to copy O#Import data

### **Copy Forms Data**

Choose True or False to copy O#Forms data.

### **Copy Adjustment Data**

Choose True or False to copy O#Adjustments data.

**NOTE:** If the Source and Destination View fields are left blank, the data will copy to the same View Member. (e.g., YTD will copy to YTD or Periodic will copy to Periodic)

## Custom Calculate

The typical use of the Custom Calculate Step is for speed of calculations during data entry and flexibility. For instance, a user could make on-the-spot changes in a Form, run this Custom Calculate and quickly experience What-if analysis based on the limited amount of data on the Form. Instead of running a full Calculate or Consolidation on a Data Unit, the Custom Calculate Data Management Step can be used to run a calculation on a slice of data within one or many Data Units. The calculation could be executed from within Data Management, by clicking Save on a Form in Workflow (through a Forms Event Handler) or related to a button on a Dashboard being used to enter Budget data, to state a few examples.

This type of calculation does not create audit information for each data cell affected, therefore will run faster than using the Copy Data Data Management Step type.

The user executing must be a member of the Scenario's Manage Data Group or the Step will fail. This helps prevent unauthorized users from launching Steps like this, which could alter or clear data unintentionally.

### Name

The name of the Data Management Step.

### Description (Optional)

A description of the Data Management Step.

### Data Management Group

The Data Management Group to which the Step belongs.

### Use Detailed Logging

If set to True, a detailed log found in the Task Activity Log will provide additional detail about the process.

### Data Units

These are the Data Unit settings for what will be affected by calculation being applied. A single Cube and Scenario can be affected by this Step. A single or multiple Entities, Entity Parents, Consolidation members and Time Periods can be affected through Filter settings for Entity, Parent, Consolidation and Time in this section using the Member Filter Builder dialog. The related Business Rule below will run once for each of the number of Data Units specified here. These Data Unit members (e.g. Entity, Scenario, Time, etc.) can be referenced in the related Business Rule with functions such as API.Pov.Entity.Name or similar, but otherwise will not need to be mentioned within the context of the Business Rule. Upon running, this will reset Calculation Status of the affected Data Units to needing calculation.

### Point of View

These are the single member entries of dimensions not in the Data Unit to be affected by the calculation. Settings are for View, Account, Flow, Origin, IC and UD1-UD8. These values are provided in the Step as a convenience and to be referenced from within the Business Rule. For instance, the UD1 member listed here in the POV setting could be referenced from within the Business Rule as api.Pov.UD1.Name. This can make the Business Rule flexible since the same rule could be used against multiple Data Management Custom Calculate Steps, but run differently based on the POV setting.

#### Business Rule / Function Name

The name of the Finance-Type Business Rule and contained Function to run when this Step runs. This allows the user to specify settings such as Durable Storage within the rule. A simple example of a Business Rule which calculates data with Durable Storage is displayed below:

```
Select Case api.FunctionType
Case Is = FinanceFunctionType.CustomCalculate
If args.CustomCalculateArgs.FunctionName.XFEqualsIgnoreCase("myFunctionName") Then
    api.Data.Calculate("A#Profit = A#Sales - A#Costs", True)
End If
End Select
```

**NOTE:** When a Calculation or Consolidation runs on this same Data Unit after this Data Management Step is run, the data saved as calculated by this Step will be cleared unless it is saved with a Storage Type of Durable. ClearCalculatedData is first step in the standard Calculation Sequence that runs during a Calculation or Consolidation on Cube data. In this case, Durable data will be ignored during a calculate or even a Force Calculate or Consolidate unless a ClearCalculated function is used within the Business Rule or Member Formula to purposely clear the Durable data. However, if a calculation recalculates even data marked as Durable, it will then be replaced by the newly calculated data.

## Application Tools

---

It is suggested that if this calculation within a Custom Calculate Data Management Step is to be replicated within a separate Member Formula or Business Rule, both can refer to the same saved Finance Business Rule function by name. To refer to this logic, see Defining a Reference to a Shared Business Rule in "Implementing Security" on page 252.

See Examples of Key Functions in "About the Financial Model" on page 2 for examples of using API.Data.Calculate and API.Data.ClearCalculated to store and clear Durable data as desired.

### Parameters

Refer to any parameters inside the Business Rule by listing them here in name-value pairs in this fashion:

Name1=Frankfurt, Name2=[Houston Heights]

Custom Parameters can be used by using the correct syntax. This will result in a prompt to the user at run time.

Name3=|!myParam!|

## Execute Business Rule

### Name

The name of the Data Management Step.

### Description (Optional)

A description of the Data Management Step.

### Data Management Group

The Data Management Group to which the Step belongs.

### Step Type

The type of Data Management Step chosen.

### Use Detailed Logging

If set to True, a detailed log found in the Task Activity Log will provide additional detail about the process.

### Business Rule

Select from the available application Business Rules to run custom scripts or procedures.

### Parameters (Optional)

This field is provided to pass Parameters or variables into the selected Business Rule.

## Export Data

### Name

The name of the Data Management Step.

**Description (Optional)**

A description of the Data Management Step.

**Data Management Group**

The Data Management Group to which the Step belongs.

**Step Type**

The type of Data Management Step chosen.

**Use Detailed Logging**

If set to True, a detailed log found in the Task Activity Log will provide additional detail about the process.

**File Share File Name**

The name given to the file being exported to the application file share.

**Include Cube in File Name**

If set to True, the Cube name will be included in the file name when exporting.

**Include Entity in File Name**

If set to True, the Entity name will be included in the file name when exporting.

**Include Parent in File Name**

If set to True, the Parent name will be included in the file name when exporting.

**Include Cons in File Name**

If set to True, the Consolidation Member name will be included in the file name when exporting.

**Include Scenario in File Name**

If set to True, the Scenario name will be included in the file name when exporting.

**Include Time in File Name**

If set to True, the Time name will be included in the file name when exporting.

**Overwrite Existing Files**

Set this to True to overwrite a previously exported file when exporting.

**Include Zeros**

If set to True, zero amount records will be included when exporting.

**Include Member Descriptions**

If set to True, Member descriptions will be included when exporting.

**Include Cell Annotations**

If set to True, cell annotations will be included when exporting.

**Include Input Data**

If set to True, input type data will be included when exporting.

## Application Tools

---

### **Include Calculated Data**

If set to True, calculated data will be included when exporting.

### **Cube**

Specify the Cube where consolidation/calculation will run.

### **Entity Filter**

Specify the Entity, list of Entities or any combination of Entity hierarchies to be included in the consolidation/calculation.

### **Parent Filter**

If alternate hierarchies are used, a Parent may be specified in order to be included in the consolidation/calculation.

### **Consolidated Filter**

Specify the Consolidation Member or Members to be included in the consolidation/calculation.

### **Scenario Filter**

Specify the Scenario Member or Members to be included in the consolidation/calculation.

### **Time Filter**

Specify the Time Member or Members to be included in the consolidation/calculation.

### **Combinations of Data Filters (use #All for all stored base-level data)**

Use standard Member Filter functionality to select specific data required for all Dimensions listed below:

Account Filter (e.g., "A#IS.Descendants, A#All")

Flow Filter

Origin Filter

IC Filter

UD1-8 Filter

### **Specific Data Filters**

Use standard Member Filter functionality in order to have fine grain control over the data exported. Use this filter to choose more specific intersections of data.

## **Export File**

Use the Export File Data Management Step to export an Extensible Document or Report Book to OneStream's File Share.

### **Name**

The name of the Data Management Step.

### **Description (Optional)**

A description of the Data Management Step.

**Data Management Group**

The Data Management Group to which the Step belongs.

**Step Type**

The type of Data Management Step chosen.

**Use Detailed Logging**

If set to True, a detailed log found in the Task Activity Log will provide additional detail about the process.

**File Share Folder**

Enter the name of the File Share Folder where the desired file is saved.

**File Name Suffix**

Enter a suffix to include in the name of the file.

**Overwrite Existing Files**

Set to True to overwrite a previously exported file when exporting.

## File Source Type

**URL**

Display a file from an internal or external web page.

**Application/System Dashboard File**

Display a file stored in a Dashboard Maintenance Unit File section.

**Application/System Database File**

Display a file stored in the Application or System Database Share.

**File Share File**

Display a file from the File Share.

**URL or Full File Name**

The URL or name of the file being used. Enter the full URL name, or click the ellipsis and browse to the desired file.

**Process Extensible Document**

If set to True, the Data Management Step will run and process the attached Extensible Document file, if set to False the unprocessed file will display which is mainly used for testing purposes.

**NOTE:** An Extensible Document is a Text, Word, PowerPoint, or Excel file that uses Parameters in its content. The file name must contain .xfDoc before the extension.

Example: StatusReport.xfDoc.docx

See Extensible Document Framework in "Presenting Data With Extensible Documents" on page 222 for more details on this feature.

### **Read Parameter Names from Extensible Doc**

If set to True, the Parameters used in the Extensible Document will prompt the user when the Data Management Sequence is run, if this is set to False, the Parameters can be specified ahead of time in the Parameters property below.

### **Parameters**

Enter a comma separated list of name value pairs.

Example: ParameterName1=ValueName1, ParameterName2=[Value Name2]

## Export Report

### **Name**

The name of the Data Management Step.

### **Description (Optional)**

A description of the Data Management Step.

### **Data Management Group**

The Data Management Group to which the Step belongs.

### **Step Type**

The type of Data Management Step chosen.

### **Use Detailed Logging**

If set to True, a detailed log found in the Task Activity Log will provide additional detail about the process.

### **File Share File Name**

The name given to file being exported to the application file share.

### **Include Object Name in File**

If set to True, the Dashboard Object will be included in the file name when exporting.

### **Overwrite Existing Files**

Set to True to overwrite a previously exported file when exporting.

## Report File Type

### **PDFs in Zip File**

This will export all contained PDF's individually in a ZIP file.

### **Combined PDF File**

This will combine and then export all report PDF's into one PDF file.

### Object Type

Indicate whether a Dashboard or Dashboard Profile object is to be exported.

### Object Name

The name of the Dashboard or Dashboard Profile being used.

### Object Parameters (Optional)

This field is provided to pass Parameters or variables used when running the report normally.

## Reset Scenario

Similar to Clear Data Step, except it clears additional related application data, yet does not create audit information for each data cell affected, therefore will run faster. It clears data within a range (including parent Entity data), audit information, Workflow Status and Calculation Status as if it never existed. This is intended for Administrator use. User executing must be a member of the Scenario's Manage Data Group or the Step will fail. This helps prevent unauthorized users from launching Steps like this, which could alter or clear data unintentionally. It is recommended to only change Manage Data Group from Nobody to a select exclusive User Group before running and then change back to Nobody afterwards. Also, ensure that your application database is backed up before performing a Reset Scenario. Note that Reset Data will clear even data marked as Durable.

### Name

The name of the Data Management Step.

### Description (Optional)

A description of the Data Management Step.

### Data Management Group

The Data Management Group to which the Step belongs.

### Use Detailed Logging

If set to True, a detailed log found in the Task Activity Log will provide additional detail about the process.

### Scenario

Choose one Scenario to be reset.

### Reset All Time Periods

Default is True, which will remove data and reset Workflow Status for all time periods. If False, the next three properties will be enabled.

### Start Year / End Year

The first and last years' worth of time to clear from the Application and Workflow Status.

### Start Time Period in First Year

Optional. Time Period referenced in First Year to begin clearing data.

# Run a Sequence or Step

While highlighting a Sequence or Step, click this button: .

When running a Sequence in Data Management, go to **System Tab > Tools > File Explorer** to find any file export.

Go to **File Share > Applications > Choose an application > DataManagement > Export > Username > Choose the latest folder**.

# Spreadsheet

The OneStream Windows App Spreadsheet feature enables users to incorporate spreadsheet functionality experienced by users of the OneStream Excel Add-in without needing to have Excel loaded on their computer. It can be used for ad hoc querying/reporting, analysis, data entry, and formatted reports. The Spreadsheet feature enables users to stay within the OneStream Windows App while being able to use functionality similar to what they know and love in Excel. Similar to the Excel Add-in, the Spreadsheet feature leverages OneStream's re-usable Cube Views for fast and easy analysis. See "Navigating the Excel Add-In" on page 1034 for more details on this feature.

- Here are some other functional highlights and benefits:
- Change point-of-view, interact with Forms, assign Cube Views, drill through to source data and update Workflow status
- Utilize familiar retrieve functions for ad-hoc reporting and analysis
- Eliminate risk of errors and duplication of efforts with standardized and centralized spreadsheet controls
- Safely edit, update and analyze data as spreadsheet forms respect Application and Workflow and security
- Eliminate spreadsheet maintenance when metadata changes because Cube Views are read dynamically
- Sheet-based calculations remain intact even when rows or columns are added
- Include multiple Cube View results in the same sheet

## Application Tools

---

The OneStream Windows App Spreadsheet feature can perform most of the tasks accomplished by the OneStream Excel Add-in, with some limitations. These limitations include, but are not limited to:

- Macros
- Solver
- Document properties
- No option to insert copied/cut cells in spreadsheet
- Preview format does not update for format types Effects (Superscript, Subscript, Strikethrough)
- Ability to manage conditional formatting – color scales are reversed
- Shift+End
- Spreadsheet border button does not store last selection
- Text Direction option from Alignment formatting
- Underline formatting does not appear same as in Excel
- Not all Chart options are available in Spreadsheet
- Ability to change chart colors in spreadsheet
- References to other sheets in Spreadsheet is a different process
- Ctrl+PageUp / Ctrl+PageDn correctly changes the tab but does not reset tab focus in spreadsheet
- CTRL+N creates a new spreadsheet, however if an existing spreadsheet was opened with unsaved changes, it closes the existing spreadsheet without saving edits made since last save

Some functionality is currently not supported in the Spreadsheet Tool. Functionality that is known not to be supported in the Spreadsheet feature at this time is listed below:

- Fill number formatting
- The ability to format part of a cell with different formatting

## Application Tools

---

- Justify alignment option
- Managing Rules under spreadsheet conditional formatting
- Spreadsheet alignment cell format "Center Across Selection"
- Undo/Redo

The following Excel Column Charts are incompatible and unavailable with the Spreadsheet feature:

- Stacked 3-D
- 100% Stacked 3-D
- Stacked Cylinder
- 100% Stacked Cylinder
- Stacked Cone
- 100% Stacked Cone
- Stacked Pyramid
- 100% Stacked Pyramid

The following Excel Line chart is incompatible and unavailable with the Spreadsheet feature:

- 3-D Line

The following Excel Bar charts are incompatible and unavailable with the Spreadsheet feature:

- Stacked 3-D Bar
- 100% Stacked 3-D Bar
- Stacked Horizontal Cylinder
- 100% Stacked Horizontal Cylinder
- Stacked Horizontal Cone
- 100% Stacked Horizontal Cone

- Stacked Horizontal Pyramid
- 100% Stacked Horizontal Pyramid

## Microsoft Excel 2016 Chart Types

There are seven new chart types added to the Spreadsheet component in the Windows application. These charts were first introduced to Microsoft Excel 2016, and are now available in our spreadsheet component as an alternative to using Excel.

The OneStream Windows application spreadsheet feature does not require that Excel be loaded on your computer. It can be used for querying and reporting, analysis, data entry, and formatted reports. The Spreadsheet feature allows you to remain in the Windows application using Excel-like functionality.

This is an enhancement to the existing spreadsheet component in the Windows application. It is available to administrators or general users with security rights to access it.

## Chart Types

Microsoft Excel 2016 introduced several chart types that help you visualize financial, statistical, and hierarchical data. They include:

- Box and Whisker
- Waterfall
- Histogram
- Pareto
- Funnel
- Sunburst
- Treemap

You can add charts to a OneStream spreadsheet worksheet in the same manner as any other chart type. Chart styles allow you to quickly change chart appearance. Styles change the background fill, specify the color of the data series, and apply different shape effects and outlines to the chart.

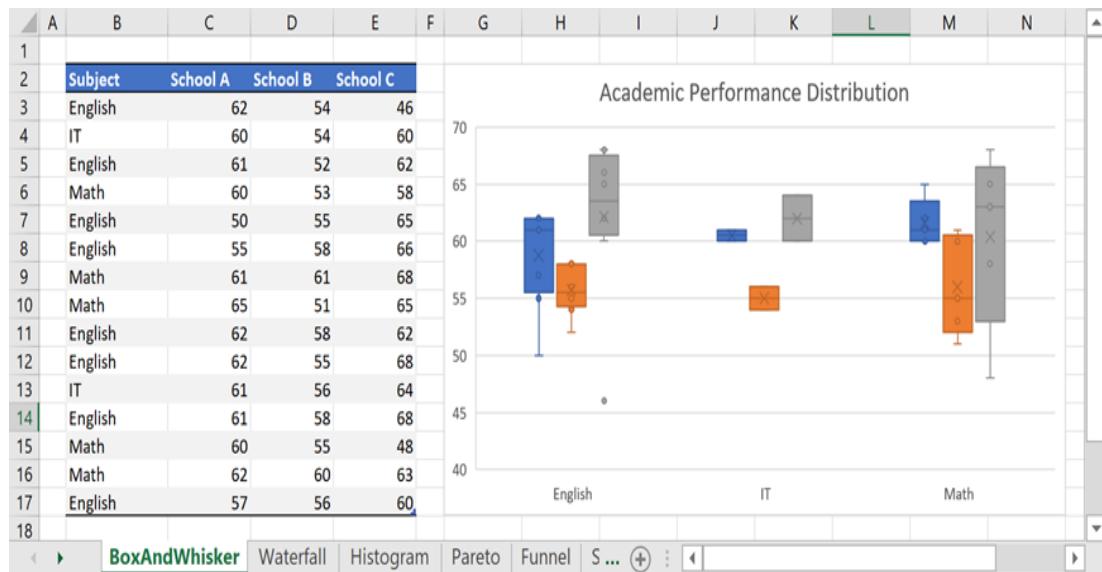
To apply Excel 2016 chart types to a predefined chart style, right-click on the chart and select one of the available styles.

## Application Tools

# Box and Whisker

A box and whisker chart shows distribution of data in quartiles, highlighting the mean and outliers. Boxes may have lines extending vertically called “whiskers.” These lines indicate variability outside the upper and lower quartiles, and any point outside those lines or whiskers is considered an outlier.

Box and whisker charts are commonly used in statistical analysis. For example, you could use a box and whisker chart to compare medical trial results or teachers' test scores.



## Waterfall

A waterfall chart shows a running total as values are added or subtracted. It is useful for understanding how an initial value (for example, net income) is affected by a series of positive and negative values. The columns are color coded so you can quickly tell positive from negative numbers. The initial and the final value columns often start on the horizontal axis, while the intermediate values are floating columns.

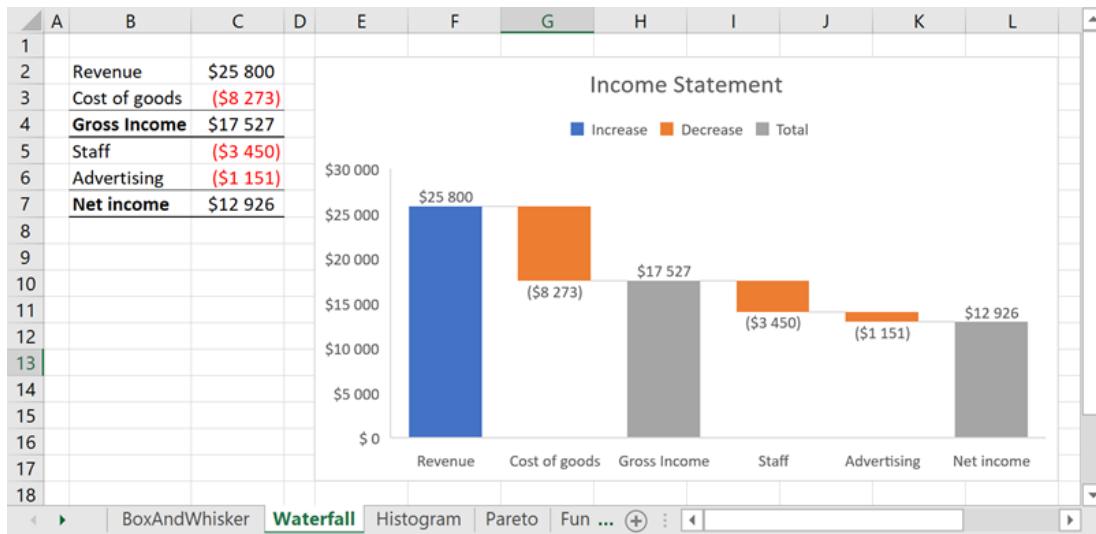
If your data includes values that are considered Subtotals or Totals, such as Net Income, you can set those values so they start on the horizontal axis at zero and don't float. To do this:

1. Right-click on the chart and select **Set as Total**.
  2. In the dialog box, select the checkbox for value(s) in the dataset to be set as Total.

## Application Tools

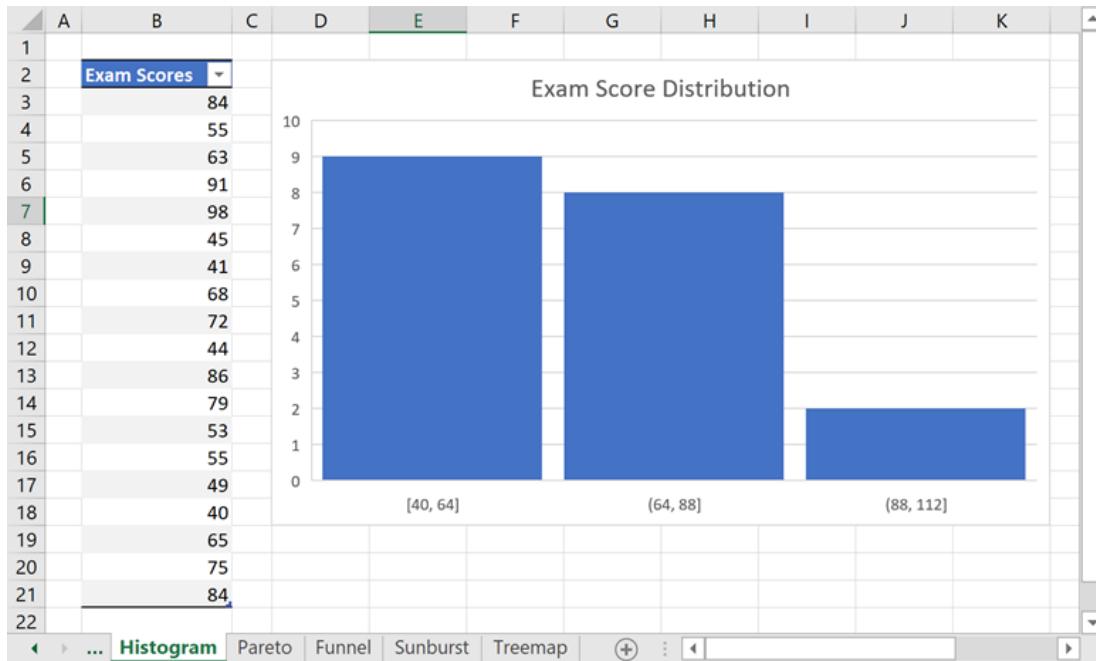
---

- To make the column float again, clear the **Set as total** checkbox.



## Histogram

Data plotted in a histogram chart shows the frequencies within a distribution. Each column of the chart is called a frequency bin.

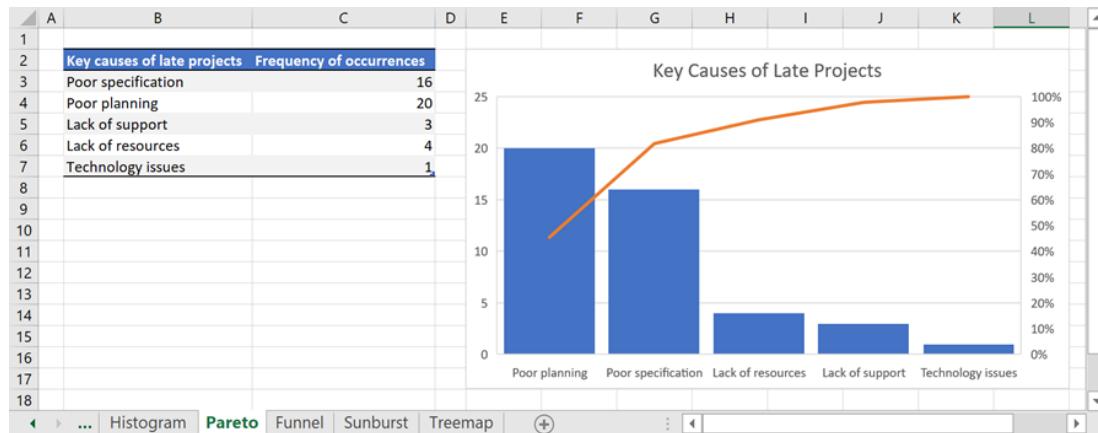


## Application Tools

---

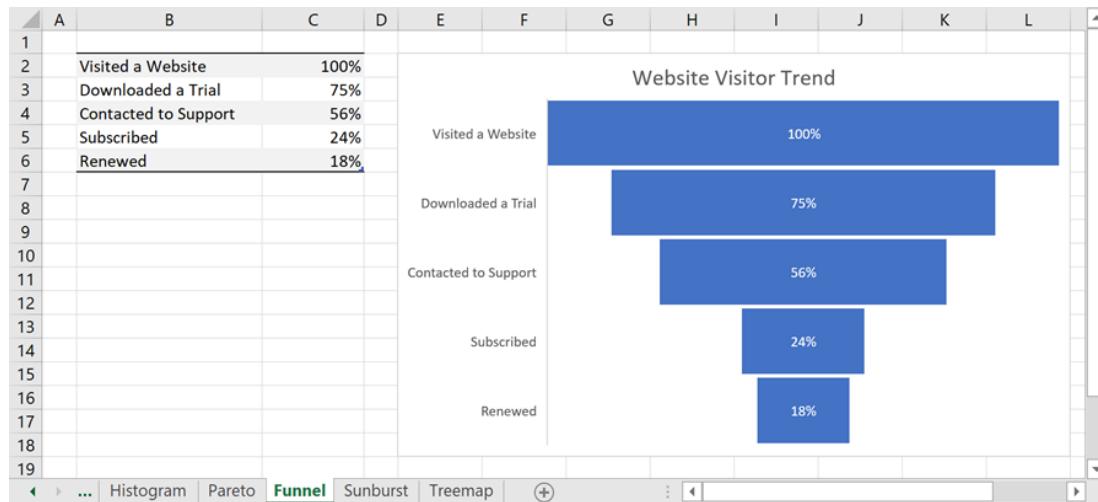
### Pareto

A pareto chart is a sorted histogram that contains columns sorted in descending order along with a line representing the cumulative total percentage.



### Funnel

Funnel charts show values across multiple stages in a process. Typically, the values decrease gradually, causing the bars to resemble a funnel. For example, you could use a funnel chart to show the number of sales prospects at each stage in a sales pipeline.

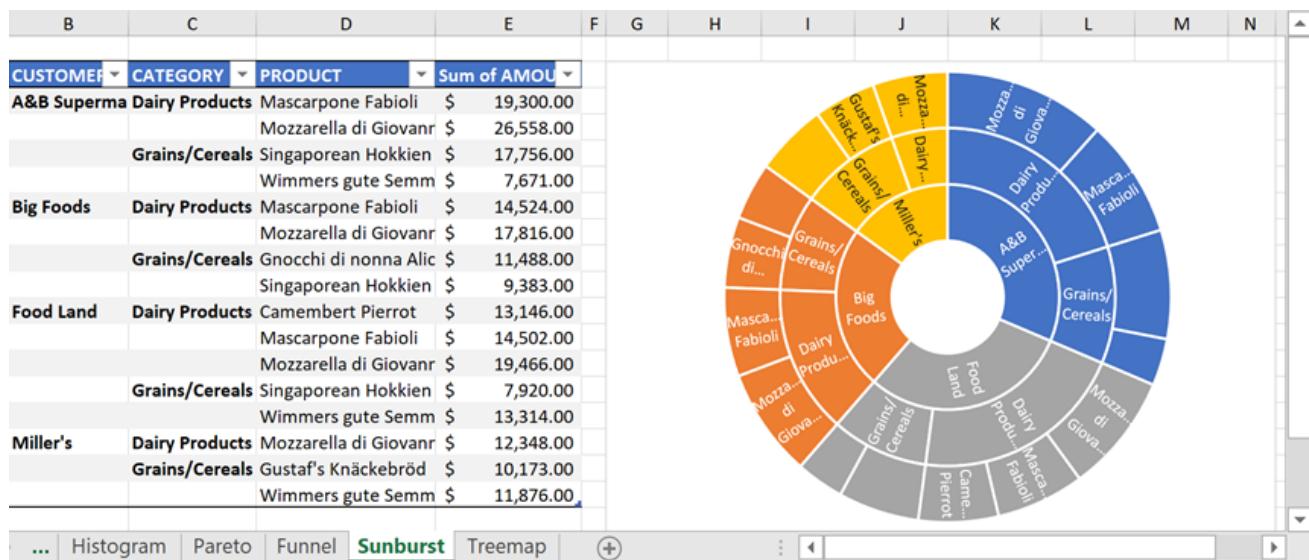


## Application Tools

---

### Sunburst

The sunburst chart displays hierarchical data and can be plotted when empty (blank) cells exist within the hierarchical structure. Each level of the hierarchy is represented by one ring or circle with the innermost circle as the top of the hierarchy. A sunburst chart without hierarchical data (one level of categories), looks similar to a doughnut chart. However, a sunburst chart with multiple levels of categories shows how the outer rings relate to the inner rings. The sunburst chart is most effective at showing how one ring is broken into its contributing pieces.

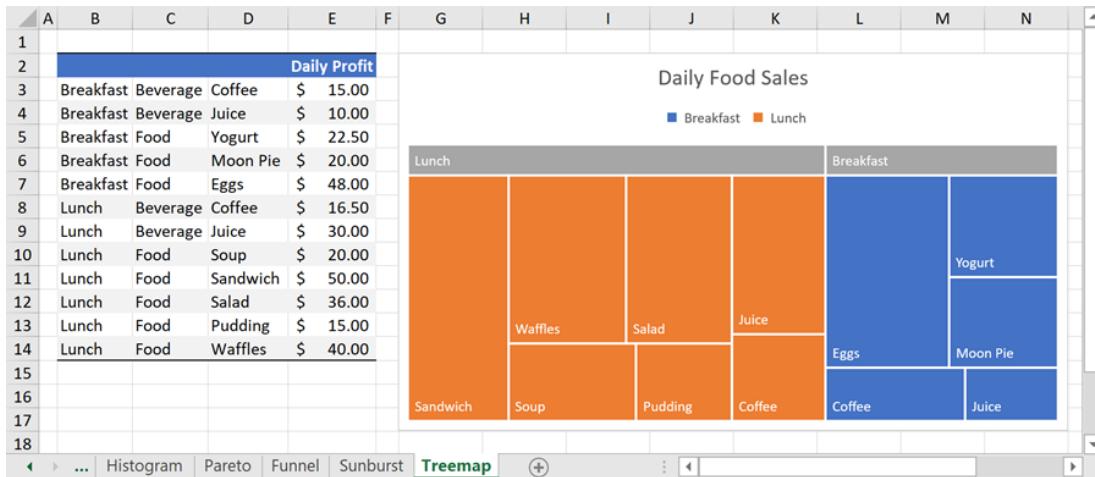


### Treemap

The treemap chart provides a hierarchical view of your data and an easy way to compare different levels of categorization. The treemap chart displays categories by color and proximity and can show data which might be difficult to display with other chart types. The treemap chart is plotted when empty (blank) cells exist within the hierarchical structure. Treemap charts are good for comparing proportions within the hierarchy.

## Application Tools

---



## Conclusion

These chart types help you visualize financial, statistical, and hierarchical data inside the Windows application spreadsheet component. They are ideal for those who prefer working inside the Windows application rather than Microsoft Excel.

## Retrieve Functions

Retrieving and changing data can be done by using functions. To see the functions and their Parameters, open the Spreadsheet feature and select the Formulas tab. Within the Spreadsheet feature, select Insert Function and select User Defined where it says to Select a category. See Retrieve Functions in "Navigating the Excel Add-In" on page 1034 for more details on this feature.

## File, Open and File, Save As Functionality

Additional options for opening and saving files exist within the Spreadsheet feature.



### Open

File type to open

### Local Folder

Select a file to be opened from a location on the local computer/network.

## Application Tools

---

### File System

Select a file to be opened from a location within the File Explorer.

**NOTE:** Displayed files can also be opened from here, or from OnePlace Documents by right clicking and selecting one of the three options – “Open in Spreadsheet Page”; “Open” (opens file directly in Excel if the application is found on the local computer) or “Open With...” (user specifies program).

## Application Dashboard File

Select a file to be opened from an Application Dashboard.

### System Dashboard file

Select a file to be opened from a System Dashboard.

### Save As

File type to be saved as.

### Local Folder

Select location on local computer/network to save a file to.

### File System

Select location within File Explorer to save a file to.

### Application Dashboard

Select Application Dashboard location to save a file to.

### System Dashboard file

Select System Dashboard location to save a file to.

## Reduce Cell Text Size

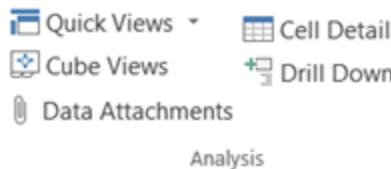
You can automatically reduce text size to fit inside a spreadsheet cell. This eliminates the need to resize the column width and prevents text from being cut off when it exceeds the cell width.

1. Inside the spreadsheet, select a single cell, multiple cells, or all cells.
2. Right-click on the selected cells and select **Format Cells**.
3. Click the **Alignment** tab.
4. Select the **Shrink to fit** checkbox.
5. Click **OK**.

# OneStream Ribbon

When using the Spreadsheet feature, there will be a OneStream menu item and a ribbon. See OneStream Ribbon "Navigating the Excel Add-In" on page 1034 for an explanation of the items within Spreadsheet that function the same as those in the Excel Add-In. Items documented below function differently than they do in the Excel Add-In, or do not exist in the Spreadsheet Tool.

## Analysis



The File Explorer option does not exist in the Spreadsheet feature because files are automatically stored. They do not exist within Excel.

## Administration



The Display Context Pane option does not exist in the Spreadsheet feature because it is not needed. The OneStream task pane automatically appears when using the Spreadsheet feature.

Under Preferences, the General options are not needed in the Spreadsheet for Windows Only tool. These options manage how Microsoft Sign In is handled, how Macros are handled, and how data is refreshed in an open workbook. The Spreadsheet feature does not utilize this functionality.

# Task Scheduler

Task Scheduler provides the ability to schedule data management sequences that execute a data management step within the application. If the sequence doesn't have a step, the job will fail.

## Application Tools

---

The screenshot shows a software application window titled "Task Scheduler". At the top, there are three view options: "Grid View" (selected), "Calendar View", and "Show Tasks for all Users". Below the header, a message says "Drag a column header and drop it here to group by that column". The main area is a grid table with the following columns: User Name, Name, Description, Sequence, Schedule, Next Start Time, Last Start Time, Expire Date/Time, State, Count, Edit, and Delete. The data in the grid includes various scheduled tasks such as "Admin Export TB Weekly", "Mgr Export Stage Archiving", and "Weekly TB".

User Name	Name	Description	Sequence	Schedule	Next Start Time	Last Start Time	Expire Date/Time	State	Count	Edit	Delete
Admin	Admin Export TB Weekly	Export Trial Balance	Export Trial Balance	One Time	NA	NA	8/24/2020 9:37:27 AM	Enabled	0		
Admin	Mgr Export Stage Archiving	Export Stage Archives	Export Stage Archives	Weekly	11/21/2020 8:20:37 AM	9/29/2020 8:53:53 AM	12/31/2200 12:00:00 AM	Enabled	4		
Admin	Weekly TB	Weekly Trial Balance	Export Trial Balance	Weekly	11/21/2020 3:50:35 PM	11/20/2020 3:50:44 PM	12/31/2200 12:00:00 AM	Enabled	6		
Admin	Yearly Full Consolidation	Yearly Full Consolidation	Full Consolidation	Monthly	1/6/2021 6:33:14 PM	NA	8/21/2021 7:26:19 PM	Enabled	0		
TSUser3	TSUser3 Export Budget Data	Export Actual (Prior Year)	Export Budget Data	Weekly	11/21/2020 2:19:00 PM	11/20/2020 2:19:05 PM	12/31/2200 12:00:00 AM	Enabled	6		
TSUser3	TSUser3 Export Stage Archiving	Export Stage Archives	Export Stage Archives	Daily	11/21/2020 2:17:44 PM	11/20/2020 2:18:05 PM	12/31/2200 12:00:00 AM	Enabled	1		
TSUser3	TSUser3 Export Trial Balance	Export Trial Balance	Export Trial Balance	Minutes	11/20/2020 4:49:41 PM	11/20/2020 4:39:47 PM	12/31/2200 12:00:00 AM	Enabled	582		

## Grid View

Go to Application > Tools > Task Scheduler.

The default view is the calendar view. You can change the view to a grid view which is initially blank until you've scheduled sequences.

The default is to Show Tasks for all Users, but you can click the option to turn it off. Once you've created tasks, they will show in the grid.

The screenshot shows a software application window titled "Task Scheduler". At the top, there are three view options: "Grid View" (selected), "Calendar View", and "Show Tasks for all Users". Below the header, a message says "Drag a column header and drop it here to group by that column". The main area is a grid table with the following columns: User Name, Name, Description, Sequence, Schedule, Next Start Date/Time, Last Start Date/Time, Expire Date/Time, State, Count, Edit, and Delete. The data in the grid includes various scheduled tasks such as "Admin Export TB Weekly", "EveryHourFrom8pmtoMidnight", and "Export Daily".

User Name	Name	Description	Sequence	Schedule	Next Start Date/Time	Last Start Date/Time	Expire Date/Time	State	Count	Edit	Delete
Admin	Admin Export TB Weekly	Export Trial Balance Weekly Fri_Sun	Export Trial Balance	Weekly	8/27/2021 9:47:27 AM	NA	8/24/2020 9:37:27 AM	Enabled	0		
Admin	EveryHourFrom8pmtoMidnight		Export Trial Balance	Minutes	8/25/2021 12:00:00 PM	8/24/2021 8:00:01 PM	8/24/2021 8:05:00 PM	Enabled	1		
Admin	Export Daily	Export Trial Balance Daily	Export Trial Balance	Daily	8/26/2021 12:05:33 AM	11/26/2020 11:05:37 PM	11/30/2020 11:04:41 PM	Enabled	15		
Admin	Export Expire Date	Export Stage Archive w_Expire Date	Export Stage Archives	Minutes	8/25/2021 11:58:56 AM	11/12/2020 11:54:05 PM	11/12/2020 11:57:54 PM	Enabled	12		
Admin	Export Monthly	Export TB Monthly	Export Trial Balance	Monthly	11/12/2021 11:01:31 PM	11/14/2020 11:01:39 PM	11/12/2021 11:00:20 PM	Enabled	3		
Admin	Export TB One Time	Test SetScenario	Export Trial Balance	Weekly	8/25/2021 6:04:26 PM	8/23/2021 6:04:41 PM	12/31/2200 12:00:00 AM	Enabled	30		
Admin	Export Trial Balance	Demo for Export Trial Balance	Export Trial Balance	One Time	NA	11/18/2020 5:00:15 PM	11/18/2021 11:51:27 AM	Enabled	1		
Admin	Export Trial Balance Budget	Export Trial Balance Monthly	Export Trial Balance	Monthly	9/2/2021 12:48:43 PM	6/28/2021 12:49:02 PM	12/31/2021 11:48:43 AM	Enabled	8		
Admin	Export Trial Balance BW	Export Trial Balance Weekly	Export Trial Balance	Weekly	8/27/2021 9:00:00 AM	3/15/2021 9:00:28 AM	3/18/2021 8:00:00 PM	Enabled	6		
Admin	Export Trial Balance Every Monday	Export Trial Balance Monthly 1st...	Export Trial Balance	Monthly	9/2/2021 6:00:00 PM	NA	2/27/2021 5:00:00 PM	Enabled	0		
Admin	Future Export	Export TB in Jan 2021	Export Trial Balance	One Time	NA	1/12/2021 11:06:05 PM	11/12/2021 11:05:48 PM	Enabled	1		
Admin	MyHourlyTask	RunEveryHourBetween10am3PM	Export Trial Balance	Minutes	8/25/2021 12:00:00 PM	8/24/2021 11:00:17 PM	12/31/2200 12:00:00 AM	Enabled	35		
Admin	TestDailyByHour		Export Trial Balance	Minutes	8/25/2021 12:40:16 PM	6/29/2021 4:40:40 PM	6/29/2021 5:40:18 PM	Enabled	8		

Whatever view you are currently on is the view you will come back to the next time you go to the page.

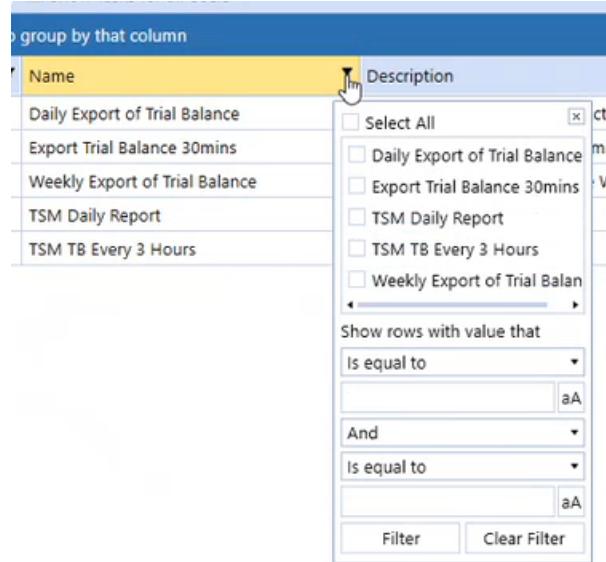
## Task Scheduler Details

Field	Definition
User Name	The user that created the task.
Name	Name of the task.
Description	Description of the task.
Sequence	Data management sequence that is run by the task scheduler.
Schedule	The time frame implemented for the specific task.
Next Start Time	The next time the task is scheduled to run.
Last Start Time	The last time the task ran.
Expire Date/Time	The time and date the task will expire and no longer run.
State	Enabled or disabled.
Count	The amount of time the task has ran.
Edit	Click to edit.
Delete	Click to delete.
If enabled by an administrator:	
Invalidate Date/Time	The time and date the task will be suspended and not run until validated.
Validate Task	Validate task to keep active

## Application Tools

---

You can filter on any of the fields in the grid that have the filter icon and you can filter on multiple selections.



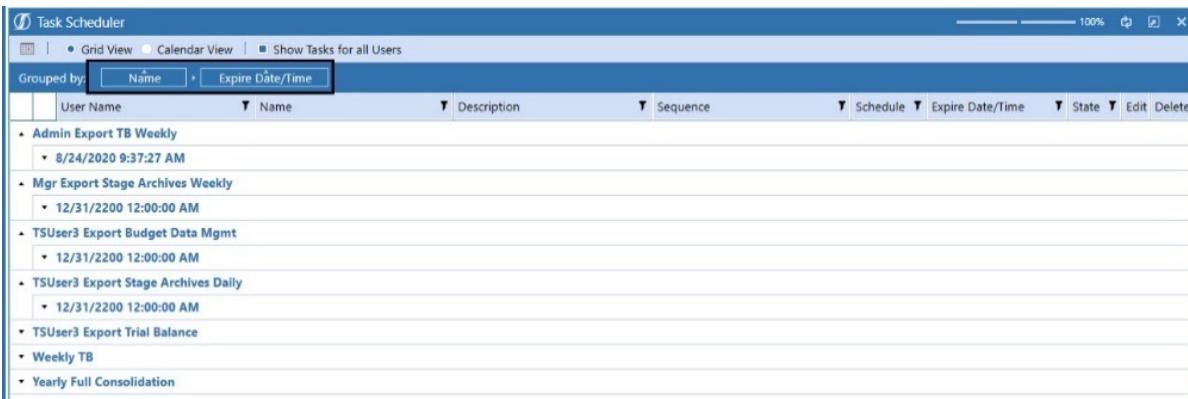
You can group by column name by dragging and dropping the group into the header bar.



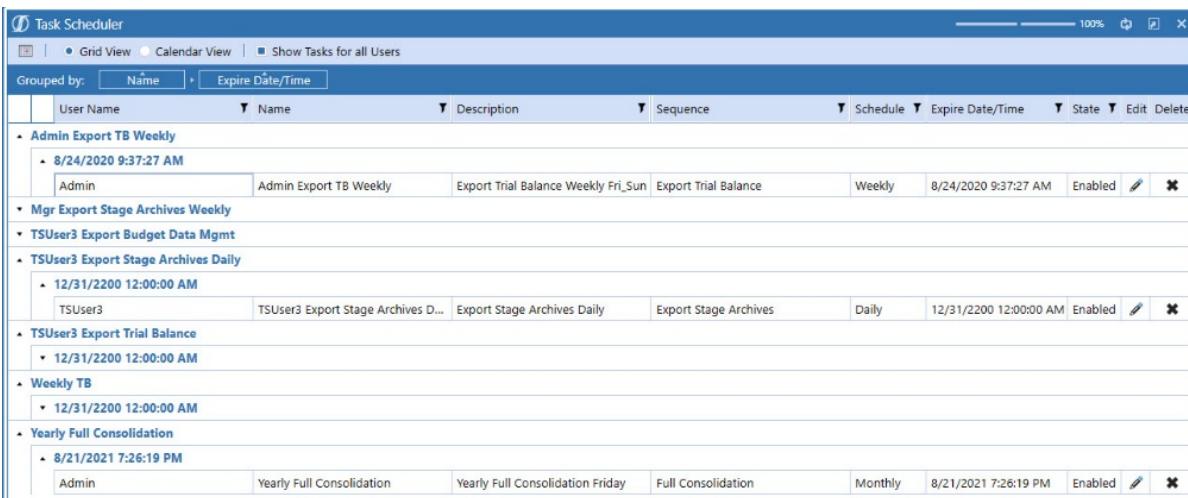
You can group by more than one column.

## Application Tools

---



You can expand or collapse by the task.

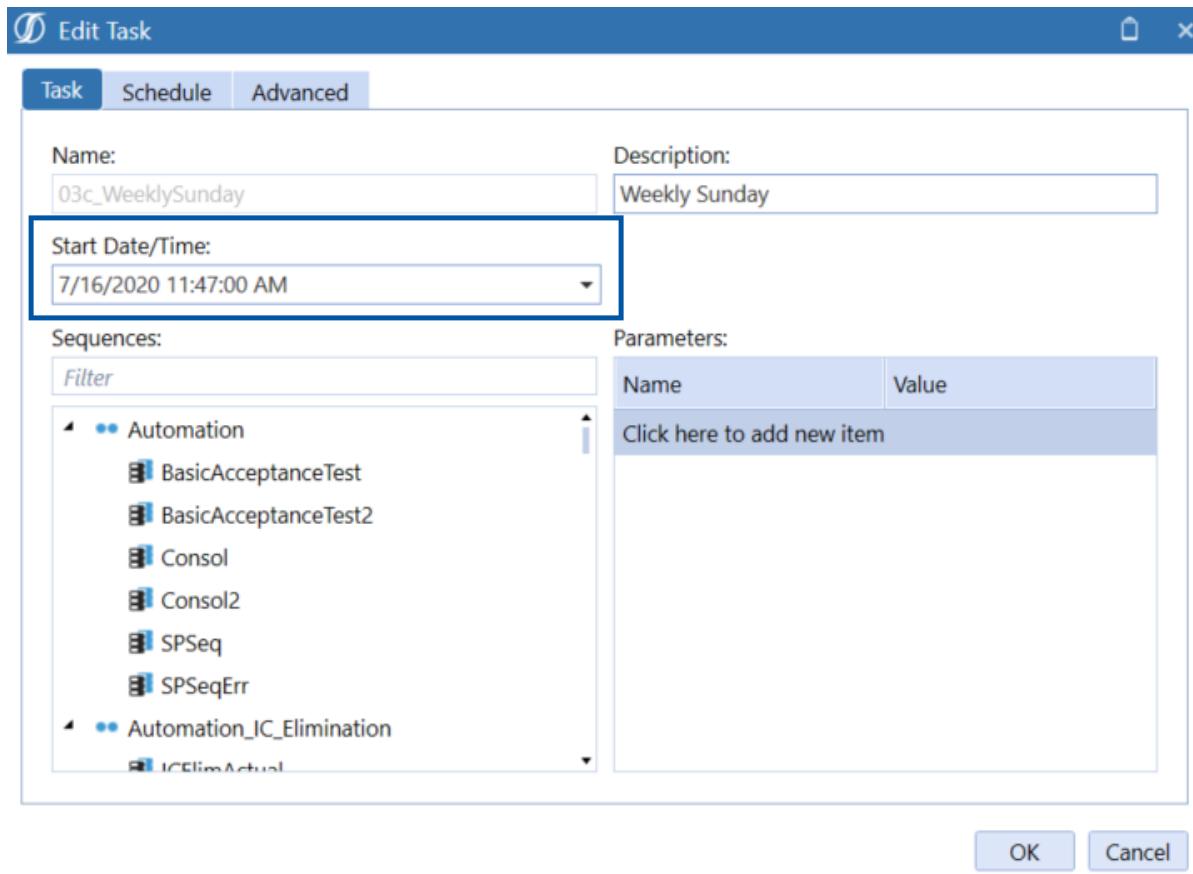


Double-click on a task or click Edit to open Edit Task. The only option that you can not change is the **Name** in the Task tab.

## Application Tools

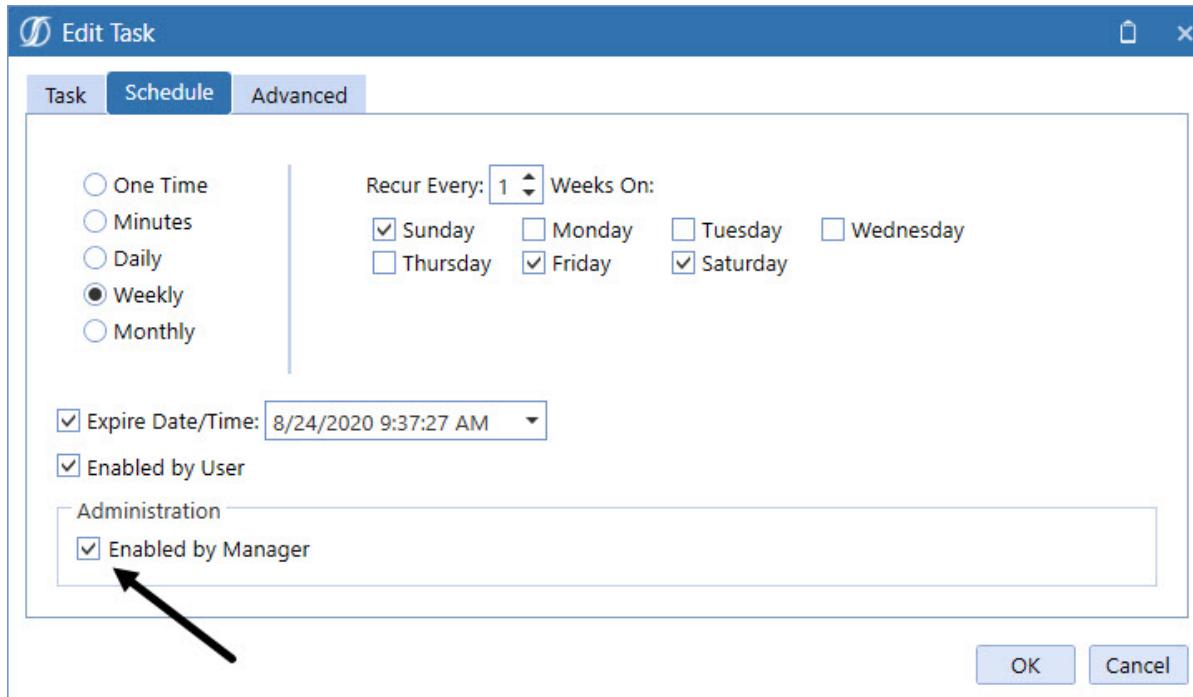
---

The **Start Date/Time** are your local time.



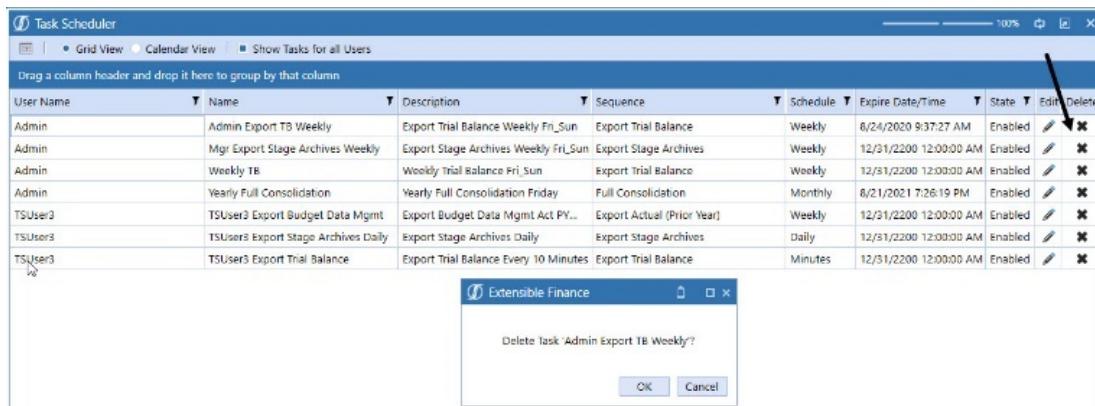
## Application Tools

If you are not the Administrator, you will not have the rights to change the **Enabled by Manager** check box.



You can do any of the following Delete options:

- Click to **Delete**.



## Application Tools

- Select the task and right-click to delete.

Drag a column header and drop it here to group by that column		
User Name	Name	Description
Admin	Daily Export of Trial Balance	Daily Export of Trial Balance Actuals
Admin	Export Trial Balance 30mins	Exports Trial Balance every 30mins
Admin	Weekly Export of Trial Balance	Weekly Export of Trial Balance Wed Th Fri
TSM	TSM Daily Report	Trial Balance Daily Report
TSM	TSM TB Every 3 Hours	Export TB Every 3 Hours

## Calendar View

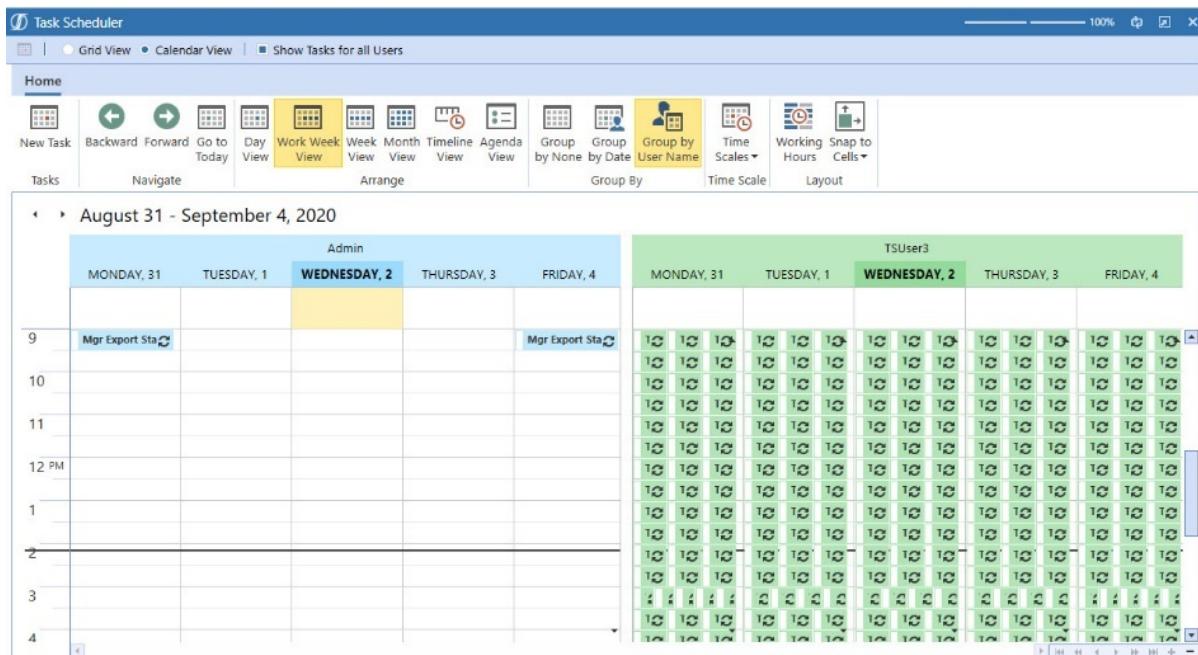
Provides you a view of the task in the calendar.

The screenshot shows the 'Task Scheduler' application window. At the top, there's a toolbar with buttons for 'Grid View' (which is selected), 'Day View', 'Work Week View', 'Month View', 'Timeline View', 'Agenda View', and several grouping and scaling options. A red arrow points to the 'Calendar View' button. Below the toolbar, the date range is set to 'August 31 - September 4, 2020'. The main area is a grid where each row represents a day from Monday to Friday. Each cell in the grid contains a small icon representing a task, and the background color of the cells varies by user, showing a repeating pattern of colors (blue, green, yellow, red) across the days.

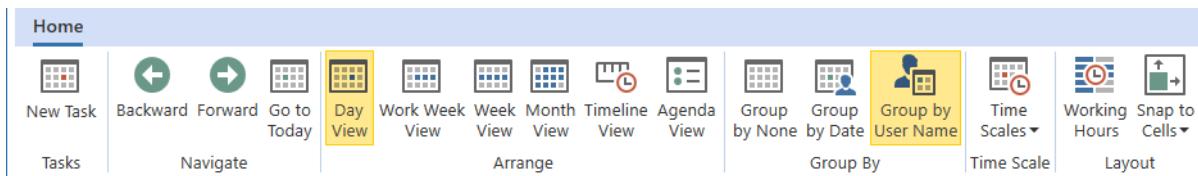
If there is more than one user showing, the tasks are color coded when Show Tasks for All Users is selected.

You can view your own tasks and all user's tasks if checked.

## Application Tools

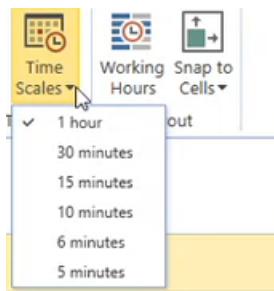


You can go backward and forward within the different views and you can view by today, work week, work month, timeline, and agenda.



You can group by the user name, date, or no group.

When activated, you can choose to view the calendar by time scales or working hours.



## Application Tools

---

When you hover over a task on the calendar you will see the information specific to the task including User Name, Task Name, Sequence Name, Schedule, State, Expire Date/Time, if enabled, the Invalidate Date/Time.

The screenshot shows the 'Work Week View' selected in the navigation bar. The calendar displays tasks for the week of August 31 to September 4, 2020. A tooltip is overlaid on a task on Monday, August 31, showing the following details:

- User Name: Admin
- Task Name: Mgr Export Stage Archives Weekly
- Sequence Name: Export Stage Archives
- Schedule: Weekly
- State: Enabled
- Expire Date/Time: 12/31/2200 12:00:00 AM

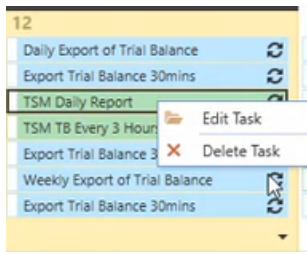
You can also create a new task, edit an existing task, or delete a task from the calendar view.

Click New Task or select a task in the calendar and click Edit Task or Delete Task.

The screenshot shows the 'Edit Task' dialog box. The task name is 'Final Daily Report' and the description is 'Final balance Daily Report'. The start date/time is '8/31/2020 11:14:03 AM'. The sequence is 'Job Sub Scenario' and the filter is 'BudgetV2'. The dialog box has tabs for Task, Schedule, and Advanced.

You can also double-click on the task to edit or you can right-click to edit or delete the task.

## Application Tools



Once the job runs, you can see the status of the job in Task Activity.

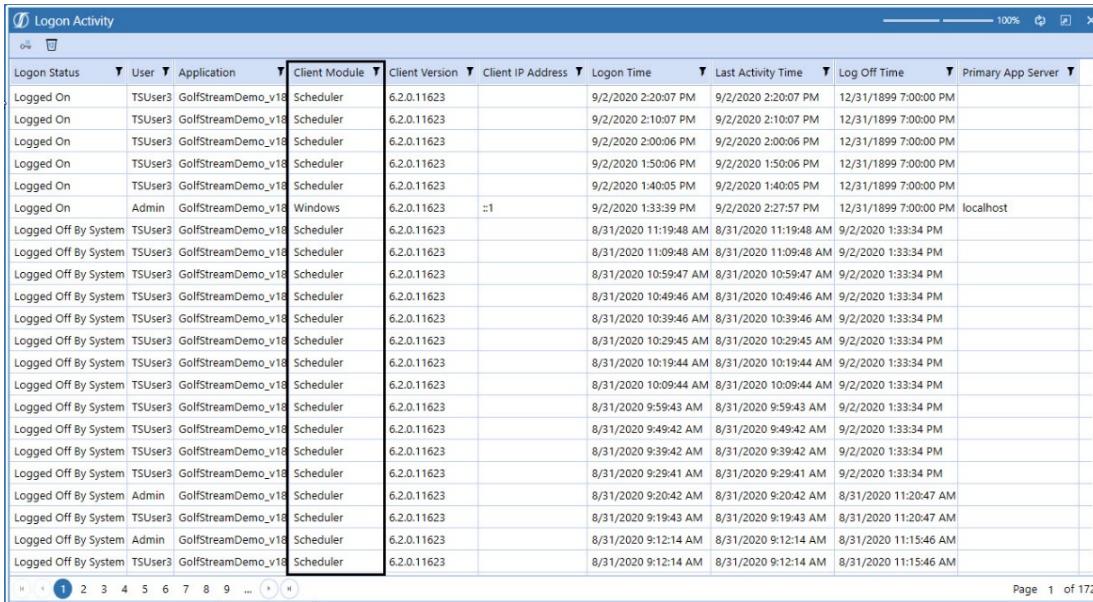
Task Type	Description	Duration	Task Status	User	Application	Server	Queued Time	Assigned Time	Start Time	End Time
Data Management Scheduled Task	TSM TB Every 3 Hours - Export Trial Balance	0:00:00:01.124	Completed	TSM	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 11:15:56 AM	8/12/2020 11:15:56 AM	8/12/2020 11:15:56 AM	8/12/2020 11:15:
Data Management Scheduled Task	TSM Daily Report - Export Trial Balance	0:00:00:00.844	Completed	TSM	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 11:14:25 AM	8/12/2020 11:14:26 AM	8/12/2020 11:14:	
Data Management Scheduled Task	Export Trial Balance 30mins - Export Trial Balance	0:00:00:00.573	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 11:10:55 AM	8/12/2020 11:10:56 AM	8/12/2020 11:10:56 AM	8/12/2020 11:10:
Data Management Scheduled Task	Daily Export of Trial Balance - Export Trial Balance	0:00:00:00.307	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 11:07:24 AM	8/12/2020 11:07:26 AM	8/12/2020 11:07:	
Data Management Scheduled Task	CEO Every 5 Mins - Export Trial Balance2	0:00:00:00.000	Failed	CEO	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 11:00:54 AM	8/12/2020 11:00:55 AM	12/31/1899 7:00:00 PM	8/12/2020 11:00:
Data Management Scheduled Task	Execute Export TB Daily - Export Trial Balance	0:00:00:00.960	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 10:58:24 AM	8/12/2020 10:58:25 AM	8/12/2020 10:58:25 AM	8/12/2020 10:58:
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.500	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 10:58:24 AM	8/12/2020 10:58:25 AM	8/12/2020 10:58:	
Data Management Scheduled Task	CEO Every 5 Mins - Export Trial Balance2	0:00:00:00.000	Failed	CEO	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 10:55:54 AM	8/12/2020 10:55:55 AM	12/31/1899 7:00:00 PM	8/12/2020 10:55:
Data Management Scheduled Task	CEO Every 5 Mins - Export Trial Balance2	0:00:00:00.000	Failed	CEO	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 10:50:54 AM	8/12/2020 10:50:55 AM	12/31/1899 7:00:00 PM	8/12/2020 10:50:
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.340	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 10:50:24 AM	8/12/2020 10:50:25 AM	8/12/2020 10:50:	
Data Management Scheduled Task	CEO Every 5 Mins - Export Trial Balance2	0:00:00:00.000	Failed	CEO	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 10:45:54 AM	8/12/2020 10:45:55 AM	12/31/1899 7:00:00 PM	8/12/2020 10:45:
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.683	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 10:42:23 AM	8/12/2020 10:42:24 AM	8/12/2020 10:42:24 AM	8/12/2020 10:42:
Data Management Scheduled Task	CEO Every 5 Mins - Export Trial Balance2	0:00:00:00.000	Failed	CEO	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 10:40:53 AM	8/12/2020 10:40:54 AM	12/31/1899 7:00:00 PM	8/12/2020 10:40:
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.477	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 10:34:05 AM	8/12/2020 10:34:07 AM	8/12/2020 10:34:07 AM	8/12/2020 10:34:
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.860	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 10:26:03 AM	8/12/2020 10:26:04 AM	8/12/2020 10:26:04 AM	8/12/2020 10:26:
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.460	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 10:18:00 AM	8/12/2020 10:18:01 AM	8/12/2020 10:18:01 AM	8/12/2020 10:18:
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.427	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 10:10:27 AM	8/12/2020 10:10:29 AM	8/12/2020 10:10:29 AM	8/12/2020 10:10:
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.390	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 10:02:25 AM	8/12/2020 10:02:26 AM	8/12/2020 10:02:26 AM	8/12/2020 10:02:
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.433	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 9:54:24 AM	8/12/2020 9:54:25 AM	8/12/2020 9:54:25 AM	8/12/2020 9:54:2
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.410	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 9:46:22 AM	8/12/2020 9:46:22 AM	8/12/2020 9:46:22 AM	8/12/2020 9:46:2
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.374	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 9:38:18 AM	8/12/2020 9:38:20 AM	8/12/2020 9:38:20 AM	8/12/2020 9:38:2
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.316	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 9:30:12 AM	8/12/2020 9:30:12 AM	8/12/2020 9:30:12 AM	8/12/2020 9:30:1
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.293	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 9:22:06 AM	8/12/2020 9:22:07 AM	8/12/2020 9:22:07 AM	8/12/2020 9:22:0
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.256	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 9:14:06 AM	8/12/2020 9:14:07 AM	8/12/2020 9:14:07 AM	8/12/2020 9:14:0
Data Management Scheduled Task	TSM2 Export Trial Balance - Export Trial Balance2	0:00:00:00.000	Failed	TSM2	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 9:13:36 AM	8/12/2020 9:13:37 AM	12/31/1899 7:00:00 PM	8/12/2020 9:13:3
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.280	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 9:06:03 AM	8/12/2020 9:06:06 AM	8/12/2020 9:06:06 AM	8/12/2020 9:06:0
Data Management Scheduled Task	TSM2 Export Trial Balance - Export Trial Balance2	0:00:00:00.000	Failed	TSM2	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 9:03:35 AM	8/12/2020 9:03:36 AM	12/31/1899 7:00:00 PM	8/12/2020 9:03:3
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.297	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 8:58:05 AM	8/12/2020 8:58:06 AM	8/12/2020 8:58:06 AM	8/12/2020 8:58:0
Data Management Scheduled Task	TSM2 Export Trial Balance - Export Trial Balance2	0:00:00:00.000	Failed	TSM2	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 8:53:34 AM	8/12/2020 8:53:36 AM	12/31/1899 7:00:00 PM	8/12/2020 8:53:3
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.426	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 8:50:04 AM	8/12/2020 8:50:05 AM	8/12/2020 8:50:04 AM	8/12/2020 8:50:0
Data Management Scheduled Task	TSM2 Export Trial Balance - Export Trial Balance2	0:00:00:00.000	Failed	TSM2	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 8:49:34 AM	8/12/2020 8:49:35 AM	12/31/1899 7:00:00 PM	8/12/2020 8:49:3
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.306	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 8:42:04 AM	8/12/2020 8:42:05 AM	8/12/2020 8:42:05 AM	8/12/2020 8:42:0
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.386	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 8:34:03 AM	8/12/2020 8:34:04 AM	8/12/2020 8:34:04 AM	8/12/2020 8:34:0
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.434	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 8:26:30 AM	8/12/2020 8:26:31 AM	8/12/2020 8:26:31 AM	8/12/2020 8:26:3
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:04.677	Completed	Admin	GolfStreamDemo_v18	BWILMOTLAP	8/12/2020 8:18:27 AM	8/12/2020 8:18:29 AM	8/12/2020 8:18:29 AM	8/12/2020 8:18:2
Data Management Scheduled Task	OneTimeAdmin - Export Trial Balance	0:00:00:00.000	Failed	Admin	GolfStreamDemo_v18	RWII MOTI AP	8/12/2020 8:10:11 AM	8/12/2020 8:10:13 AM	12/31/1899 7:00:00 PM	8/12/2020 8:10:5

The Task Type is Data Management Scheduled Task.

The Description is the name of the task separated by a hyphen followed by the sequence.

# Logon Activity

1. Go to **System > Logon Activity**.
2. In the Client Module column, you can see that the log in was through Scheduler.



Logon Status	User	Application	Client Module	Client Version	Client IP Address	Logon Time	Last Activity Time	Log Off Time	Primary App Server
Logged On	TSUser3	GolfStreamDemo_v18	Scheduler	6.2.0.11623		9/2/2020 2:20:07 PM	9/2/2020 2:20:07 PM	12/31/1899 7:00:00 PM	
Logged On	TSUser3	GolfStreamDemo_v18	Scheduler	6.2.0.11623		9/2/2020 2:10:07 PM	9/2/2020 2:10:07 PM	12/31/1899 7:00:00 PM	
Logged On	TSUser3	GolfStreamDemo_v18	Scheduler	6.2.0.11623		9/2/2020 2:00:06 PM	9/2/2020 2:00:06 PM	12/31/1899 7:00:00 PM	
Logged On	TSUser3	GolfStreamDemo_v18	Scheduler	6.2.0.11623		9/2/2020 1:50:06 PM	9/2/2020 1:50:06 PM	12/31/1899 7:00:00 PM	
Logged On	TSUser3	GolfStreamDemo_v18	Scheduler	6.2.0.11623		9/2/2020 1:40:05 PM	9/2/2020 1:40:05 PM	12/31/1899 7:00:00 PM	
Logged On	Admin	GolfStreamDemo_v18	Windows	6.2.0.11623	::1	9/2/2020 1:33:39 PM	9/2/2020 2:27:57 PM	12/31/1899 7:00:00 PM	localhost
Logged Off By System	TSUser3	GolfStreamDemo_v18	Scheduler	6.2.0.11623		8/31/2020 11:19:48 AM	8/31/2020 11:19:48 AM	9/2/2020 1:33:34 PM	
Logged Off By System	TSUser3	GolfStreamDemo_v18	Scheduler	6.2.0.11623		8/31/2020 11:09:48 AM	8/31/2020 11:09:48 AM	9/2/2020 1:33:34 PM	
Logged Off By System	TSUser3	GolfStreamDemo_v18	Scheduler	6.2.0.11623		8/31/2020 10:59:47 AM	8/31/2020 10:59:47 AM	9/2/2020 1:33:34 PM	
Logged Off By System	TSUser3	GolfStreamDemo_v18	Scheduler	6.2.0.11623		8/31/2020 10:49:46 AM	8/31/2020 10:49:46 AM	9/2/2020 1:33:34 PM	
Logged Off By System	TSUser3	GolfStreamDemo_v18	Scheduler	6.2.0.11623		8/31/2020 10:39:46 AM	8/31/2020 10:39:46 AM	9/2/2020 1:33:34 PM	
Logged Off By System	TSUser3	GolfStreamDemo_v18	Scheduler	6.2.0.11623		8/31/2020 10:29:45 AM	8/31/2020 10:29:45 AM	9/2/2020 1:33:34 PM	
Logged Off By System	TSUser3	GolfStreamDemo_v18	Scheduler	6.2.0.11623		8/31/2020 10:19:44 AM	8/31/2020 10:19:44 AM	9/2/2020 1:33:34 PM	
Logged Off By System	TSUser3	GolfStreamDemo_v18	Scheduler	6.2.0.11623		8/31/2020 10:09:44 AM	8/31/2020 10:09:44 AM	9/2/2020 1:33:34 PM	
Logged Off By System	TSUser3	GolfStreamDemo_v18	Scheduler	6.2.0.11623		8/31/2020 9:59:43 AM	8/31/2020 9:59:43 AM	9/2/2020 1:33:34 PM	
Logged Off By System	TSUser3	GolfStreamDemo_v18	Scheduler	6.2.0.11623		8/31/2020 9:49:42 AM	8/31/2020 9:49:42 AM	9/2/2020 1:33:34 PM	
Logged Off By System	TSUser3	GolfStreamDemo_v18	Scheduler	6.2.0.11623		8/31/2020 9:39:42 AM	8/31/2020 9:39:42 AM	9/2/2020 1:33:34 PM	
Logged Off By System	TSUser3	GolfStreamDemo_v18	Scheduler	6.2.0.11623		8/31/2020 9:29:41 AM	8/31/2020 9:29:41 AM	9/2/2020 1:33:34 PM	
Logged Off By System	Admin	GolfStreamDemo_v18	Scheduler	6.2.0.11623		8/31/2020 9:20:42 AM	8/31/2020 9:20:42 AM	8/31/2020 11:20:47 AM	
Logged Off By System	TSUser3	GolfStreamDemo_v18	Scheduler	6.2.0.11623		8/31/2020 9:19:43 AM	8/31/2020 9:19:43 AM	8/31/2020 11:20:47 AM	
Logged Off By System	Admin	GolfStreamDemo_v18	Scheduler	6.2.0.11623		8/31/2020 9:12:14 AM	8/31/2020 9:12:14 AM	8/31/2020 11:15:46 AM	
Logged Off By System	TSUser3	GolfStreamDemo_v18	Scheduler	6.2.0.11623		8/31/2020 9:12:14 AM	8/31/2020 9:12:14 AM	8/31/2020 11:15:46 AM	

Page 1 of 172

# Security Roles

1. Go to Application > Tools > Security Roles > Application User Interface Roles.
2. You must have the TaskSchedulerPage role to see the Task Scheduler page. You can only view all user tasks with this role.

The screenshot shows the 'Application Tools' interface with the 'Security Roles' page selected. The left sidebar contains navigation links for OnePlace, Application, System, and various tools like Cube, Workflow, Data Collection, Presentation, and Tools. Under Tools, 'Task Scheduler' is highlighted with a yellow background, and its sub-options 'Text Editor' and 'Load/Extract' are visible. The main content area is titled 'Security Roles' and lists application security roles and user interface roles. A specific row for 'TaskSchedulerPage' is highlighted with a blue border, showing it is associated with the 'TaskSchedulerReadOnly' role. The table structure is as follows:

Role	Granted To
Application Security Roles	
Application User Interface Roles	
ApplicationLoadExtractPage	Everyone
ApplicationPropertiesPage	Administrators
ApplicationSecurityRolesPage	Administrators
BookAdminPage	Administrators
BusinessRulesPage	Administrators
CertificationQuestionsPage	Administrators
ClientUpdaterPage	Administrators
ConfirmationRulesPage	Administrators
CubeAdminPage	Administrators
CubeViewsPage	Administrators
DashboardAdminPage	Administrators
DataManagementAdminPage	Administrators
TaskSchedulerPage	TaskSchedulerReadOnly
DataSourcesPage	Administrators
DimensionLibraryPage	Administrators
FxRatesPage	Administrators
FormTemplatesPage	Administrators
JournalTemplatesPage	Administrators
SpreadsheetPage	Everyone
TextEditor	Administrators
TimeDimProfilesPage	Administrators
TransformationRulesPage	Everyone
WorkflowChannelsPage	Administrators
WorkflowProfilesPage	Administrators

## Application Tools

---

3. Go to Application Security Roles and you will see ManageTaskScheduler and TaskScheduler.

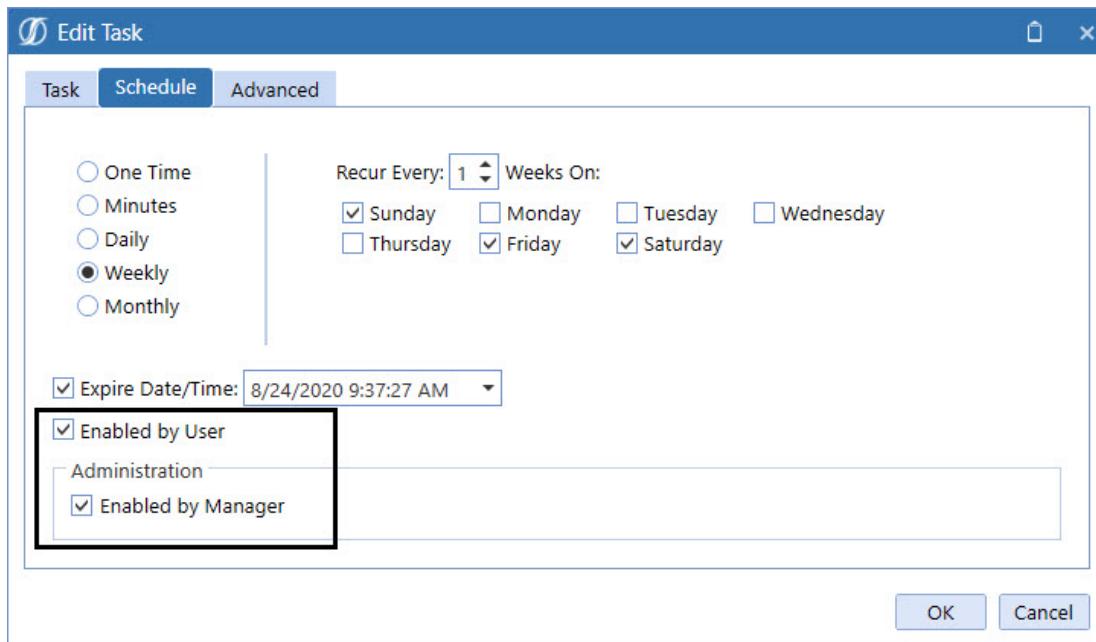
Security Roles	
<input type="checkbox"/> Application Security Roles	
AdministerApplication	Administrators
AdministerDatabase	Nobody
OpenApplication	Everyone
ModifyData	Everyone
ViewAllData	Administrators
CreateAuditAttachments	Administrators
CreateFootnoteAttachments	Administrators
CertifyAndLockDescendants	Administrators
UnlockAndUncertifyAncestors	Administrators
PreserveImportData	Administrators
RestoreImportData	Administrators
UnlockWorkflowUnit	Administrators
ViewSourceDataAudit	Everyone
EncryptBusinessRules	Nobody
ManageApplicationProperties	Administrators
ManageMetadata	Everyone
ManageFXRates	Everyone
LockFXRates	Administrators
UnlockFXRates	Administrators
ManageData	Administrators
ManageCubeViews	Everyone
ManageDataSources	Everyone
ManageTransformationRules	Everyone
ManageConfirmationRules	Everyone
ManageCertificationQuestions	Everyone
ManageWorkflowChannels	Administrators
ManageWorkflowProfiles	Administrators
ManageJournalTemplates	Everyone
ManageFormTemplates	Everyone
ManageApplicationDashboards	Administrators
ManageApplicationDatabaseFiles	Administrators
ManageTaskScheduler	TaskSchedulerManager
TaskScheduler	TaskSchedulerCreate

TaskScheduler allows you to create new tasks, edit your tasks, validate tasks if the setting is on. You can view all user tasks but only edit your own. You will not have access to load and extract. You cannot change the task name.

## Application Tools

---

ManageTaskScheduler allows you to create your own tasks, view every task no matter who created it, edit your own tasks and other user's tasks, delete or disable your own tasks. You cannot enable or disable a task that is not your own, but you can disable the user's task in the Administration section. You can load and extract. You cannot change the task name.



## Load and Extract

If you are a manager, you can load additional task scheduler files.

## Load

1. Go to **Application > Tools > Load/Extract**.
2. Select the file to load.
3. Click **Open**.
4. Click **Load**.

## Application Tools

---

### 5. View in the Task Scheduler.

User Name	Name	Description	Sequence	Schedule	Next Start Date/Time	Last Start Date/Time	State	Count	Edit	Delete
Admin	AllMonthsDaysDayOf		Export Trial Balance	Monthly			Enabled	0		
Admin	CEOtestTask	Export Stage Archives to Share Drive	Export Stage Archives	Weekly			Enabled	0		
Admin	CountMyRunsBy\$Mins		Export Trial Balance	Minutes			Enabled	0		
Admin	Daily Export of Trial Balance	Daily Export of Trial Balance Actuals	Export Trial Balance	Weekly			Enabled	0		
Admin	Daily TB Export	Export Trial Balance Daily	Export Trial Balance	Daily			Enabled	0		
Admin	DFPA Monthly	Export TB Monthly	Export Trial Balance	Monthly			Enabled	0		
Admin	DFPA Onetime	OneTime Export Stage Archive	Export Stage Archives	Daily			Enabled	0		
Admin	Execute Export TB Daily	Daily Export Trial Balance Actuals DM Se...	Export Trial Balance	Weekly			Enabled	0		
Admin	Export Trial Balance 30mins	Exports Trial Balance Every 30mins	Export Trial Balance	Minutes			Enabled	0		
Admin	FirstToLastSepDec	FirstToLastSepThruDec	Export Trial Balance2	Monthly			Enabled	0		
Admin	Karin2		Export Trial Balance	One Time			Enabled	0		
Admin	KarinFrench		Export Trial Balance	Daily			Enabled	0		
Admin	MonthlyDatesUI		Export Trial Balance	Monthly			Enabled	0		
Admin	Monthly	Monthly Test	Export Trial Balance	Monthly			Enabled	0		
Admin	MonthlyOn		Export Trial Balance	Monthly			Enabled	0		
Admin	MonthlyOn AMVersion		Export Trial Balance	Monthly			Enabled	0		
Admin	MonthlyOnCheckMikes		Export Trial Balance	Monthly			Enabled	0		
Admin	NewTaskForSarah		Export Trial Balance	Weekly			Enabled	0		
Admin	NewTestTask		Export Trial Balance	One Time			Enabled	0		
Admin	OneTimeAdmin	TS OneTime Admin	Export Trial Balance	Minutes			Enabled	0		
Admin	SendEmailNotification	Expiring Users	Export Trial Balance	Weekly			Enabled	0		
Admin	SequenceWithh0Step		Export Trial Balance2	Daily			Enabled	0		
Admin	SPFA Daily	Daily Export of Stage Dta	Export Stage Data	Daily			Enabled	0		
Admin	TestAllDaysOMonth		Export Trial Balance	Monthly			Enabled	0		
Admin	TestMyEditTaskForNoStartTimeChange		Export Trial Balance	One Time			Enabled	0		
Admin	TestRefreshGridView		Export Trial Balance	One Time			Enabled	0		
Admin	TestSecurityWithAdmin		Export Trial Balance	One Time			Enabled	0		
Admin	TestTime		Export Trial Balance	One Time			Enabled	0		
Admin	ValidateMytask		Export Trial Balance	Weekly			Enabled	0		
Admin	Weekly	Weekly	Export Trial Balance	Weekly			Enabled	0		
Admin	Weekly Export of Trial Balance	Weekly Export of Trial Balance Wed Th Fri	Export Trial Balance	Weekly			Enabled	0		
BWilmot	FrenchCalendar		Export Trial Balance2	Daily			Enabled	0		
BWilmot	TestDailyByMin		Export Trial Balance	Minutes			Enabled	0		
CEO	CEO Every 5 Mins	change CEO	Export Trial Balance2	Minutes			Enabled	0		
CEO	CEOtestTask2	Daily Export of Trial Balance	Export Trial Balance2	Weekly			Enabled	0		
Sr_FPRA	SPFA Onetime	Brian as Admin Update	Execute Batch File Load	One Time			Enabled	0		

## Extract

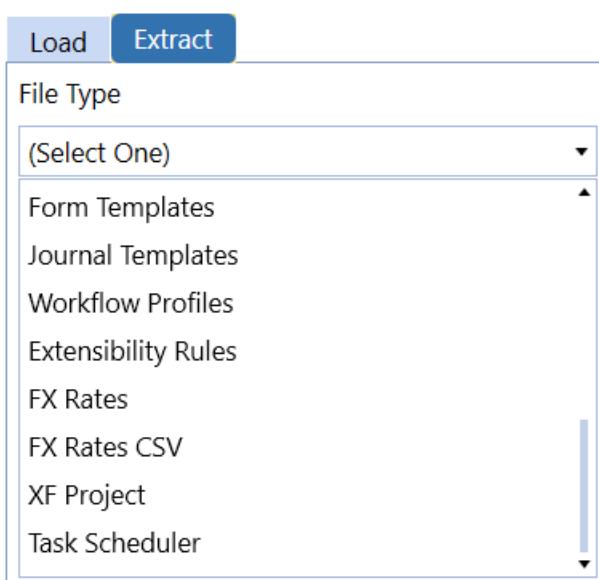
1. Go to Application > Tools > Load/Extract.

2. Click Extract.

## Application Tools

---

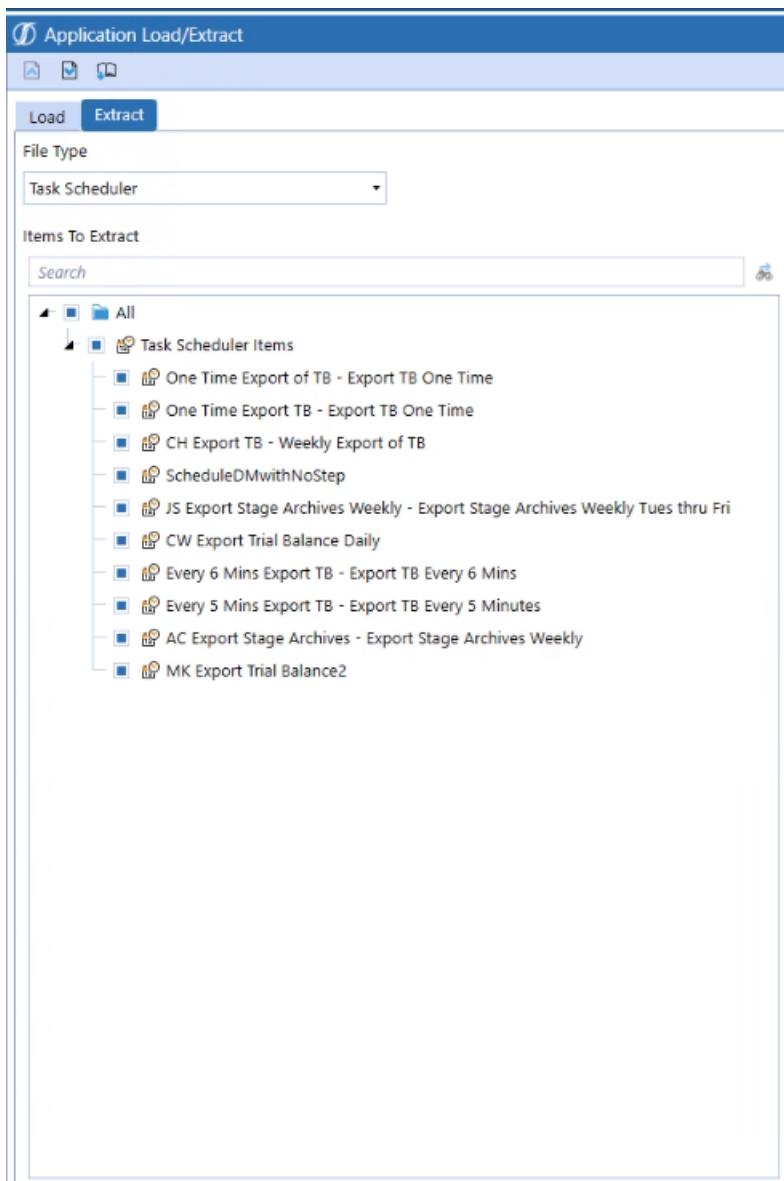
3. Select the File Type of Task Scheduler.



4. Select the task and click **Extract**.

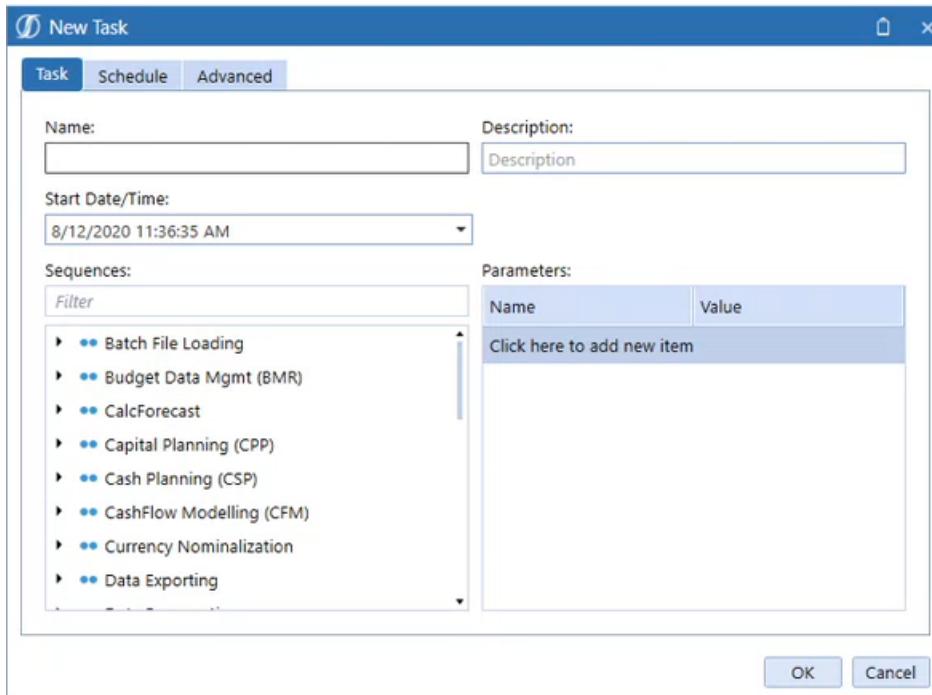
## Application Tools

---



# Create a New Task

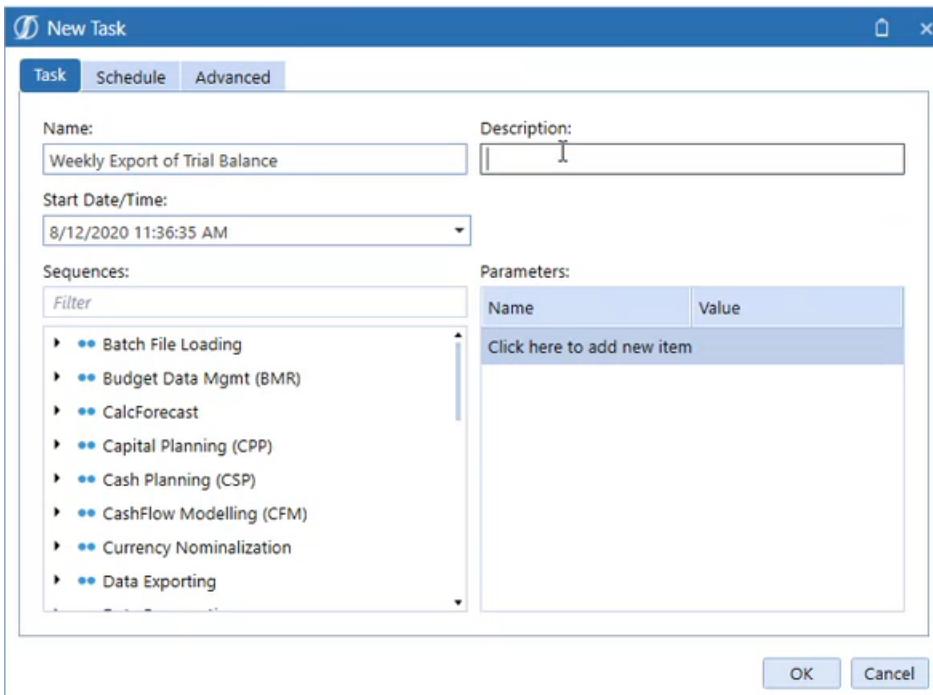
1. Click Create Scheduled Task  . The New Task dialog box opens.



2. Click **Task**.

## Application Tools

---

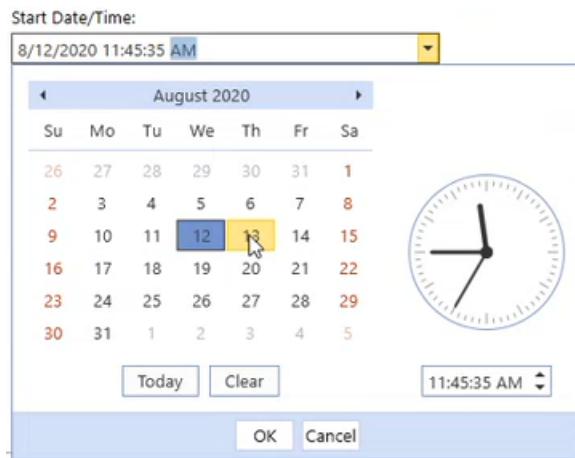


3. Enter the following details:

- a. Name of the task.
- b. The Description.
- c. Start Date/Time, which you can enter or click the drop-down to select. This should be set to the time you want the task to start running. If you don't specify a time, the value defaults to the current date and time.

## Application Tools

---



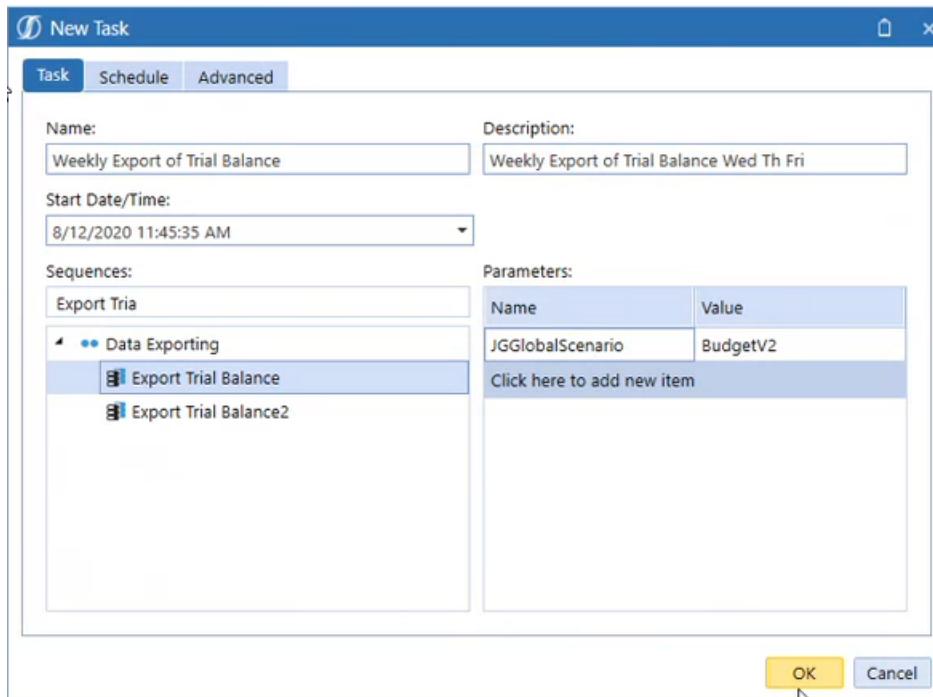
- d. Select the data management Sequence either by scrolling through the list or entering the sequence in the Filter field.

Sequences:	Parameters:
Export Trial • Data Exporting Export Trial Balance Export Trial Balance2	Name Value JGGlobalScenario BudgetV2 Click here to add new item

- e. When you select a sequence, if there are Parameters that have been set up for the sequence, they will show. You can add additional validated parameter settings, if the parameters are not valid, the job will fail.

## Application Tools

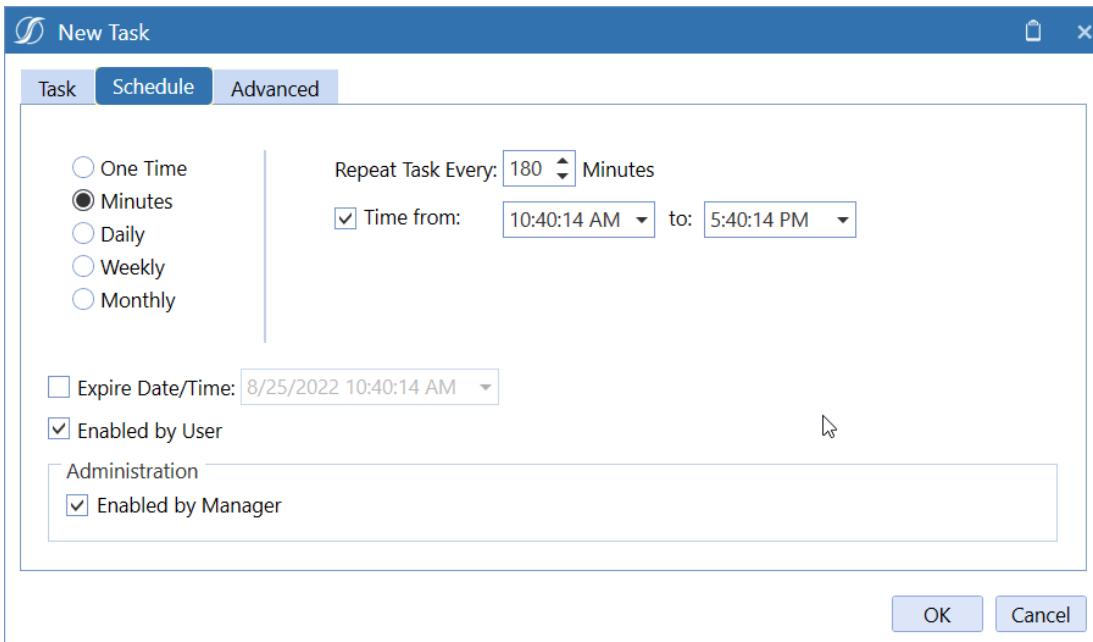
---



- f. If you click **OK** before you schedule it, the default is to run it one time.
4. Click **Schedule**.

## Application Tools

---



5. Enter Schedule details:

- One Time triggers the task to run once based on the time in the Start/Date Time: field on the Task tab.
- Minutes triggers the task to run on a recurring basis from 5 – 180 minutes. You can set tasks to run during predetermined times by typing the start time in the Time From: field, then typing the stop time in the To: field. By default, the Time from: field is unchecked.

If you create a task, for example, that is set to run every 30 minutes, starting at 2:30pm and ending at 5:30pm, the first run of the task occurs at 2:30pm and runs every 30 minutes during that timeframe.

**NOTE:** Calendar entries are created in the Calendar view even though they may fall outside of the selected run time frame. This means, for example, that a task scheduled to run every 30 minutes between the hours of 2:00pm and 5:00pm, will display all day every 30 minutes.

- Daily you can choose how many times it recurs.
- Weekly you can choose how many times it recurs and the days that it runs.

## Application Tools

---

- e. Monthly you can choose how many times it recurs and the days that it runs.
  - f. Expire Date/Time when and what time it should end.
  - g. Enabled if checked to run.
  - h. Administration Enabled if checked to run.
6. Click **Advanced** to set the number of times to retry a task if it fails, the maximum is three.



7. Click **OK** and the new task has been added to the Grid View and the Calendar View.

The screenshot shows the 'Task Scheduler' grid view with several tasks listed. A modal dialog box is open in the foreground, displaying the message 'Successfully Saved 'Admin Export TB Weekly''. The grid view columns include User Name, Name, Description, Sequence, Schedule, Expire Date/Time, State, Edit, and Delete.

User Name	Name	Description	Sequence	Schedule	Expire Date/Time	State	Edit	Delete
Admin	Admin Export TB Weekly	Export Trial Balance Weekly Fri_Sun	Export Trial Balance	One Time	8/24/2020 9:37:27 AM	Enabled		
Admin	Mgr Export Stage Archives Weekly	Export Stage Archives Weekly Fri_Sun	Export Stage Archives	Weekly	12/31/2200 12:00:00 AM	Enabled		
Admin	Weekly TB	Weekly Trial Balance Fri_Sun	Export Trial Balance	Weekly	12/31/2200 12:00:00 AM	Enabled		
Admin	Yearly Full Consolidation	Yearly Full Consolidation Friday	Full Consolidation	Monthly	8/21/2021 7:26:19 PM	Enabled		
TSUser3	TSUser3 Export Budget Data Mgmt	Export Budget Data Mgmt Act PY...	Export Actual (Prior Year)	Weekly	12/31/2200 12:00:00 AM	Enabled		
TSUser3	TSUser3 Export Stage Archives Daily	Export Stage Archives Daily	Export Stage Archives	Daily	12/31/2200 12:00:00 AM	Enabled		
TSUser3	TSUser3 Export Trial Balance	Export Trial Balance		Minutes	12/31/2200 12:00:00 AM	Enabled		

## Text Editor

This is used to create, edit and view text documents like those created in Microsoft Word. This component only works in the OneStream Windows App version.

The OneStream Windows App Text Editor feature can perform many of the tasks that can be completed in Microsoft Word, with some limitations. These limitations include, but are not limited to:

## Application Tools

---

- Mail Merge
- Application of pre-defined Styles
- The Shape Fill and Shape Outline tools
- Previous and Next Comment buttons
- Spelling
- Insert Fields
- Add Watermarks

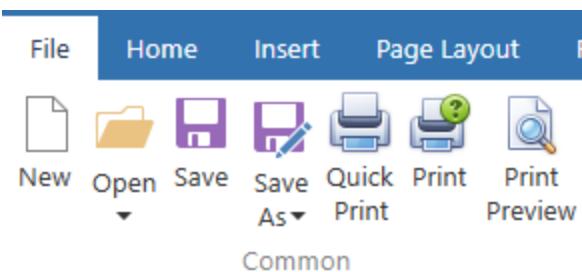
The following shortcut keys function differently in the Text Editor feature than they do in Microsoft Word. The Text Editor functionality is listed below:

- CTRL+N – Used to create a new blank document. If an existing document exists and has not been saved, using this keystroke combination will close the existing document without saving and create a new document.
- SHIFT + F3 – Used to launch the Find command
- SHIFT + F4 – Used to change case of selected word
- CTRL+SHIFT+Page Up – Used to move the cursor to the previous page
- ALT+I – Used to increase the letter/number increments within an outline or numbered list

## Text Editor Ribbon

The ribbon is organized as follows:

### Common



## Application Tools

---

### New

This creates a new text document. If clicked when an open document has unsaved changes, a window appears asking if the current changes should be saved before the new document is created.

### Open

File type to open

### Local Folder

Select a file to be opened from a location on the local computer/network

### File System

Select a file to be opened from a location within the File Explorer

**NOTE:** Displayed files can also be opened from here or from OnePlace | Documents by right clicking and selecting one of the three options – “Open in Text Editor Page”; “Open” (opens file in Compatibility mode, directly in Word if the application is found on local computer,) or “Open With...” (user specifies program).

### Application Dashboard File

Select a file to be opened from an Application Dashboard

### System Dashboard file

Select a file to be opened from a System Dashboard

### Save

Save changes to the open file using the current file name. Only available after the file has been given a name.

### Save As

File type to be saved as

### Local Folder

Select location on local computer/network to save a file to

### File System

Select location within File Explorer to save a file to

### Application Dashboard

Select Application Dashboard location to save a file to

### System Dashboard file

Select System Dashboard location to save a file to

### Quick Print

Send the document to the default printer without changing any printer/printing properties.

## Application Tools

---

### Print

Displays the Print dialog box for setting options to print the displayed document.

### Print Preview

Displays the Print Preview dialog box that shows how printed document looks. The Print Preview Ribbon associated with this button remembers the setting used last that dictates if it is hidden or displayed. The next time Print Preview is opened, the Ribbon is initialized the same way it was left.

## Clipboard



### Paste

Inserts copied data into the document

### Cut

Removes and transfers selected data from the document to the clipboard for placement in a different location.

### Copy

Copies selected data to the clipboard for inclusion in a different location.

### Paste Special

Displays the Paste Special dialog box for additional pasting options.

## Font



Provides options for changing the text found in the document. Examples include bold, italic, underline and text color.

## Application Tools

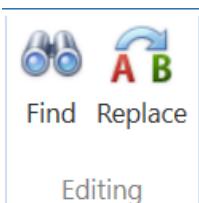
---

### Paragraph



Provides options for formatting text in the document. Examples include bullets, numbering, indenting and paragraph alignment

### Editing



#### Find

Enter text to be located within the document.

#### Replace

Enter information to replace located text within the document.

### Pages



Page

Pages

#### Page

Insert a page break at the current location within the document.

### Tables



Table

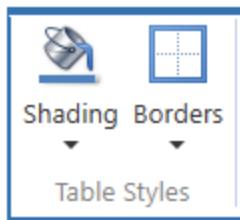
Tables

#### Table

Inserts a table at the current location within the document.

### Table Styles

This toolbar within the Design tab only displays when working within a table.



#### Shading

Used to add color to the background of the selected cells.

#### Borders

Used to customize the borders of the selected cells. Used in combination with the Pen Color button.

#### Draw Borders

This toolbar within the Design tab only displays when working within a table.



## Application Tools

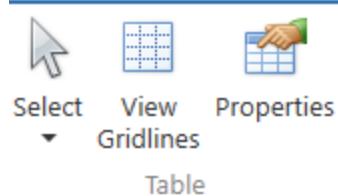
---

### Pen Color

Used in combination with the Borders button. This button changes the color of the border lines selected when using the Borders button. Click this button first to select the border color, then use the Borders button to define where border lines should be displayed.

### Table

This toolbar within the Layout tab only displays when working within a table.



#### Select

Used to select the current cell, column, row or entire table.

#### View Gridlines

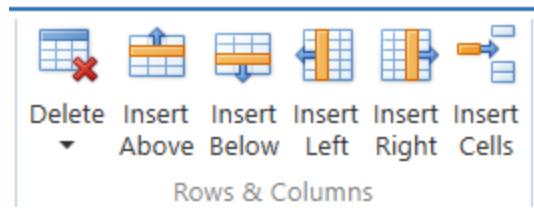
Used to show or hide the gridlines within a table. When turned on, the gridlines only appear where the display of cell borders has been turned off.

#### Properties

Used to display the Table Properties dialog box. Advanced formatting options such as margins, alignment, text wrapping, borders and shading can all be managed from here.

## Rows & Columns

This toolbar within the Layout tab only displays when working within a table.



#### Delete

Used to delete cells, rows, columns or the entire table.

#### Insert Above

Used to add a new row directly above the selected cell.

## Application Tools

---

### Insert Below

Used to add a new row directly below the selected cell.

### Insert Left

Used to add a new column directly to the left of the selected cell.

### Insert Right

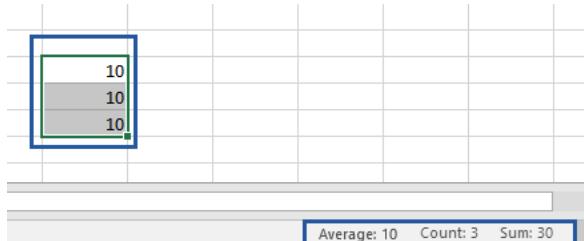
Used to add a new column directly to the right of the selected cell.

### Insert Cells

Used to insert a single cell into the table. The Insert Cell dialog box appears, with options to shift cells right, shift cells down, insert an entire row or insert an entire column.

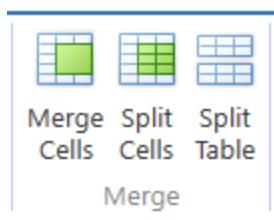
### Sum Status Bar

If you highlight cells or group of cells in Spreadsheet, you can see the Average, Count, and Sum values at the bottom of the spreadsheet.



## Merge

This toolbar within the Layout tab only displays when working within a table.



### Merge Cells

Used when two or more cells are selected to join/merge them into one cell.

### Split Cells

Used to split the selected cell(s) into smaller cells. The Split Cells dialog box displays and the number of new columns and rows needed can be entered.

## Application Tools

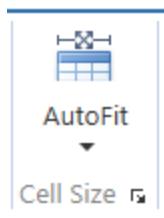
---

### Split Table

Used to split the table selected into two tables. The row that the selected cell(s) belongs to will become the first row of the new table that is created.

### Cell Size

This toolbar within the Layout tab only displays when working within a table.

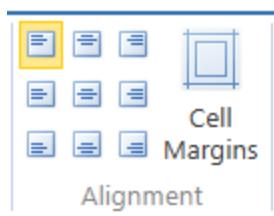


### AutoFit

Used to automatically resize the width of the column based on the text within. Options include autofitting each cell based on the content within it, autofitting the table to take up the width of the window and setting the columns to a fixed width.

### Alignment

This toolbar within the Layout tab only displays when working within a table.



Various options are available to adjust the alignment of text with the table cells. Examples include align top left, center left, bottom left, top center, center, bottom center, top right, center right, bottom right. Cell margins and spacing can be set using the Cell Margins button.

### Illustrations



Picture

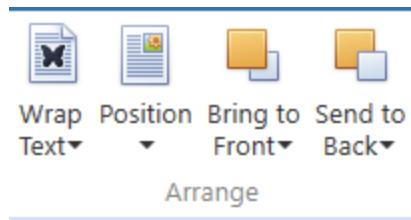
### Illustrations

#### Picture

Inserts a picture file at the current location within the document. Standard picture file options are available to select from.

### Arrange

This toolbar within the Format tab only displays when an inserted picture has been selected.



#### Wrap Text

Changes the way text wraps around the selected object. Six different wrap options are available including Square, Tight, Through, Top and Bottom, Behind Text and In Front of Text.

#### Position

Positions the selected object on the page. Text is automatically set to wrap around the object.

#### Bring to Front

Brings the selected object forward so that it is hidden by fewer objects in front of it. Three options are available including Bring Forward, Bring to Front and Bring in Front of Text.

#### Send to Back

Sends the selected object backward so that it is hidden by the objects in front of it. Three options are available including Send Backward, Send to Back and Send Behind Text.

### Links



Bookmark Hyperlink

#### Links

##### Bookmark

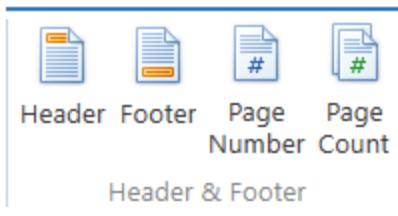
Creates a bookmark for selected text and assigns a name to that specific area of the document. Hyperlinks can be made to move directly to that location.

##### Hyperlink

Creates a link to a webpage, a file, an application, an email address or a place in the same document.

**NOTE:** To send an email enter the following in the address field –  
mailto:emailusername@domainhostname. For example,  
mailto:jdoe@onestreamsoftware.com

### Header & Footer



##### Header

Insert a Header into the document or go to the Header section if a Header already exists within the document.

##### Footer

Insert a Footer into the document or go to the Header section if a Header already exists within the document.

## Application Tools

---

### Page Number

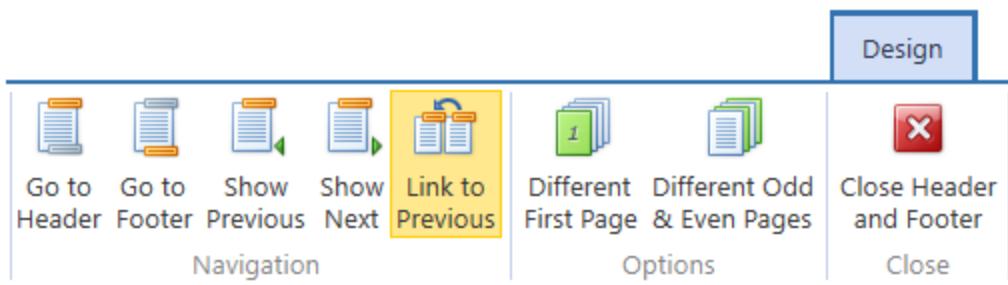
Inserts the current page number, wherever the cursor is located within the Header/Footer areas of the document.

### Page Count

Inserts the total number of pages in the document, wherever the cursor is located within the Header/Footer areas of the document.

## Navigation

This toolbar within the Design tab only displays when working within the Header/Footer areas of the document.



### Go to Header/Go to Footer

Activates the Header/Footer section on the page so it can be edited.

### Show Previous

If the document has been broken into sections, this navigates to the previous section's Header/Footer.

### Show Next

If the document has been broken into sections, this navigates to the next section's Header/Footer.

### Link to Previous

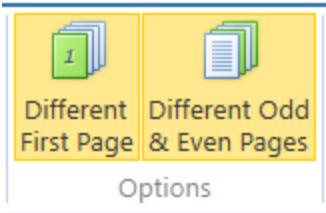
Creates a link to the previous section so the Header/Footer in this section contains the same content as the previous section.

## Options

This toolbar within the Design tab only displays when working within the Header/Footer areas of the document.

## Application Tools

---



### Different First Page

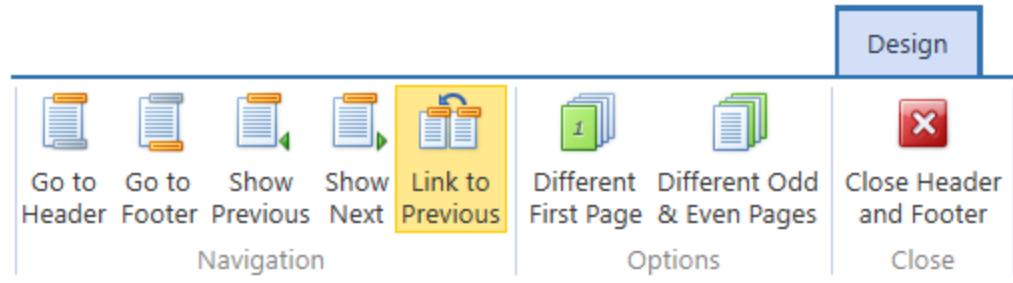
Used to insert a unique Header/Footer for the first page only of the document

### Different Odd & Even Pages

Used to insert different Header/Footers on even and odd pages.

## Close

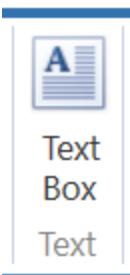
This toolbar within the Design tab is only displayed when working within the Header/Footer areas of the document.



### Close Header and Footer

Used to close out of the Header/Footer Tools and return to the document.

## Text



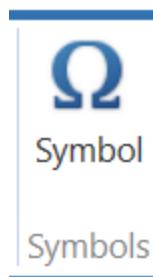
## Application Tools

---

### Text Box

Inserts a text box into the document.

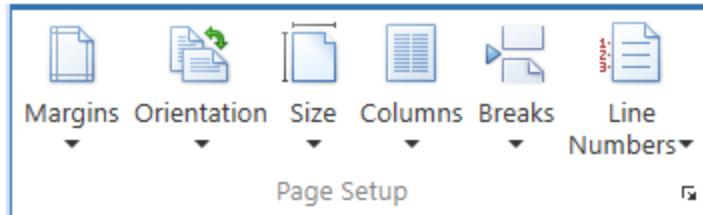
### Symbols



### Symbol

Inserts standard characters and those not found on the keyboard into the document.

### Page Setup



#### Margins

Used to set pre-defined or custom margin sizes for the entire document, or the current section.

#### Orientation

Used to change the pages between landscape and portrait layouts.

#### Size

Used to set the paper size for the current section.

#### Columns

Used to split the text into two or more columns.

#### Breaks

Used to insert page, column or section breaks (Next, Continuous, Even or Odd).

## Application Tools

---

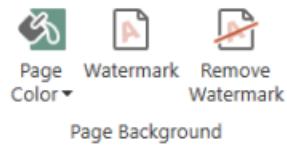
### Line Numbers

Used to insert line numbers in the margins on each line of the document. Options include restarting numbering on each page, each section, suppressing numbers for current paragraph and custom settings.

### Page Setup

Opens the Page Setup dialog box where margins, orientation, paper size, headers/footers and header/footer placement can all be set/edited.

## Page Background



### Page Color

Used to choose a background color for all the pages in the document.

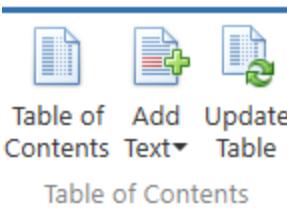
### Watermark

Insert and configure text and image watermarks within Text Editor documents. See "Text Editor Watermarks" on page 839.

### Remove Watermark

Remove a watermark from the Text Editor document.

## Table of Contents



### Table of Contents

Used to create a Table of Contents for the document.

### Add Text

Used to add the current paragraph into the Table of Contents

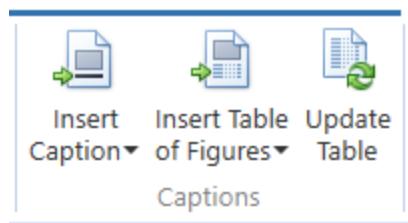
## Application Tools

---

### Update Table

Used to update the Table of Contents so that all entries reference the correct page number.

## Captions



### Insert Caption

Used to add a caption to a picture or another image. It is used to describe the object associated with it and appears below the object. Captions can be created for the following items:

#### Figures Caption

Used to add a caption to a figure or picture within the document.

#### Tables Caption

Used to add a caption to a table within the document.

#### Equations Caption

This option is not used or supported because equations cannot be created in the OneStream Windows App Text Editor.

#### Insert Table of Figures

Similar to the Table of Contents, this is used to insert a table that lists all the figures or tables that are contained in the document. The following types of tables can be created.

#### Table of Figures

Includes a list of all the Figures in the document that have captions associated with them.

#### Table of Tables

Includes a list of all the Tables in the document that have captions associated with them.

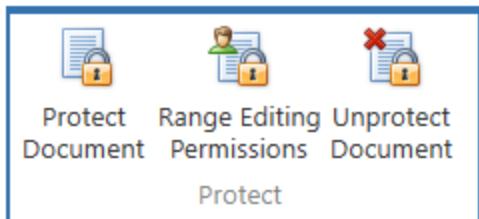
#### Table of Equations

This option is not used or supported because equations cannot be created in the OneStream Windows App version of Text Editor.

#### Update Table

Used to update the Table of Figures to include all entries within the document.

### Protect



#### Protect Document

Used to add a password to the document, so that when protected, only users who know the password can edit the document or add comments. There are options for both of these, and only one can be implemented at a time.

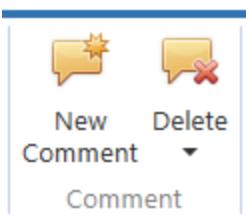
#### Range Editing Permissions

This is not supported or used in the OneStream Windows App Text Editor.

#### Unprotect Document

Used to unprotect a document that has been locked for edited. Click the button and provide the previously created password to unprotect the document.

### Comment



#### New Comment

Used to add comments to the document. Automatically numbers the comments as they are added to the document. The comment located at the top of the document is numbered 1 and additional comments are numbered sequentially down the document based on their position. If a comment is added or deleted, remaining comments in the document are automatically renumbered based on their location within the document.

#### Delete

Used to delete comments in the document. Options include the following:

#### Delete Comment

Used to delete the selected comment only.

## Application Tools

---

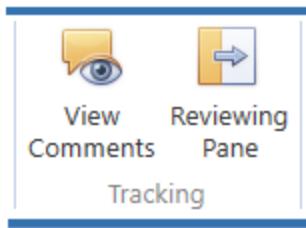
### Delete All Comments Shown

This option is not supported in the Text Editor for OneStream Windows App.

### Delete All Comments

Used to delete all comments in the document.

## Tracking



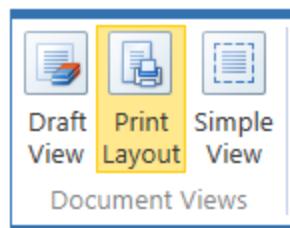
### View Comments

Used to show Comments that have been added to the document. They will be displayed in a column on the right side of the document.

### Reviewing Pane

Used to show/hide Comments within the document in a separate window. The window opens on the left side of the document and can be used to select comments.

## Document Views



### Draft View

Used to view the document as a draft for quicker editing. Certain features, such as Headers/Footers are not visible when using this view.

### Print Layout

Used to show how the document will look on the printed page.

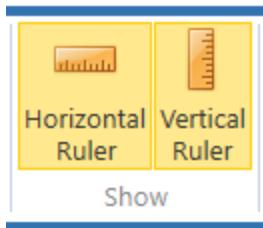
### Simple View

Used to show the document as a simple memo. Page Layout features are ignored in this view.

## Application Tools

---

### Show



#### Horizontal Ruler

Used to display the Horizontal Ruler, which is used to measure and line up objects in the document.

#### Vertical Ruler

Used to display the Vertical Ruler, which is used to measure and line up objects in the document.

### Zoom



#### Zoom Out

Used to change the view to see more of the page at a reduced size.

#### Zoom In

Used to change the view to get a close-up view of the document.

### Spell Checking

A text spell check feature is available only when using the Windows Application. This feature is set as a default to be inactive. To enable the Spell Check feature, users must have access to the Application / Tools / Text Editor. In the Text Editor Tool, the Review ribbon will allow the user to activate Spell Check using the Spell Check button.

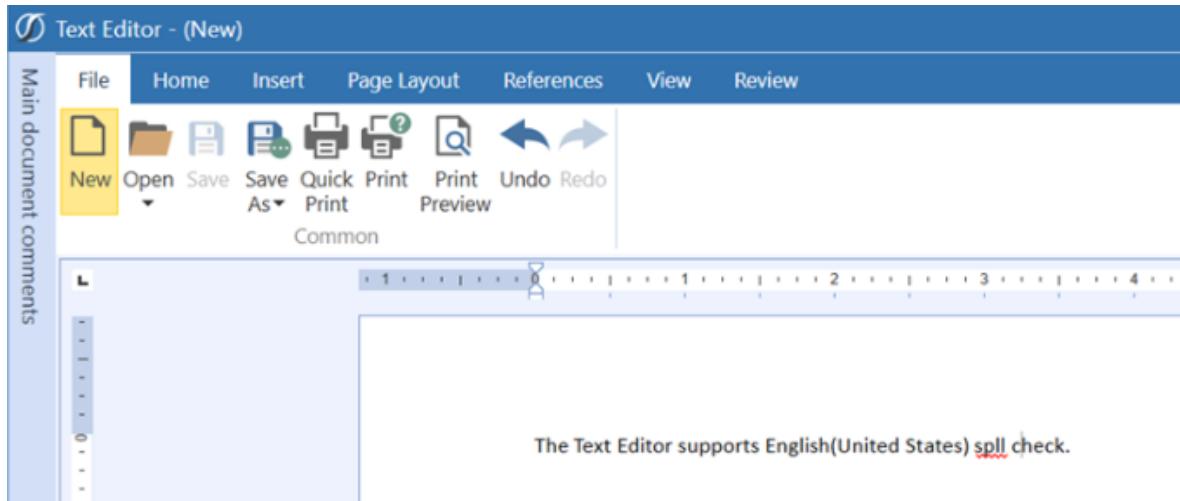
## Application Tools

---

The Spell Check feature is enabled for English Culture only. The culture is determined by each user's culture assigned in OneStream User Security. The culture is assigned to the OneStream application on the Application Server Configuration Utility as "en-US". Users with cultures other than English (United States) will not have Spell Check available.

### Spell Check in Text Editor

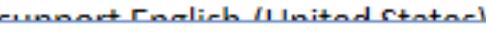
The Spell Check will be active as the user enters text, identifying any errors.

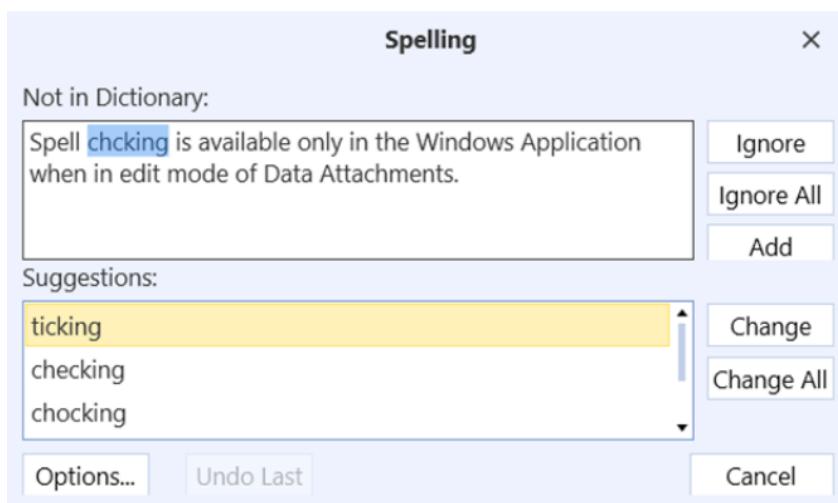
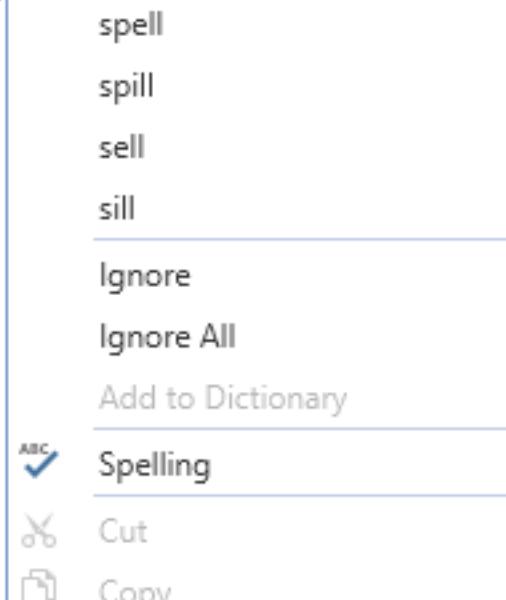


To access the Spell Check options, double-click to select the error, then a right-click will expose the Spell Check menu choices. Choosing the "Ignore" option will only be retained for the current session. Closing and re-opening to the edit mode will re-check any previously ignored items.

## Application Tools

---

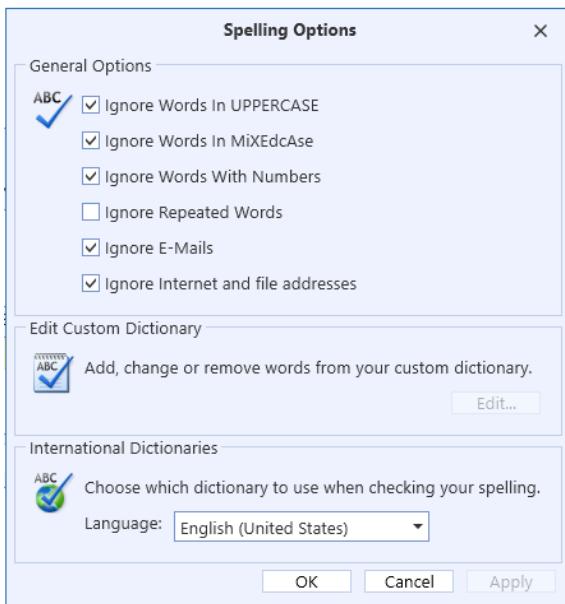
The Text Editor  spell check.



The Options button will allow the user to modify the Spell Check behavior within the current task session. The settings are not persisted as user preference.

## Application Tools

---



## Text Editor Watermarks

Watermarks help you to protect a company's assets, such as documents, from unauthorized use. In OneStream, you can create watermarks using images or text.

### Create an Image Watermark

1. In the Text Editor, click the **Page Layout** tab and then click **Watermark**. The Watermark dialog box opens.
2. Select the **Picture watermark** option.
3. Browse to an image on your local drive and select it.
4. Select the **Scale** option and adjust the image size by percentage, if needed.
5. Select the **Washout** checkbox to apply a washout effect to the image. Deselect to keep the original colors.
6. In the **Apply to** drop-down menu, select how to apply the watermark:

- **Whole document:** Applies the watermark to every page in the document
- **Section:** Applies the watermark to the section in which the cursor is currently positioned

**NOTE:** Sections are created using the Breaks menu on the Page Layout tab.

7. Click **Apply** or **OK**.

### Create a Text Watermark

1. In the Text Editor, click the **Page Layout** tab and then click **Watermark**. The Watermark dialog box opens.
2. Select the **Text watermark** option
3. Type the watermark text or select from the pre-defined options.
4. Apply **Font**, **Font Size**, and **Color** options as needed.
5. Select the **Semitransparent** checkbox to apply a semitransparent effect to the text. Deselect to keep the original opacity.
6. Select **Diagonal** or **Horizontal** layout.
7. In the **Apply to** drop-down menu, select how to apply the watermark:

- **Whole document:** Applies the watermark to every page in the document
- **Section:** Applies the watermark to the section in which the cursor is currently positioned

**NOTE:** Sections are created using the Breaks menu on the Page Layout tab.

8. Click **Apply** or **OK**.

### Modify a Watermark

You can modify a watermark by clicking the Watermark button. The Watermark dialog box opens where you can make modifications as needed.

### Remove a Watermark

1. In the Text Editor, click the **Page Layout** tab and then click **Remove Watermark**. The Watermark dialog box opens. This removes all watermarks from the document.
2. Alternatively, open the Watermark dialog box by clicking the **Watermark** button.
  - a. In the Watermark dialog box, select the **No Watermark** option.
  - b. Click **Apply** or **OK**.

### Load/Extract Application Artifacts

You can import and export sections of the Application using an XML Format.

**TIP:** Only Administrators have access to this portion of the tool.



#### Extract

Choose an item from the drop-down list, click the Extract icon to start the extract process, then name the output file.



#### Load

After browsing to the file, click the Load icon to initiate the process.



#### Extract and Edit

This option is available for all extracts and allows the ability for the end user to edit the XML file as needed.

These are the list of items that can be extracted:

#### Application Zip File (All except for Data and FX)

This option will export all the Application structures except for data and the FX rates. After clicking on the extract icon, the end user will be presented with a pop-up box to save it in a zipped format. After clicking yes, the end user can then name his/her file. This is the most complete option in the list and can be used to create a copy of the Application.

#### Application Security Roles

This covers all the security roles found under Application | Tools | Security Roles. The screen will display Items To Extract and by default, All is chosen. The defined roles will be exported accordingly.

### Application Properties

This covers all the Application Properties found under Application | Tools | Application Properties. After choosing this option, the screen will display Items To Extract and by default All is chosen.

### Workflow Channels

This covers all the Workflow Channels found under Application | Workflow | Workflow Channels. The screen will display Items To Extract and by default All is chosen.

### Metadata

This has multiple sections such as Business Rules, Time Dimension Profiles, Dimensions, and Cubes. The metadata can be found under Application | Cube | Dimension Library. OneStream can search for Dimension Member and Relationship changes by username and time stamp. For more details on the Find Modified Items feature, see Extracting and Loading Dimensions in "Implementing Security" on page 252.

### Cube Views

This has multiple sections such as Groups and Profiles which go together. Cube Views can be found under Application | Presentation | Cube Views. The screen will display Items To Extract and by default All is chosen. Choose specific items in each section, or turn the sections on or off.

### Data Management

This has multiple sections such as Groups and Profiles which go together. Data Management can be found under Application | Tools | Data Management. The screen will display Items To Extract and by default All is chosen. Choose specific items in each section, or turn the sections on or off.

### Application Workspaces

This has multiple sections such as Maintenance Units and Profiles. Under the Groups, Dashboard Components, Dashboard Adapters, and Dashboard Parameters go together. Workspaces can be found at Application | Presentations | Workspaces. The screen will display Items To Extract and by default All is chosen. Choose specific items in each section, or turn the sections on or off.

### Confirmation Rules

This has multiple sections such as Groups and Profiles which go together. Confirmation Rules can be found under Application | Workflow | Confirmation Rules. The screen will display Items To Extract and by default All is chosen. Choose specific items in each section, or turn the sections on or off.

### Certification Questions

This has multiple sections such as Groups and Profiles which go together. Certification Questions can be found under Application | Workflow | Certification Questions. The screen will display Items To Extract and by default All is chosen. Choose specific items in each section, or turn the sections on or off.

### Transformation Rules

This has multiple sections such as Business Rules, Groups, and Profiles which go together. Transformation Rules can be found under Application | Data Collection | Transformation Rules. The screen will display Items To Extract and by default All is chosen. Choose specific items in each section, or turn the sections on or off.

### Data Sources

This has multiple sections such as Business Rules and Data Sources which go together. Data Sources can be found under Application | Data Collection | Data Sources. The screen will display Items To Extract and by default All is chosen. Choose specific items in each section, or turn the sections on or off.

### Form Templates

This has multiple sections such as Groups and Profiles which go together. Form Templates can be found under Application | Data Collection | Form Templates. The screen will display Items To Extract and by default All is chosen. Choose specific items in each section, or turn the sections on or off.

### Journal Templates

This has multiple sections such as Groups and Profiles which go together. Journal Templates can be found under Application | Data Collection | Journal Templates. The screen will display Items To Extract and by default All is chosen. Choose specific items in each section, or turn the sections on or off.

### Workflow Profiles

This has multiple sections such as Workflow Profiles and Workflow Profile Templates. Workflow Profiles can be found under Application | Workflow | Workflow Profiles. The screen will display Items To Extract and by default All is chosen. Choose specific items in each section, or turn the sections on or off. When loading Workflow Profiles via an XML file extracted previously or from a different Application, the Load process clears old property settings if they are not specified in the loaded XML file. This approach ensures that any property edits made to the Workflow Profile upon extracting it are honored when the same XML file is re-loaded.

### Extensibility Rules

This only exports the Extensibility Rules in the Business Rules section; all others are exported with the object to which they are tied. (e.g., if there is an Account description rule under the Parser section, it will be exported with the Data Source.) Extensibility Rules can be found under Application | Tools | Business Rules | Extensibility Rules. The screen will display Items To Extract and by default All is chosen. Choose specific items in each section, or turn the sections on or off.

### FX Rates

Rates have their own Cube. This has multiple sections such as All FX Rate Types and All Time Periods. FX Rates can be found under Application | Cube | FX Rates. The screen will display Items To Extract and by default All is chosen. Choose specific items in each section, or turn the sections on or off.

### FX Rate CSV

This is a CSV extract of the application FX Rates which contains FxRateType, Time, SourceCurrency, DestCurrency, Amount, HasData. FX Rates can be found under Application | Cube | FX Rates.

# User Authentication and Single Sign-On

When you log into an application, your username and password are authenticated to confirm that you are a valid OneStream user.

If you are in a OneStream-hosted environment, see the *Identity and Access Management Guide* for information about authentication with OneStream IdentityServer.

If you are in a self-hosted environment, administrators can configure an environment to use native authentication, one external identity provider, or both native authentication and one external identity provider.

The following external identity providers are supported:

- Microsoft Active Directory (MSAD)
- Lightweight Directory Access Protocol (LDAP)
- Three OpenID Connect (OIDC) identity providers:
  - Azure Active Directory (Azure AD)
  - Okta
  - PingFederate
- SAML 2.0 identity providers (for example, Okta, PingFederate, Active Directory Federation Services [ADFS], and Salesforce)

For instructions on configuring authentication in a self-hosted environment, see the *Installation and Configuration Guide*.

# How Users are Configured for Authentication

When an administrator creates users in **System > Security**, they specify authentication settings.

- For native authentication:
  - **External Authentication Provider:** In the drop-down menu, select **(Not Used)**.
  - **External Provider User Name:** Leave this field blank.
- For an external identity provider:
  - **External Authentication Provider:** In the drop-down menu, select the external identity provider.
  - **External Provider User Name:** Enter the user name configured in the external identity provider in this field.

See [About Managing Users and Groups](#).

If you are in a OneStream-hosted environment, see the *Identity and Access Management Guide* for information about authentication with OneStream IdentityServer.

For instructions on configuring authentication in a self-hosted environment, see the *Installation and Configuration Guide*.

# About Managing Users and Groups

See the following topics for help defining and managing users and groups:

- "Ways to add Users and Groups to an Application" below.
- "Creating and Managing Users" below.
- "Creating and Managing Groups" on page 856.

For information about how group-based assignment to system security roles determines a user's access to artifacts and capabilities, see "Manage System Security" on page 877.

## Ways to add Users and Groups to an Application

You can add users and groups to an application in three ways:

- Define them manually in System Security. See "Creating and Managing Groups" on page 856 and "Creating and Managing Users" below.
- Load them in a bulk import using an XML load file that contains user and group properties and parameters. We suggest building this file using sample security Excel template provided with the Sample Templates MarketPlace solution. See "Creating and Managing Users" below and "Creating and Managing Groups" on page 856.
- Use APIs. See "Managing Users and Groups Using BRApi Functions" on page 884.

## Creating and Managing Users

By default, only an Administrator can perform the tasks described in the following topics. An Administrator can enable other users to perform these tasks by giving them specific system security roles.

In this topic:

- "Tips and Best Practices" on the next page.
- Creating Users.
- Managing Users.
- Loading and Extracting Users.

## Requirements

To let non-Administrators create and manage users and groups, grant them the required security roles. This involves assigning the group to which non-Administrator belongs, to the required roles.

1. Click **System > Security > System Security Roles**.
2. Click **Edit** by these roles to assign the group in which the user is a member:
  - **ManageSystemSecurityUsers**: Assignees can create, modify and manage users.
  - **ManageSystemSecurityGroups**: Assignees can define, modify and manage groups.
  - **ManageSystemSecurityRoles**: Assignees can manage roles to provide group-based, functionally-tailored access to artifacts and tools.

System Security Roles		
ManageSystemDashboards	Administrators	...
ManageSystemDatabaseFiles	Administrators	...
ManageFileShare	Everyone	...
ManageFileShareContents	Administrators	...
AccessFileShareContents	Administrators	...
RetrieveFileShareContents	Administrators	...
EncryptSystemBusinessRules	Nobody	...
ViewAllLogonActivity	Administrators	...
ViewAllErrorLog	Administrators	...
ViewAllTaskActivity	Administrators	...
ManageSystemSecurityUsers	Administrators	...
ManageSystemSecurityGroups	Administrators	...
ManageSystemSecurityRoles	Administrators	...

To revoke the ability to manage users, groups and roles, remove assignees from the relevant role. See "Manage System Security" on page 877.

# Tips and Best Practices

- The Administrator is assigned to all system security roles (roles) so can always manage application and system-wide users, groups, artifacts, data, and tools. Removing or reassigning the Administrator group or assigning other users and groups to roles does not revoke the Administrator's access.
- The Administrator cannot be disabled and is unaffected by inactivity thresholds that disable users who try to log in after a specific period of time elapses.

# Creating Users

1. Click **System > Security > Users**.
2. Click **Create User**.
3. Enter a name and description.
4. From **User Type**, select the license type purchased that governs the user's access to artifacts and associated OneStream offerings:
  - **Interactive**: They can use all features and tools.
  - **View**: They can access data, reports, and dashboards in a production environment and associated database, but cannot load, calculate, consolidate, certify, or change data.
  - **Restricted**: They cannot use some MarketPlace Solutions features such as Lease, Account Reconciliation and more due to contractual limitations.
  - **Third Party Access**: They can access applications with a third-party application by logging in using a named account. They cannot change data, modify artifacts or access the Windows application or a browser-based application.
  - **Financial Close**: They can use Account Reconciliation and Transaction Matching MarketPlace solutions.
5. Set **Is Enabled** to **True** to activate the user. Select **False** to deactivate the user.

6. The information in **Status** will reflect the user's activity, such as their latest login. **Inactivity Threshold** displays the number of days a user can remain active in the system without logging in. The user receives an error if they try to log in after the specified number of days elapses. See "Creating Users" on the previous page.
7. Read "About User Authentication" below, then "Specify Authentication Settings" below.

## About User Authentication

You can add and authenticate users as:

- Native users that are managed locally in OneStream.
- External users referenced by an external identity provider (IdP).

## Specify Authentication Settings

1. From **External Authentication Provider**, indicate how to authenticate the user:
  - To use native authentication: Select **Not Used** and enter the user's password in **Internal Provider Password**. The first time the user logs in, they can change their password.
  - To use external authentication:
    - a. Select the appropriate external IdP from **External Authentication Provider**.
    - b. In **External Provider User Name**, enter the user name in the IdP. For example, if a user's name in Azure AD is Azure\_LHall@azure.com, enter Azure\_LHall@azure.com. This name must be unique and match the user name in the IdP.
2. "Specify Preferences and Group Membership" below.

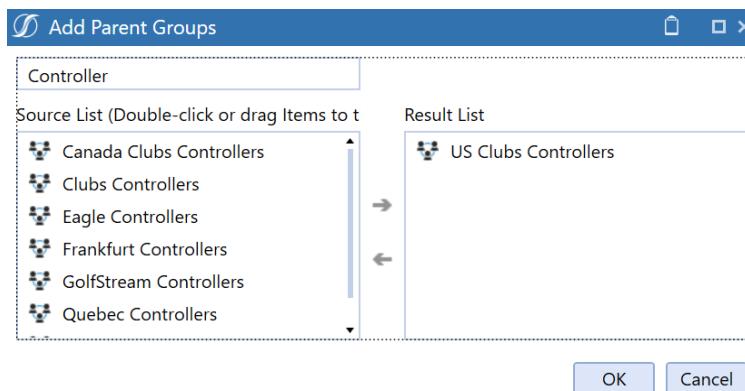
## Specify Preferences and Group Membership

1. In **Email** enter the email address with which the user can receive alerts and messages, such as those generated with business rules.
2. In **Culture** select the user's locale. Supported locals and languages are specified during OneStream server configuration. See "International Settings" on page 886.

3. In **Grid Rows Per Page** specify how many rows to display on grids before a page break. Consider the rate of connectivity and screen resolution.
4. Use **Custom Text** to personalize aspects of functionality given the user's responsibilities. For example, you could define a text field to:
  - Act as a metadata tag, limiting who the user can email.
  - Filter a distribution list or to provide text and images for the user's default workflow profile.
  - Launch a functionally-tailored view of the user's workspace, such as reporting for controller or executive.

See "Text 1-8" in "[Entity Dimension](#)" on page 337.

5. In **Group Membership**, click **Add Groups** to include the user in the groups that provide access to the features and tools that the user needs.



If the appropriate group does not exist, define it. See "[Creating and Managing Groups](#)" on page 856.

6. Click **OK**, **Save** and then **Load**.

When processing finishes, review the user list and user settings to ensure the loaded users are correctly defined.

## Managing Users

Perform the following tasks to edit, copy, delete and perform other management tasks. You can also manage users with API functions.

## About Managing Users and Groups

---

1. Ensure that you have the necessary security role. See "Managing Users" above.
2. Click **System > Security > Users**.
3. Select a user, then the action to take. For example:
  - **Delete Selected Item:** Permanently delete the user.
  - **Copy Selected Item:** Create a new user based on the selected user's settings. See "Copying Users" below.
  - **Show all parent groups for user:** See the groups in which the user is a member. Since group assignment to system security roles and user interface roles determines what a user can do, use this option to identify their access to artifacts.
4. Click **Save**.

## Copying Users

1. Click **System > Security > Users**.
2. Select the user, then click **Copy Selected Item**.
3. In **New Name**, enter a user name.
4. Select **Copy References made by parent groups** to add the user to the original user's groups except exclusion groups.
5. Click **OK** and modify other settings as needed. See "Managing Users" above.
6. Click **Save**.

## Disabling Users

Administrators can disable users, preventing them from logging in by:

- Modifying a user's account to set **Is Enabled** to **False**. See "Disabling a User Manually" below.
- Specifying an inactivity threshold to deactivate all users - native and external - that try to log in after a particular amount of time elapses. This threshold applies to users in the

## About Managing Users and Groups

---

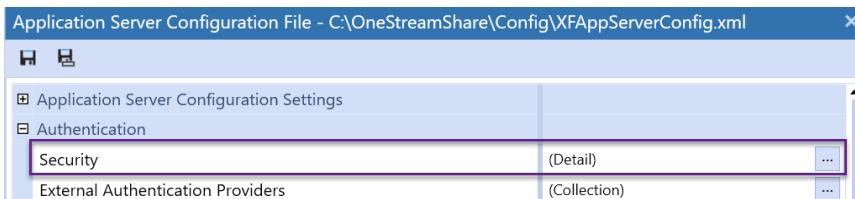
Administrator group, but not the Administrator. Users that try to log in after the threshold expires receive an error. See "Defining an Inactivity Threshold" below.

### Disabling a User Manually

1. Click **System > Security**.
2. Expand **Users** and select the user to disable.
3. From **Is Enabled**, select **False**.
4. Click **Save**.

### Defining an Inactivity Threshold

1. Launch the Server Configuration Utility.
2. Select **File > Open Application Server Configuration File** and browse to: **OneStreamShare\Config\XFAppServerConfig.xml**.
3. In **Authentication**, click **Security**.



4. In **Logon Inactivity Threshold**, enter the number of days after which inactive users who try to log in, receive an error.
5. Click **OK** then **Save**.
6. In a command prompt, reset IIS.
7. In the application, select **System > Security > Users**.
8. Select a user and review their **Logon Inactivity Threshold**.

## Enabling Users

1. Select **System > Security**.
2. Expand **Users** and select the user.
3. In **General**, set **Is Enabled** to **True**.
4. Click **Save**.

## Removing Users From Groups

Because group membership determines access to artifacts and tools, be sure that you want to remove a user from a group.

1. Select **System > Security**.
2. Expand **Users** then click the user.
3. In **Group Membership**, select the group and click **Remove Selected Item**.
4. Click **Save**.

## Managing Users With BRApi

You can perform some user and group management tasks using BRApi functions such as CopyUser, DeleteUser, and CopyGroup. The ability to use these functions is governed by your role as an Administrator and may require a Manage System Security role. See "Manage System Security" on page 877.

For example, if a dashboard's design includes a button that uses a BRApi function to support adding users, users that click the button are validated to ensure they are in the Administrator group or associated with the **ManageSystemSecurityUsers** role.

## Loading and Extracting Users

You can bulk load user accounts, groups, and security instead of defining them manually. You can load these artifacts in two ways:

- Use the **SecurityTemplate.xlsx** file provided by the SampleTemplates MarketPlace solution to generate an xml load file. We recommend this option. See the *OneStream Sample Templates User Guide*.
- Extract users in the sample GolfStream application, modify the xml export file to specify user and group properties, and then load the file. See "Loading Users" below.

Administrators must load users and groups into a new application or one without existing users and groups. If user accounts exist, loaded users are validated by comparing their current security settings to those in the xml load file.

## Requirements

By default, only Administrators can load and extract users. To let other users load and extract users, grant them these system security roles:

- SystemLoadandExtractPage System. This is a User Interface role.
- ManageSystemSecurityUsers
- ManageSystemSecurityGroups
- ManageSystemSecurityRoles

## Loading Users

1. Click **System > Tools > Load/Extract**.
2. Select **Load** and browse to an xml load file.
3. Click **Load**.
4. When processing finishes, review the user list and user settings to ensure the loaded users are correctly defined.

## Extracting Users

You can extract users, groups, and security roles to an xml export file.

1. Click **System > Tools > Load/Extract**.
2. Click **Extract** then select **File Type > Security**.

3. In **Items to Extract**, select **Users**. Select **Export Unique IDs** to also extract each user's ID.
4. Perform a task:
  - Click **Extract** to specify where to save the extract file.
  - Click **Extract and Edit** to view and modify the data in an XML Editor.

# Creating and Managing Groups

By default, only an Administrator can perform the tasks described in the following topics. An Administrator can enable other users to perform these tasks by assigning specific system security roles. See "Requirements" below.

In this topic:

- "About Groups and Inherited Security" on the next page.
- "About Exclusion Groups" on page 858.
- "Tips and Best Practices" on page 858.
- Creating Groups.
- "Creating and Managing Groups" above.
- "Creating and Managing Groups" above.
- "Creating and Managing Groups" above.

## Requirements

To let non-Administrators create and manage users and groups, grant them group-based access to the required security roles.

1. Click **System > Security > System Security Roles**.
2. Click **Edit** by these roles to assign the group in which the user is a member:

## About Managing Users and Groups

---

- **ManageSystemSecurityUsers:** Assignees can create, modify and manage users.
- **ManageSystemSecurityGroups:** Assignees can define, modify and manage groups.
- **ManageSystemSecurityRoles:** Assignees can manage roles to provide group-based, functionally-tailored access to artifacts and tools.

System Security Roles	
ManageSystemDashboards	Administrators
ManageSystemDatabaseFiles	Administrators
ManageFileShare	Everyone
ManageFileShareContents	Administrators
AccessFileShareContents	Administrators
RetrieveFileShareContents	Administrators
EncryptSystemBusinessRules	Nobody
ViewAllLogonActivity	Administrators
ViewAllErrorLog	Administrators
ViewAllTaskActivity	Administrators
ManageSystemSecurityUsers	Administrators
ManageSystemSecurityGroups	Administrators
ManageSystemSecurityRoles	Administrators

To revoke the ability to manage users, groups and roles, remove assignees from the relevant role. See "Manage System Security" on page 877.

## About Groups and Inherited Security

You cannot assign individual users access to tools and artifacts. System security roles (roles) to which you assign groups determine this access. Create groups to grant large numbers of users or other groups the functionally-tailored access that they require.

You can define nested, hierarchical groups to best suit your organizational entities, workflows and reporting structures. Nested groups contain lower-level child groups. In this case, child groups - and the users that they contain - inherit the access defined for the parent group.

**NOTE:** Groups cannot be externally authenticated.

# About Exclusion Groups

Use exclusion groups to grant almost everyone - except a particular group or a small number of users - access to a tool or artifact. For example, if everyone but Jim Fey and Lee Diaz must create dashboards, define an exclusion group in which Jim and Lee are members set to "Deny Access". All other members are set to "Allow Access". To easily grant access to dashboards, assign the exclusion to which gives everyone access except Lee and Jim access.

Because groups providing access to data and artifacts may contain many groups, removing just a few users - to reflect a corporate reorganization for example, from the group hierarchy can be time consuming. To handle the reorganization, create an exclusion group with the users involved and apply it to the roles they no longer need.

## Tips and Best Practices

- The Administrator is assigned to all roles, so they can always manage artifacts, data and tools. Assigning other groups to security roles does not revoke the Administrator's access.
- Child groups, nested in higher-level parent groups, can access the tools and artifact that the parent group can, given the parent group's assignment to system security roles (roles). This access is inherited in group hierarchies, from a parent level downward to child groups.
- Removing child groups from parent groups revokes access to the tools and artifacts that the parent group provides, based on its assignments to roles.
- In an exclusion group, access to artifacts is determined based on the exclusion order you specify, regardless of a user's membership in a group. To ensure that users who are in several groups can not access artifacts but everyone else can, put:
  - The groups to which the users belong at the top, set to "Allow Access".
  - The individual users below the groups and set the users to "Deny Access".

## Creating Groups

1. If you are not an Administrator, be sure you meet the "Creating Groups" above.
2. Click **System > Security**.
3. Click **Groups** then **Create Group**.

4. Enter an intuitive name and description.
5. In **Group Membership** specify the users or child groups to add to the group you are defining, or select a parent group to which to add your new group. Perform any task:
  - To include users and groups in the new group, click **Add User | Add Child Group** and selecting the user or group. Members can access the artifacts and tools for each system security role to which the group is assigned.
  - To revoke group membership, select users and groups and click **Remove**. This revokes their access to the tools and artifacts group membership provides based on role assignments.
  - To nest the group in a parent group, click **Add Groups in Parent Groups That Contain This Group** and select a higher level group. Members in this parent group inherit its access and permissions based on role assignments.
6. Click **Save**.

## Creating Exclusion Groups

Define an exclusion group to grant almost everyone - except a particular group or a small number of users - group access to a tool or artifact.

1. Click **System > Administration > Security**.
2. Click **Groups** then **Create Exclusion Group**.
3. Enter an intuitive name and description that indicates who is being restricted by this group. For example, to omit a department, consider Everyone-But-<Department name>.
4. In **Group Membership** click **Add Child Groups** or **Add Users** to specify who to include in the group.
5. To prevent members accessing artifacts to which the group has role-based access, set particular users or groups to **Deny Access**. Otherwise, set members to **Allow Access**.
6. Use the arrows to order the exclusions carefully, because access is granted - regardless of a user's membership in a group, based on the order you specify.

## About Managing Users and Groups

---

For example, if Amelia and Bob are in the Frankfurt Controller group, the order below does not restrict them from artifacts, even though they are listed first, because they are in the Frankfurt Controller group.

The screenshot shows a configuration interface for a group. On the left, there's a sidebar with 'General' and 'Group Membership' sections. Under 'Group Membership', there's a list of users and groups: 'Amelia' (Deny Access), 'Bob' (Deny Access), and 'Frankfurt Controller' (Allow Access). A purple box highlights the 'Allow Access' dropdown for the group. To the right of the list are up and down arrow buttons for reordering.

To ensure Amelia and Bob can not access artifacts, put Frankfurt Controller first, set to **Allow Access**. Put Amelia and Bob below the group, set to **Deny Access**.

7. Click **Save**.

## Managing Groups

1. If you are not an Administrator, meet the "Managing Groups" above.
2. Click **System > Security**.
3. Select **Groups or Exclusion Groups**, then select a group.
4. Click to perform any task:
  - **Delete Selected Item:** Permanently delete a group.
  - **Rename Selected Item:** Change a group's name.
  - **Copy Selected Item:** Create a group based on the existing group's settings. Select **Copy child user and group references** to add the same users or groups to the new group.
  - **Show All Users:** View the users in the group.
  - **Show All Groups:** View the lower level groups that are nested in the group.

5. In **Group Membership**, change who is in the current group and modify the parent groups associated with the current group. Perform any task:
  - To withdraw users and groups, select them and click **Remove**. This revokes their access to artifacts and tools based on the system security roles (roles) to which the group is assigned.
  - To add users and groups, click **Add User | Add Child Group**. This grants them access to the artifacts and tools the group provides based on it's role assignments.
  - To withdraw the current group from a parent group, select the parent and click **Remove**. This revokes access to the artifacts and tools provided by the parent group based on it's role assignments.
  - To include the current group in another parent group, click **Add Groups** and select the group.

## Managing Group Membership

You can add groups and users to, and remove them from, other groups. This may created a hierarchy or impact system security role-based access to artifacts and tools. See "Managing Groups" on the previous page.

1. Click **System > Security**.
2. Expand **Groups** and select the group.
3. In **Group Membership**, perform any task:
  - To include a user or group, click **Add User** or **Add Group** and select the user or group to add.
  - To withdraw the current group from a parent group, select the parent and click **Remove**. This revokes access to the artifacts and tools provided by the parent group based on it's role assignments.
  - To include the current group in a parent group, click **Add Groups** and select the group.

## Loading and Extracting Groups

You can bulk load groups, users and security roles instead of defining them manually. You can load groups in two ways:

- Use the **SecurityTemplate.xlsx** file provided by the SampleTemplates MarketPlace solution to generate an xml load file. We recommend this option. See the *OneStream Sample Templates User Guide*.
- Extract the groups in the sample GolfStream application, modify the xml export file to specify your group properties, and then load the file. See "Loading Groups" below.

Administrators must perform the load for a new or empty application that does not contain users or groups. To enable other users to load and extract groups, assign them the necessary security roles. See "Creating and Managing Users" on page 847.

## Requirements

To enable other users to load and extract groups, Administrators must assign the group in which these users are members, access to these roles:

- SystemLoadandExtractPage (user interface role).
- ManageSystemSecurityGroups
- ManageSystemSecurityUsers
- ManageSystemSecurityRoles

See "Manage System Security" on page 877.

## Loading Groups

1. If you are not an Administrator, see "Requirements" above.
2. Click **System > Tools > Load/Extract**.
3. Select **Load** and browse to an XML load file.
4. Click **Load**.
5. When processing finishes, review the **Groups** list and individual group settings to ensure the loaded groups are correctly defined.

## Extracting Groups

1. If you are not an Administrator, see "Requirements" on the previous page.
2. Click **System > Tools > Load/Extract**.
3. Click **Extract** then select **File Type > Security**.
4. In **Items to Extract**, select **Security Groups**.
5. Perform a task:
  - Click **Extract** to specify where to save the extract file.
  - Click **Extract and Edit** to modify the extracted data in an XML Editor.

# System Security Roles

The following roles apply to the entire system, not just one Application.

**Manage System Dashboards** - Use to manage all System Dashboards regardless of access to certain Dashboards. This role links the SystemPane Role and the System User Interface Roles section, meaning this security role must include this group in order to be active.

**Manage System Database Files** - There are two file systems in the Framework database (such as the System database) and each Application database. Users with the ManageSystemDatabaseFiles and ManageApplicationDatabaseFiles roles have read and write access to all folders and files in these file systems, respectively. Other users can be given read and/or write access to specific folders in the database file systems using the a folder's security settings. This option ties the FileExplorerPage role from the System User Interface Roles section, meaning this security role must include this group in order to be active.

**Manage File Share** - The File Share is a Windows folder that Application Servers can read and write. It is configured in the XFAppServerConfig.xml file using the FileShareRootFolder setting. The File Share is a server-side storage area where external systems or IT can stage and upload ifles. Users with the ManageFileShare role can edit these folder and files using File Explorer. This option ties the FileExplorerPage role from the System User Interface Roles section, meaning this security role must include this group in order to be active.

## Application Server Configurations

Application server configurations can now be made by Administrators and advanced IT persona. Common requests that resulted in support calls can now be addressed internally by advanced OneStream users with the proper access. This process also eliminates the need for an IIS restart for the changes to take place. Understand that as settings changed and saved, they are automatically applied every two minutes.

There are two system roles that manage six categories of configurations. The application maintains an audit trail so changes to the configurations are accessible. Upon install, the application server configurations are enabled by default. To mitigate misuse of configurations settings, Customer Support can disable full features, sections, and property changes via XML/App config.

# System Roles

Two roles exist to grant Administrators and advanced IT users the ability to change server configurations:

### System Security Role: ManageSystemConfiguration

- Default to Nobody
- Not automatically granted to Administrator

### System User Interface Role: SystemConfigurationPage

- Default to Administrator Group - Administrators will have read-only rights
- System User Interface Role should be “view only” to all users assigned

# System Configurations

There are six categories of system configurations:

- General
- Environment
- Memory
- Multithreading
- Recycling
- Database Server Connections

**NOTE:** Memory and Multithreading system configurations must be enabled. Customers should contact support to have this configuration change made for them.

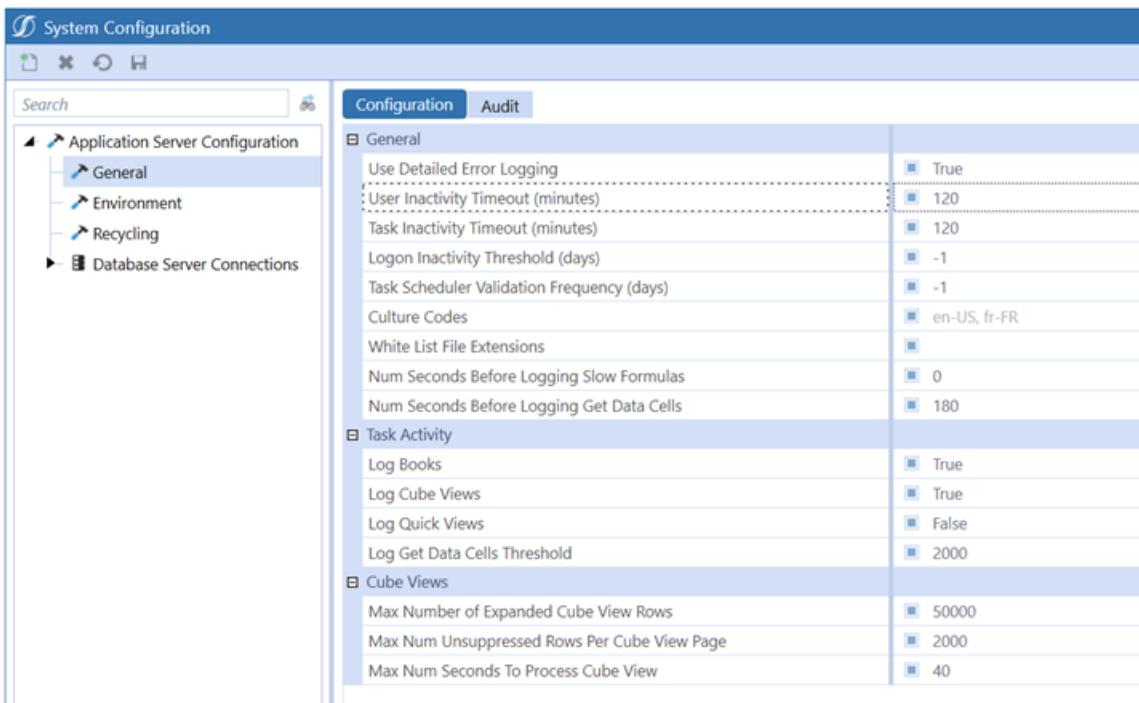
An IIS restart is not required to capture changes as they are automatically applied every two minutes. Saved changes are tracked via the Audit tab, which is available for each server configuration.

When enabled, permitted users can adjust system configurations in app by navigating to System>Administration>System Configuration.

# General System Configurations

When the box and setting are greyed out, the property is set based on the configuration file. Deselect the box to adjust your custom value then save. This overrides the configuration file. Select the property's box again to default back to the configuration file.

**NOTE:** Improperly modifying certain server configurations can impact overall environment or application performance. Consult OneStream support for recommendations.



## General Menu

**Use Detailed Error Logging:** When true, stack trace information is shown. When false, stack trace information is not shown.

**User Inactivity Timeout (minutes):** Set the number of minutes a user is timed out due to inactivity.

**Task Inactivity Timeout (minutes):** Set the number of minutes a task is timed out due to inactivity.

## System Security Roles

---

**Logon Inactivity Threshold (days):** Set the Logon Inactivity Threshold (days) to the number of days of inactivity before the user can no longer access the system. Set to -1 to disable the setting.

**Task Scheduler Validation Frequency (days):** Set the number of days in which the Task Scheduler validation runs.

**Culture Codes:** Set the appropriate code for display settings using the standard Microsoft Local designation for each language. (ex. en-US)

**White List File Extensions:** When blank, any file type can be saved to a root folder then uploaded. Add custom file types by clicking the box then the ellipses to restrict the types of files which will be supported in the File Explorer.

**Num Seconds Before Logging Slow Formulas:** Set the number of seconds before slow formulas are logged. This will enable logging of formulas and impact consolidation performance. Disable when no longer required.

**Number Seconds Before Logging Get Data Cells:** Set the number of seconds before Get Data Cells are logged. Default is 180 and should only be increased for debug purposes

## Task Activity Menu

**Log Books:** When set to True (default), a log is created in Task Activity when the items are included as Task Activity steps for that specific book. The intention of this feature is to verify entries in the Task Activity grid and the settings in the configuration file work as expected.

**Log Cube Views:** When set to True, a log is created in Task Activity when a Cube View is opened, a report is run or an export to Excel is completed in the data explorer. The intention of this feature is to analyze data analysis performance.

**Log Quick Views:** When set to True, a log is created in Task Activity when a new Quick View is created or when rows/columns are shifted/moved around. The intention of this feature is to analyze data analysis performance.

**Log Get Data Cells Threshold:** This logs the calls to GetDataCells and GetDataCellsUsingScript. It includes context information such as the Excel file name or the Cube View name. It only creates logs if the number of Data Cells being requested is equal to or greater than the value provided in this field.

## Cube Views Menu

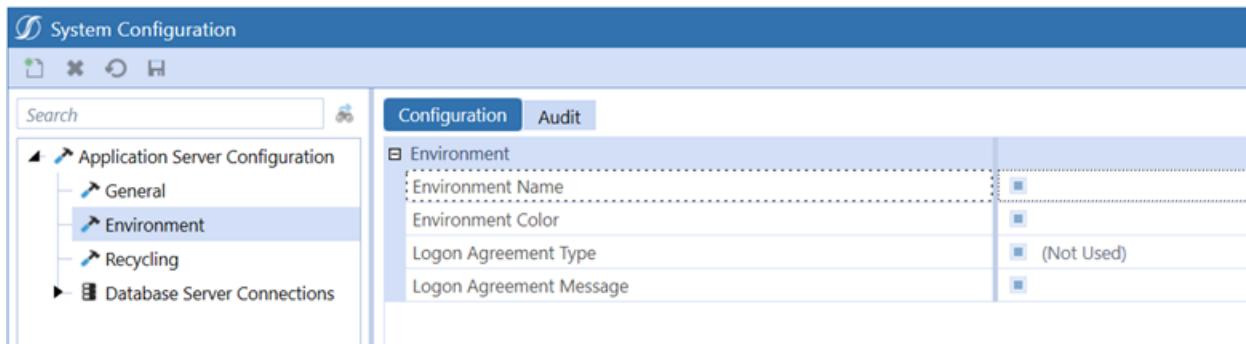
The Cube View configurations will impact all Cube Views in an application environment. They are designed to optimize the Cube View performance through managing the page size and initiating paging which will maximize the availability of server resources. These settings can be overridden to tailor the design and performance of a specific Cube View, using the General Settings/Common/Paging properties found on each Cube View.

**Max Number of Expanded Cube View Rows:** Set the max number of rows displayed when using an expanded Cube View.

**Max Number of Unsuppressed Row Per Cube View Page:** The default value of 2000 is used, which is determined by the settings on the OneStream Server Configuration Utility. The maximum value is 100,000 rows. If the Cube View performs well, but you want 2500 rows to display, for example you may want something in the tree to display in the first page, then you would increase the rows.

**Max Number Seconds To Process Cube View:** This setting impacts paging behavior. The default value of 20 will be used which is determined by the OneStream Server Configuration Utility. The maximum value is 600 seconds.

## Environment System Configurations



## Environment

The configurations allow the Administrator to tailor the environment login process for the user by providing a custom label or by triggering an acceptance criteria upon each login.

**Environment Name:** Enter the name to be displayed (in white) for the environment. You can enter up to 150 characters.

## System Security Roles

---

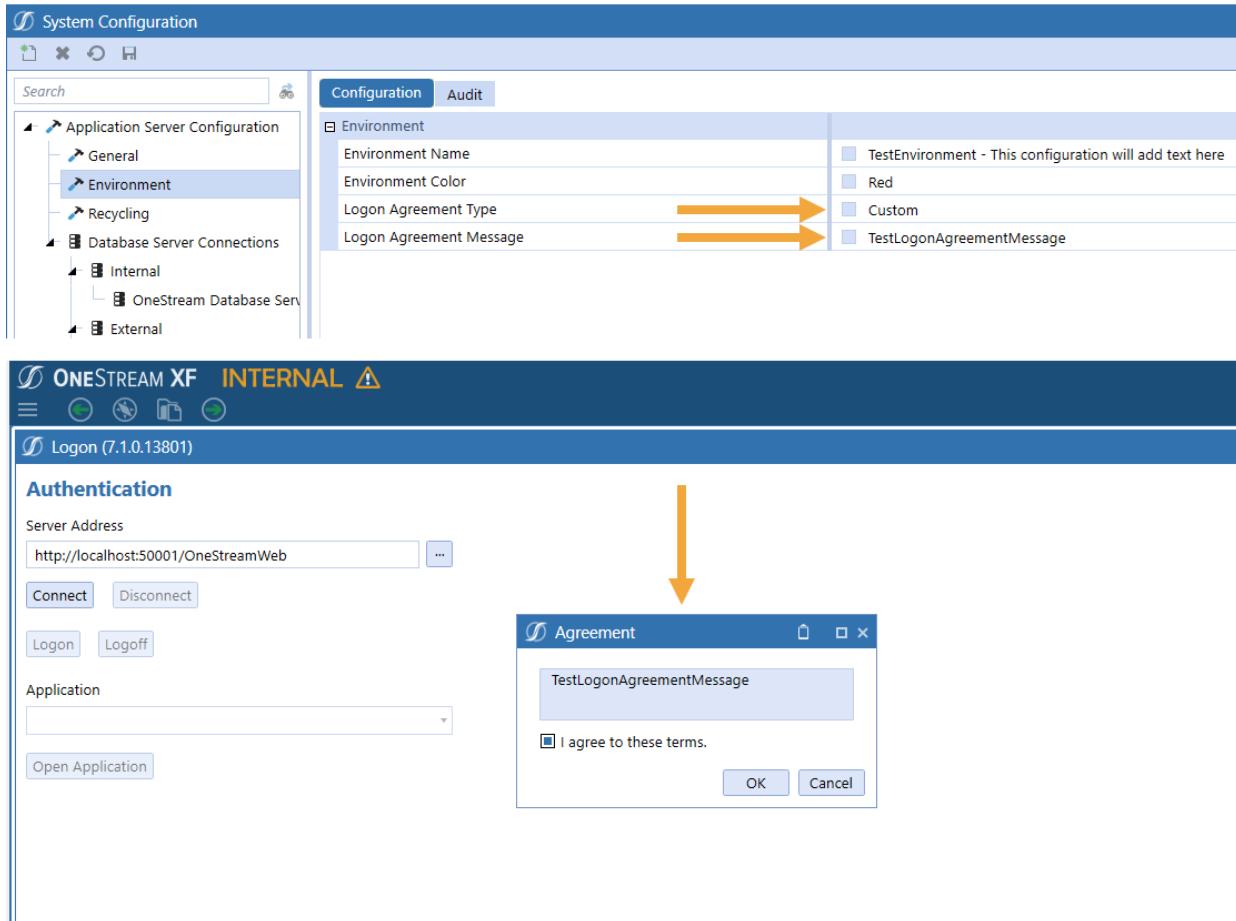
**Environment Color:** Specify a provided environment color or enter a hex value to display the name on a colored background.

The screenshot displays two windows side-by-side. The left window is titled 'System Configuration' and shows a tree view of configurations under 'Application Server Configuration'. The 'Environment' node is selected. The right window is titled 'Configuration' and shows the 'Environment' section with four fields: 'Environment Name', 'Environment Color', 'Logon Agreement Type', and 'Logon Agreement Message'. A red arrow points from the 'Environment Color' field to a color swatch labeled 'Red'. Another red arrow points from the 'Logon Agreement Message' field to a red box containing the text 'TestEnvironment - This configuration will add text here'. Below these windows is another window titled 'Logon (7.1.0.13801)' which shows a 'User Name' field containing 'Administrator'. A large orange arrow points upwards from the 'Logon' window towards the 'Logon Agreement Message' field in the configuration window.

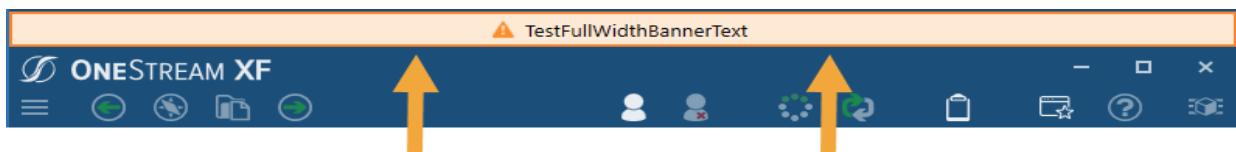
**Logon Agreement Type:** To display a specific message after a user logs on, select Custom and enter the message text.

**Logon Agreement Message:** Enter the message text.

## System Security Roles



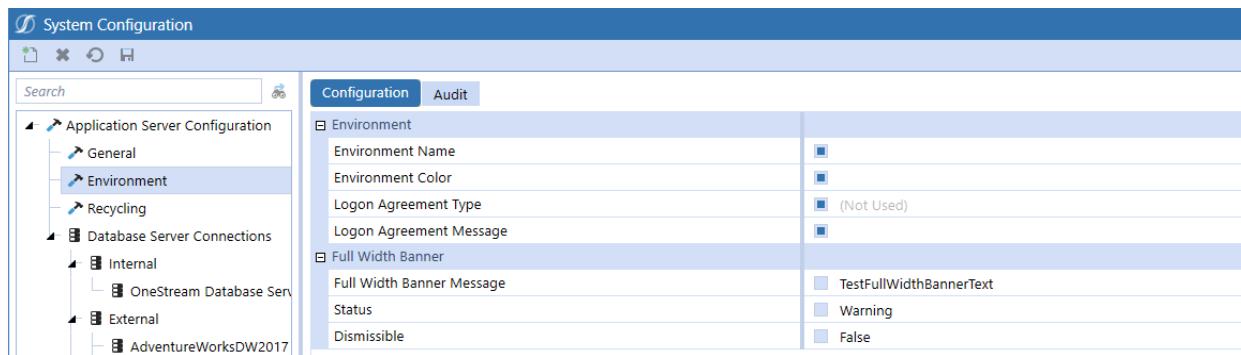
## Full Width Banner



The configurations allow you to display information across the top banner indicating varying levels of severity. The banner can be permanently displayed or dismissible.

## System Security Roles

---



**Full Width Banner Message:** Enter the banner message text to be displayed.

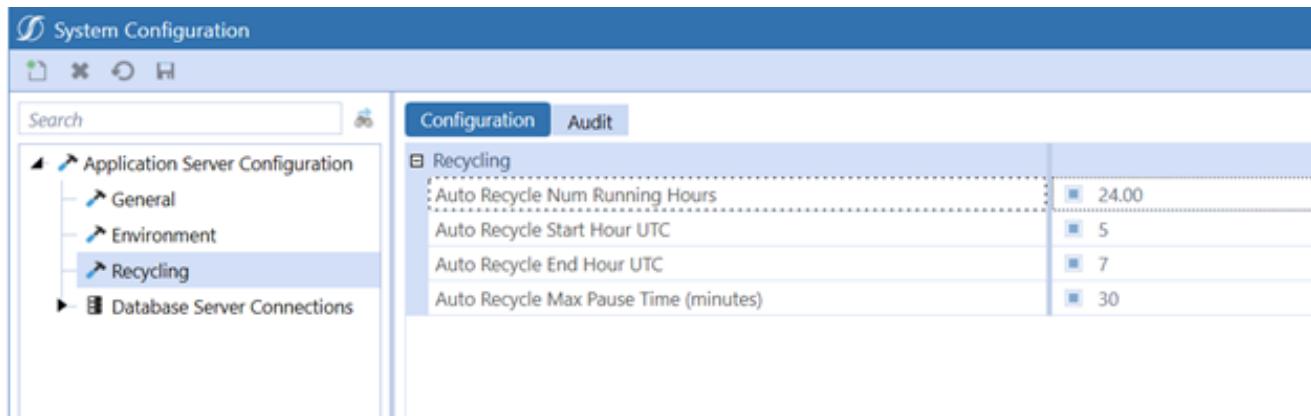
**Status:** Indicate the severity of the message by selecting Informational, Warning, Successful, or Critical from the drop-down menu.

**Dismissible:** When True, the banner may be dismissed after log in. When False, the banner cannot be dismissed.

Type	Sample
Informational	Informational Banner Sample Text
Warning	Warning Banner Sample Text
Successful	Successful Banner Sample Text
Critical	Critical Banner Sample Text

# Recycling System Configurations

Overtime, server memory may become increasing fragmented, which can affect performance and stability. The default configuration of a daily recycling process is standard. The System Configuration page allows the administrator to tailor when these events occur to be best suited for their implementation.



## Recycling Menu

**Auto Recycle Num Running Hours:** Default is 24, which means once a day, the server will recycle. Automatic Recycling allows Application Servers a chance to recycle, which is a recommended practice. These first four settings control this behavior.

**Auto Recycle Start Hour UTC:** Default is 5, which means 05:00 UTC time. This is the earliest time in a day when a server can automatically recycle. It is best to set this and the End Hour to be a range of time with the lowest amount of Application Server activity.

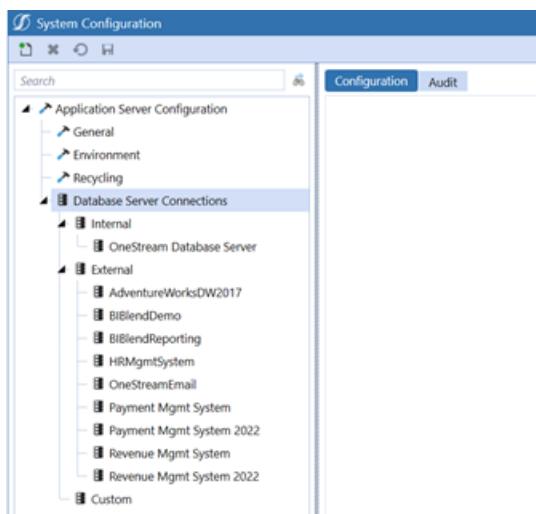
**Auto Recycle End Hour UTC:** Default is 7, which means 07:00 UTC time. This is the latest time in a day when a server can automatically recycle.

**Auto Recycle Max Pause Time (minutes):** Default is 30. This means that when it is time to recycle a server automatically, it will first pause from accepting more server tasks, but allow for existing assigned tasks to complete processing for 30 minutes before recycling. If there are no active tasks for this server, it will recycle when the time comes.

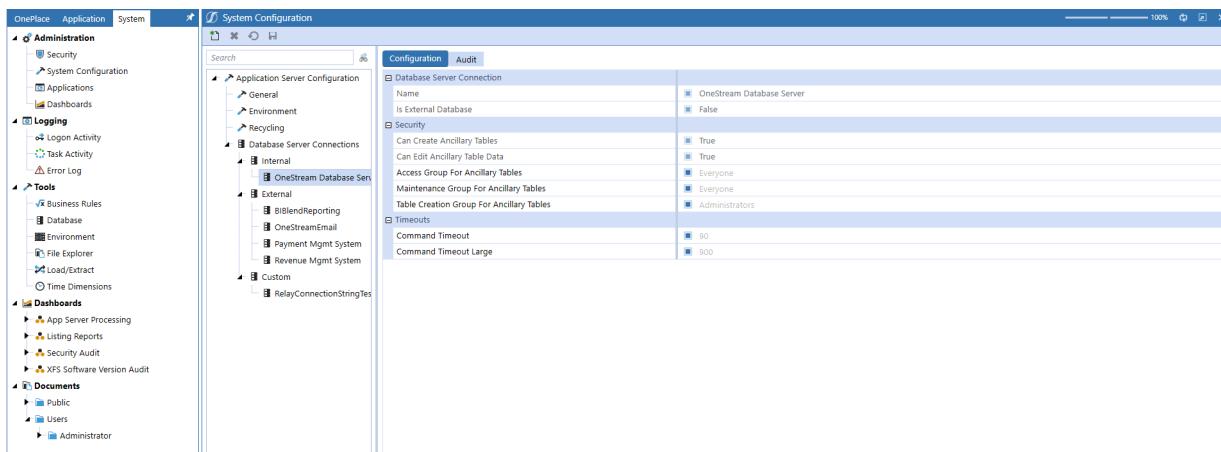
# Database System Configurations

The Database Server Connections provides the Administrator with greater visibility into the structure of their environment. Additionally, the administrator can quickly and easily establish database connections to support their model's design requirements.

An IIS restart is not required to capture changes as they are automatically applied every two minutes. Saved changes are tracked via the Audit tab, which is available for each server configuration



## Database Server Connections

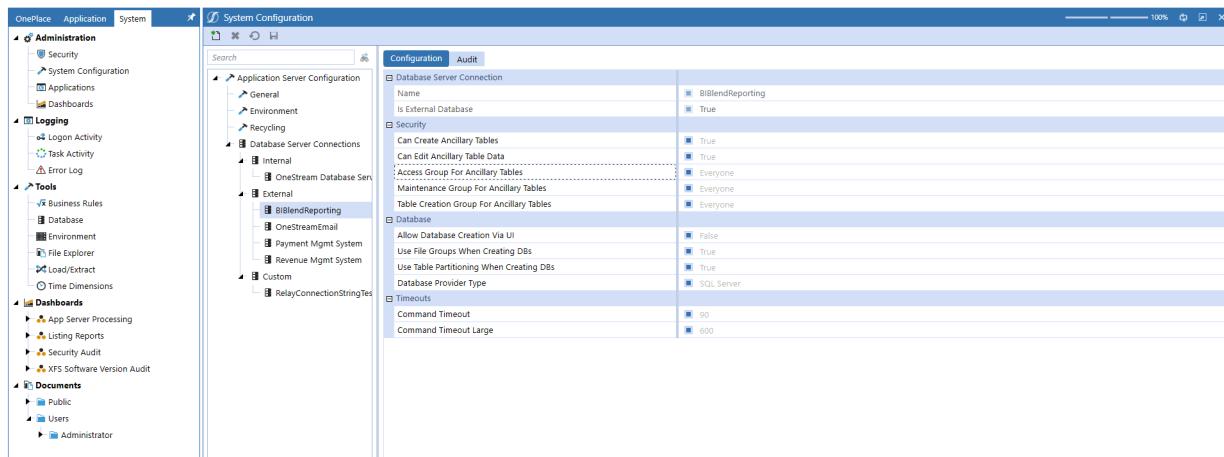


## System Security Roles

---

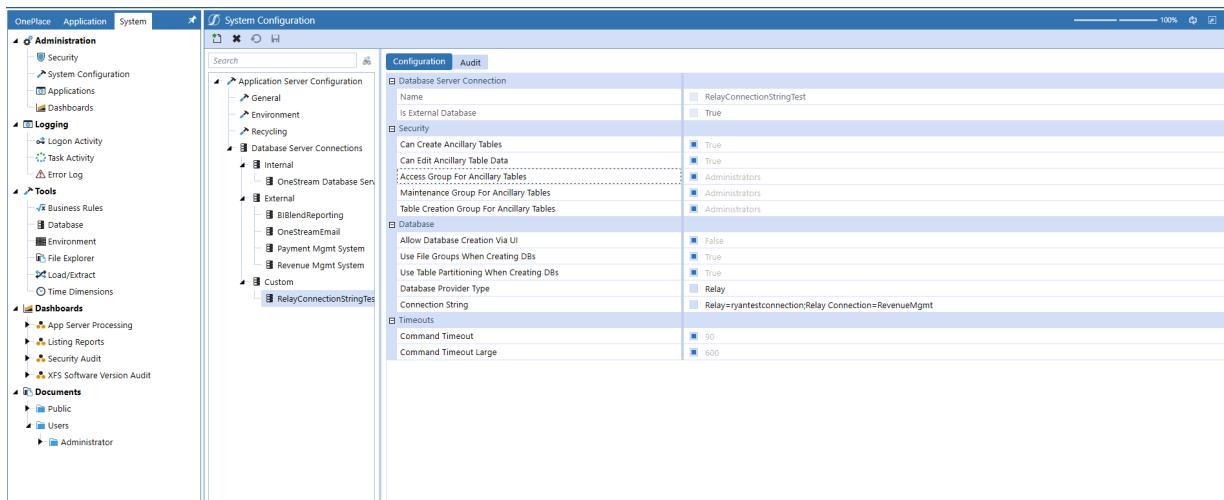
### Internal Menu

Configure Database Server Connection, Security, and Timeouts settings from this menu.



### External Menu

Configure Database Server Connection, Security, Database, and Timeouts settings from this menu.



### Custom

Configure Database Server Connection, Security, Database, and Timeouts settings from this menu.

## **System Security Roles**

---

External only and the Connection String will be visible without displaying password and Customer can Edit.

- Connection String will allow for Edits
- Connection String will display “?” where password is used
- Connection String can be modified without changing password and saved with “?” not overriding original password
- Connection String password can be modified and upon save will display “?” in UI
  - Connections Strings using the following will be masked in the UI: pwd; pass; password
  - All others will display password and need to be addressed separately – majority of connection strings password will be handled above

Pre-existing External Connections, when moving over to Customer Connections, should use the previous name to avoid breaking any connections with Task Scheduler, Business Rules, etc.

## **Audits**

Configuration setting changes are tracked via the Audit tab. View changes by navigating to **System > System Configuration > Audit**.

## System Security Roles

The screenshot shows the 'System Configuration' interface with the 'Audit' tab selected. The left sidebar lists 'Application Server Configuration' sections such as General, Environment, Recycling, Database Server Connections (Internal, External), OneStream Database Services, and Custom. The main area displays a table of audit logs. The table has columns: Category, Name 1, Instance, Name 2, Use Default From, Use Default To, Value From, and Value To. Rows include DatabaseServerConnection properties like CanCreateAncillaryTables, CanEditAncillaryTableData, CommandTimeout, ConnectionString, DbProviderType, MaintenanceGroupForAncillaryTables, TableCreationGroupForAncillaryTables, UseFileGroupsWhenCreatingDBs, UseTablePartitioningWhenCreatingDBs, AllowDBCreationViaUI, CanCreateAncillaryTables, CanEditAncillaryTableData, CommandTimeout, CommandTimeoutLarge, RelayConnectionString, and more. Other rows show Environment properties like EnvironmentColor, EnvironmentName, LogonAgreementMessage, and LogonAgreementType.

System Database Tables audits are viewed by navigating to **System > Tools > Database > SystemConfig**.

The screenshot shows the OneStream XF INTERNAL interface. The left sidebar includes sections for Administration (Security, System Configuration, Applications, Dashboards), Logging (Log Activity, Task Activity, Error Log), Tools (Business Rules, Database, File Export, Load Extract, Time Dimensions), Dashboards (App Server Processing, Listing Reports, Security Audit, XFS Software Version Audit), and Documents (Public, Users, Admin). The 'Database' section is highlighted with a yellow box. The main area displays a detailed audit log table. The table columns are: Tier, Category, PropertyName1, InstanceName, PropertyName2, Timestamp, Username, and UseDefault. The log entries show various database connection properties being audited, such as DatabaseServerConnection properties like AccessGroupForAncillaryTables, AllowDBCreationViaUI, AzureElasticPoolMaxCpuSetting, AzureElasticPoolMinCpuSetting, AzureElasticPoolName, AzureResourceGroup, AzureSqlObjective, AzureSqlScalingType, AzureSqlServerName, AzureSqlStorageMaxSize, AzureSqlThumbnailRuleName, CanCreateAncillaryTable, CanEditAncillaryTableData, CommandTimeout, CommandTimeoutLarge, ConnectionString, DbProviderType, IsConnectionStringEncrypted, IsExternalDatabase, MaintenanceGroupForAncillaryTables, Name, TableCreationGroupForAncillaryTables, UseFileGroupsWhenCreatingDBs, UseTablePartitioningWhenCreatingDBs, and CultureCodes, all recorded by the user 'Test' at 3/9/2022 6:40:15 PM.

## System Security Roles

---

The Environment will show what settings the server is currently using. The server updates every 2 minutes. Here the user can see what settings are being picked up. The “Last Updated” column shows the date and time settings were changed. Regardless of what type of connection (Internal, External, Custom) any available configurations that can be changed will show up here.

Environment audits are viewed by navigating to **System > Tools > Environment > Application Server Sets**

The screenshot shows the 'Environment' application interface. On the left is a navigation tree with sections like Monitoring, Web Servers, Mobile Web Servers, Web To App Connections, Application Server Sets (with a Standard node), and Database Servers (with OneStream Database Server expanded to show various database instances). On the right is a table titled 'Behaviors' with columns: Property, Item, Value, Description, and Last Updated. The table lists numerous properties for 'BBlendReporting' and 'Custom Connection' objects, such as AccessGroupForAncillary, AllowDBCreationViaUI, AzureElasticPoolMaxDTU, AzureElasticPoolMinDTU, AzureElasticPoolName, AzureServiceLevelObject, AzureSQLEdition, AzureSQLScalingType, AzureSQLStorageMaxSiz, CanCreateAncillaryTable, CanEditAncillaryTable, CommandTimeoutLarge, DbProviderType, MaintenanceGroupForAll, Name, TableCreationGroupForAll, UseTablePartitioningWhen, ConfigurationFolder, and more. Most values are set to 'Everyone' or 'False', with some specific values like 1600 or 200 for DTU limits.

Property	Item	Value	Description	Last Updated
BBlendReporting	AccessGroupForAncillary	Everyone		6/2/2022 2:13:01 PM
BBlendReporting	AllowDBCreationViaUI	False		6/2/2022 2:13:01 PM
BBlendReporting	AzureElasticPoolMaxDTU	1600		6/2/2022 2:13:01 PM
BBlendReporting	AzureElasticPoolMinDTU	200		6/2/2022 2:13:01 PM
BBlendReporting	AzureElasticPoolName			6/2/2022 2:13:01 PM
BBlendReporting	AzureServiceLevelObject			6/2/2022 2:13:01 PM
BBlendReporting	AzureSQLEdition			6/2/2022 2:13:01 PM
BBlendReporting	AzureSQLScalingType	NotUsed		6/2/2022 2:13:01 PM
BBlendReporting	AzureSQLStorageMaxSiz	0		6/2/2022 2:13:01 PM
BBlendReporting	AzureSQLSystemBusiness			6/2/2022 2:13:01 PM
BBlendReporting	CanCreateAncillaryTable	True		6/2/2022 2:13:01 PM
BBlendReporting	CanEditAncillaryTable	True		6/2/2022 2:13:01 PM
BBlendReporting	CommandTimeoutLarge	600		6/2/2022 2:13:01 PM
BBlendReporting	DbProviderType	SqlServer		6/2/2022 2:13:01 PM
BBlendReporting	MaintenanceGroupForAll	Everyone		6/2/2022 2:13:01 PM
BBlendReporting	Name	BBlendReporting		6/2/2022 2:13:01 PM
BBlendReporting	TableCreationGroupForAll	Everyone		6/2/2022 2:13:01 PM
BBlendReporting	UseTablePartitioningWhen	True		6/2/2022 2:13:01 PM
BusinessRulesAssemblyFolder				6/2/2022 2:13:01 PM
ConfigurationFolder		C:\OneStreamShare		6/2/2022 2:13:01 PM
Custom Connection	AccessGroupForAncillary	Administrators		5/13/2022 4:06:47 PM
Custom Connection	AllowDBCreationViaUI	False		5/13/2022 4:06:47 PM
Custom Connection	AzureElasticPoolMaxDTU	1600		5/13/2022 4:06:47 PM
Custom Connection	AzureElasticPoolMinDTU	200		5/13/2022 4:06:47 PM
Custom Connection	AzureElasticPoolName			5/13/2022 4:06:47 PM
Custom Connection	AzureServiceLevelObject			5/13/2022 4:06:47 PM
Custom Connection	AzureSQLEdition			5/13/2022 4:06:47 PM
Custom Connection	AzureSQLScalingType	NotUsed		5/13/2022 4:06:47 PM
Custom Connection	AzureSQLStorageMaxSiz	0		5/13/2022 4:06:47 PM
Custom Connection	AzureSQLSystemBusiness			5/13/2022 4:06:47 PM
Custom Connection	CanCreateAncillaryTable	True		5/13/2022 4:06:47 PM

## Manage System Security

Manage System Security allows non-administrators to manage users, roles and groups and facilitates a comprehensive, functionally-tailored way to separate user and group responsibilities.

By default Administrators have access to all security roles. Assigning other groups to roles does not remove an Administrator's complete system access. Users with Manage System Security roles can access to the System User Interface Roles of SystemAdministrationLogon and SystemPane.

## System Security Roles

---

Roles are not exclusionary or limiting. If granted, users can get additional functionality through their membership in groups having corresponding role assignments. See "System Security Roles" on page 864.

For example:

- A member of the IT team who is not an administrator might need to manage the system security roles.
- An employee in the Accounting department who reports to the Controller over consolidations may manage groups and not need access to other areas of the system.

You can delete these security roles to non-Administrators:

- **ManageSystemSecurityUsers**: Grants permission to manage users.
- **ManageSystemSecurityGroups**: Grants permission to manage groups, exclusion groups and group membership.
- **ManageSystemSecurityRoles**: Grants permission to manage system security role assignment.

## Access System Security Roles

1. From the System tab, go to **Administration > Security**.
2. Select **System Security Roles**.

The screenshot shows a software interface titled "Security - System Security Roles". On the left, there's a navigation sidebar with icons for users, groups, and exclusion groups. Below it is a "Filter" section with a dropdown menu set to "System Security Roles". The main area is a table titled "System Security Roles" with two columns: "Role" and "Group". The table lists various system roles and their assigned groups. The last three rows, which are the ones described in the text as being deletable by non-administrators, are highlighted with a dark blue border.

System Security Roles	
ManageSystemDashboards	Administrators
ManageSystemDatabaseFiles	Administrators
ManageFileShare	Everyone
ManageFileShareContents	Administrators
AccessFileShareContents	Administrators
RetrieveFileShareContents	Administrators
EncryptSystemBusinessRules	Nobody
ViewAllLogonActivity	Administrators
ViewAllErrorLog	Administrators
ViewAllTaskActivity	Administrators
ManageSystemSecurityUsers	Administrators
ManageSystemSecurityGroups	Administrators
ManageSystemSecurityRoles	Administrators

# Manage System Security Users

This role enables you to:

- Create users
- Modify users
- Delete users
- Disable users

This role enables you to specify these user properties:

- General
- Authentication
- Preferences
- Custom Text

### Limitations

Users with the Manage System Security role cannot create, modify, or delete administrators, directly or indirectly. Also, they cannot:

- Add or remove themselves to or from groups or roles.
- Delete themselves.
- Add other users to Manage System Security privileges.
- Add or remove groups they are members of from roles.

To manage group membership or copy users, the **ManageSystemSecurityGroup** is required.

# Manage System Security Groups

This role lets you manage groups and exclusion groups. You can also:

## System Security Roles

---

- Add or remove members to or from groups and exclusion groups.
- Copy groups except groups with Administrator privileges.

### Limitations

Users with this role cannot:

- Modify the Administrators group.
- Assign users to a group that establishes Administrator privileges.
- Modify your membership in other groups.
- Modify the parent group of a group in which the user is a member.

## Manage System Security Roles

This role lets you manage system security roles. However, you cannot:

- Modify the **ManageSystemSecurityRole** itself because it requires Administrator level privileges.
- Assign the Everyone or Nobody groups that require Administrator level privileges.
- Add a group to a role of which you are a member.

## Load and Extract

Load and Extract functionality of Security requires a user to have permissions for all three of the Manage System Security roles, as well as the System User Interface Roles of **SystemLoadandExtractPage**.

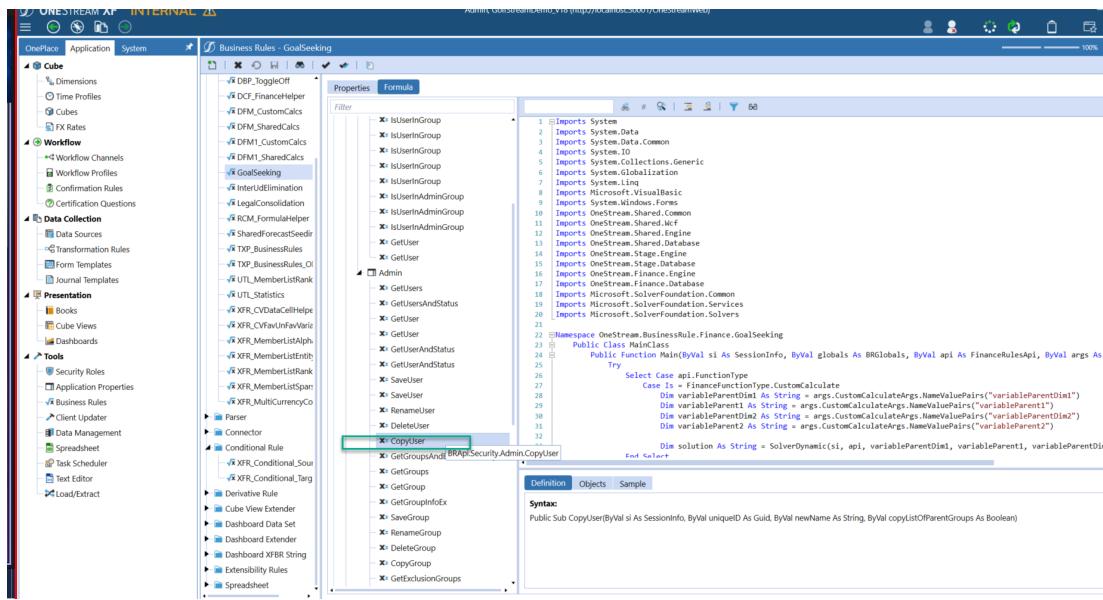
The controls limiting Manage System Security user's capabilities is enforced during the Load and Extract process. Validation occurs by comparing the current state of security in the target environment to the changed state determined by the processing of the source file. Therefore, the validation of XML loads for Manage System Security users requires that security is pre-existing to determine the changed state.

For example, although Manage System Security users cannot create Administrators, if the current Administrator Groups existed in the target environment prior to the XML load, then the XML will pass the validation and will be loaded. However, when an empty or new environment exists with no pre-existing users and groups, then an Administrator would need to perform the load.

## System Security Roles

### BRApi

You can manage user and group system security using BRApi functions such as CopyUser, DeleteUser and CopyGroup. These are controlled by the assigned Manage System Security role. See "System Security Users and Groups" on page 884 for more information.



For example, if a dashboard is created to insert new users, and a dashboard button executed a BRApi to insert a user, the system validates that the user clicking the button is in the Administrators Group or has role permission to ManageSystemSecurityUsers.

## Combined Roles

When granted access to more than one of these roles, you gain more functionality within the scope of the designed capabilities and restrictions. For example, if you have both the role of users and groups, you can copy a User or you can also can add a user to a group. Certain functionality requires assignment of combined roles, such as Load and Extract.

## File Share Security Roles

### ManageFileShareContents

## System Security Roles

---

- Exposes the Contents Folder in File Explorer\FileShare under Application and System.
- Grants full rights to create, upload, download and delete folders.

### AccessFileShareContents

- Exposes the Contents Folder in File Explorer\FileShare under Application and System.
- Allows only the ability to see the Contents folder and its sub-files, and allows download.

### RetrieveFileShareContents

- The Contents Folder is not exposed to the user in the File Share for Application or System using the File Explorer.
- All files are accessible through the OneStream application, such as through Dashboards and Business Rules.

See Application Security Roles in Application Tools for more information on Application based security roles.

**Encrypt System Business Rules** - This allows a user to encrypt/decrypt a rule from the Business Rule screen in the System tab to obfuscate the contents of the rule from all users..

See Business Rules in under Application Tools for more information on Business Rule Encrypt (Decrypt) functionality and use.

**View All Logon Activity** - If assigned the required access to view the System tab and the System tab | Logging | Logon Activity page, users in the assigned security group can see the Logon activity for all users in the environment.

**View All Error Log** - If assigned the required access to view the System tab and the System tab | Logging | Error Log page, users in the assigned security group can see the Error Log for all users in the environment.

**View All Task Activity** - Users in the assigned security group can view the tasks and detailed child-steps through the Task Activity icon in the toolbar. If they have access to the System tab and page, they can also see this information in the System tab | Logging | Task Activity page.

# System User Interface Roles

The following roles grant access to key features and tools.

## **System Security Roles**

---

### **SystemAdministrationLogon**

This is for the System Administration application and set to Administration by default. When a security group is assigned to this role, it becomes available in the Application list during logon.

### **SystemPane**

Based on the security group chosen, this role lets you access the System Tab found at the bottom left of the screen. By default, only administrators have access to this tab.

### **ApplicationAdminPage**

Based on the security group chosen, this role lets you access the Application tab at the bottom left of the screen. By default, only administrators can access this tab.

### **SecurityAdminPage**

Based on the selected security group, this role lets you view but not modify security artifacts and settings other than your password on System > Administration. By default, only administrators have access to this section.

### **SystemDashboardAdminPage**

Based on the security group chosen, this will allow a user access to the Dashboard section found under System Tab > Administration > Dashboards. This section ties to the ManageSystemDashboards which is under the System Security Roles in the above section. By default, only administrators have access to this.

### **ApplicationServersPage**

Based on the security group chosen, this will allow a user access to the Application Servers section found under System Tab > Tools > Application Servers. By default, only administrators have access to this.

### **DatabasePage**

This role lets you access the Database Page on System > Tools. By default, only system administrators have access to this section.

### **FileExplorerPage**

Based on the selected security group, this role lets you access the File Explorer on System > Tools. This section ties to the ManageSystemDatabaseFiles and ManageFileShare roles. By default, only administrators can access the File Explorer.

### **SystemLoadExtractPage**

Based on the security group chosen, this role lets you access Load/Extract on System > Tools but you cannot actually import or extract items. By default, only administrators have access to this section.

### **ErrorLogPage**

Based on the security group chosen, this role lets you access Error Log on System > Logging. By default, only administrators can access to this section.

### **LogonActivityPage**

Based on the selected security group, this role lets you access to Logon Activity on System > Logging. You can view all users, but cannot log them off. By default, only administrators have access to this section.

### **TaskActivityPage**

Based on the selected security group, this role lets you access Task Activity in System > Logging. By default, only administrators can access to this section.

### **TimeDimensionPage**

Based on the security group chosen, this will allow a user access to the Time Dimension on System > Tools. By default, only administrators have access to this section.

# System Security Users and Groups

Administrators click **System > Administration > Security** to define and manage users, groups and system security roles (roles). Users, groups and roles are not application-specific. See "Creating and Managing Groups" on page 856 and "Creating and Managing Users " on page 847.

Every user must be assigned a user ID. Users can be added as native users or as references to users stored in external repositories (e.g. Active Directory). Users can be externally authenticated with these standard providers.

- LDAP
- MSAD
- Okta
- PingFederate
- Azure AD
- SAML

For information about external authentication with standard providers, see the *Installation and Configuration Guide*.

# Managing Users and Groups Using BRApi Functions

Administrators can manage users and groups with the following BRApi functions:

## System Security Roles

---

```
BRApi.Security.Admin.GetUsers
BRApi.Security.Admin.GetUser
BRApi.Security.Admin.GetUser
BRApi.Security.Admin.SaveUser
BRApi.Security.Admin.RenameUser
BRApi.Security.Admin.DeleteUser
BRApi.Security.Admin.CopyUser
BRApi.Security.Admin.GetGroupsAndExclusionGroups
BRApi.Security.Admin.GetGroups
BRApi.Security.Admin.GetGroup
BRApi.Security.Admin.GetGroupInfoEx
BRApi.Security.Admin.SaveGroup
BRApi.Security.Admin.RenameGroup
BRApi.Security.Admin.DeleteGroup
BRApi.Security.Admin.CopyGroup
BRApi.Security.Admin.GetExclusionGroups
BRApi.Security.Admin.GetExclusionGroup
BRApi.Security.Admin.SaveExclusionGroup
BRApi.Security.Admin.RenameExclusionGroup
BRApi.Security.Admin.DeleteExclusionGroup
BRApi.Security.Admin.CopyExclusionGroup
BRApi.Security.Admin.GetSystemRoles
BRApi.Security.Admin.GetApplicationRoles
BRApi.Security.Admin.GetRole
BRApi.Security.Admin.CopyExclusionGroup
```

## Examples

Get a UserInfo object and change the User Description

```
Dim objUserInfo As UserInfo = BRApi.Security.Admin.GetUser(si, "Administrator")
If Not objUserInfo Is Nothing Then
    objUserInfo.User.Description = "New Description"
    BRApi.Security.Admin.SaveUser(si, objUserInfo.User, False, Nothing, TriStateBool.Unknown)
End If
```

Get a Group and UserInfo object and add the Group to the User's list of parent Groups

```
Dim objGroupInfo As GroupInfo = BRApi.Security.Admin.GetGroup(si, "TestGroup")
```

## System Security Roles

---

```
If Not objGroupInfo Is Nothing Then

    Dim objUserInfo As UserInfo = BRApi.Security.Admin.GetUser(si, "TestUser")
    If Not objUserInfo Is Nothing Then

        If (Not objUserInfo.ParentGroups.ContainsKey(objGroupInfo.Group.UniqueID)) Then
            Dim parentGroupIDs As List(Of Guid) = objUserInfo.ParentGroups.Keys.ToList()
            parentGroupIDs.Add(objGroupInfo.Group.UniqueID)

            BRApi.Security.Admin.SaveUser(si, objUserInfo.User, True, parentGroupIDs,
            TriStateBool.Unknown)
        End If
    End If
End If
```

## Create a User

```
Dim objUser As User = New User()
objUser.Name = "NewUser"
objUser.Text1 = "Test Text 1"
BRApi.Security.Admin.SaveUser(si, objUser, False, Nothing, TriStateBool.Unknown)
```

## Create a Group

```
Dim objGroup As Group = New Group()
objGroup.Name = "NewGroup"
Dim objGroupInfo As GroupInfo = New GroupInfo()
objGroupInfo.Group = objGroup
BRApi.Security.Admin.SaveGroup(si, objGroupInfo, False, Nothing, TriStateBool.Unknown)
```

## International Settings

Culture-aware number and text formatting can be used with:

- The web-based client
- The Excel add-in

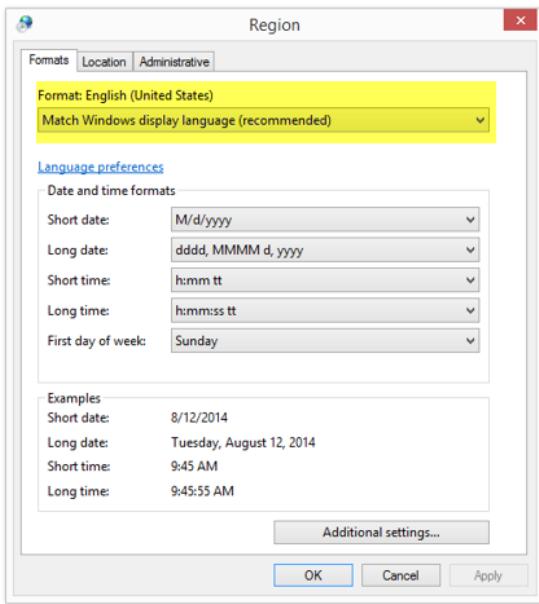
- Server components

For consistency, we recommend that each user's Culture setting match the Windows Regional settings of their primary computer.

## Specifying Culture Settings on the Client Machine Operating System

Number and text formatting for Microsoft Excel is controlled by the client operating system's Windows Regional settings. To configure these settings:

1. Access the operating system's **Control Panel** and select **Region**.
2. Click **Format** and set properties to reflect regional preferences. For example:



3. Click **Location** and set the **Home Location** to reflect the operating system's location such as Canada, Germany or Australia.

## Specifying Server Side User Culture Settings

For the web based client and server components, specify culture settings for users by modifying user profiles in Security. The necessary culture codes must first be defined in the Application Server Configuration file.

# Working With System Applications

After logging on as an Administrator, you can open the System Administration application that lets you flexibly create a new application based on existing databases or copied databases. For existing Applications, click **System > Administration > Applications**.



### Create Application Reference

Click to link an application and an existing database if they are from the same release version. Select a Database Server Connection and enter a Database Schema Name. The new Application will display.

**Allow Database Creation Via UI** must be **True** in the Application Server Configuration.



### Create Application Database

Click to generate a new database.

#### Application Name

Enter the name of the new application.

#### Description

Provide a brief description (optional).

#### Database Server Connection

Select the database server from the drop-down list.

#### Database Schema Name

This is created by default and stored in SQL.

#### Time Dimension Definition

Click the ellipsis and select the desired Time Dimension XML file. See Time Dimensions in "System Tools" on page 894 for details.

The new application displays in the Application list when you log on.

**NOTE:** Database Status indicates if a selected application has an attached database. If the status is "Error", the database was deleted or detached and will not be available in the Application list.



### Delete Selected Application Reference

Click to remove an application's database link. This will not delete database files, just references in the application.



### Test Database Connection

Select an existing application and click the test checkbox to verify an active connection.

## System Dashboards

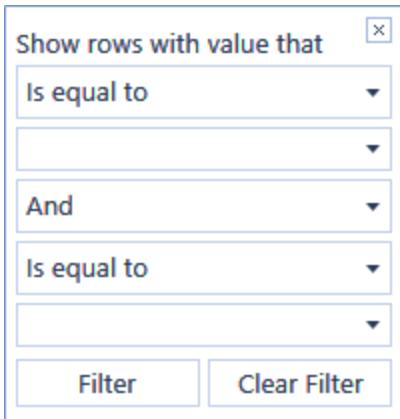
System Dashboards are similar to Application Dashboards but differ in that you can use OneStream Framework database objects such as security and users. Application Dashboards use Application-level data stored in Cubes.

See Application Dashboards in "Presenting Data With Books, Cube Views and Dashboards" on page 547 for information about designing Dashboards.

# Logging

Each logging section display content - that you can sort and filter - in a grid. To sort, click a column heading and select ascending or descending.

Click on  to filter by criteria such as the following:



Show rows with value that

Is equal to

And

Is equal to

Navigate pages by clicking page numbers, the forward button, the back button and so on. These buttons are to the bottom left of a grid. To export data, right-click, select **Export** and then an export file type.



## Logon Activity

Use the following to identify who is logged on who logged off.

### Logoff Selected Session

Administrators can use this to logoff any user session



### Clear Logon Activity

Clears all logon activity

This grid will display:

## Logging

---

### Logon Status

This shows when users logged on and logged off.

### User

This shows the user ID.

### Application

This shows the application the user is logged into.

### Client Module

This will display items such as Excel.

### Client Version

This will display the version of the application you are using.

### Client IP Address

This will display the end users IP address.

### Logon Time

A time stamp of when the user logged in.

### Last Activity Time

A time stamp of the user's last activity.

### Logoff Time

A time stamp of when the user logged off.

### Primary App Server

This shows the application server the end user is utilizing.

## Task Activity

You can access task activity to identify user actions in two ways:

- Click **Task Activity** at the top right section of the web client.
- Select **System > Logging > Task Activity**.



### Clear task activity for current user

Clears all activity for that user.



### Clear task activity for all users

Clears all activity for ALL users.



### Selected task information

Gives the ability to drill down through all steps for that activity.



### Running task progress

Gives the ability to view progress of other users' activities.



### Refresh

This refreshes the Task Activity in order to see any changes made

#### Task Type

This shows the user's activity. (e.g., Consolidate, Process Cube, Load and Transform, Clear Stage Data, etc.)

#### Description

This shows the details of the activity. (e.g., the POV, Multiple Data Units, etc.)

#### Duration

This displays the length of time the activity took.

#### Task Status

This shows the status of the activity. (e.g., Completed, Failed, etc.) Canceling the task will transition it from Canceling to Canceled.

#### User

This shows the user ID.

#### Application

This shows the application to which the task was processed.

#### Server

This shows the application server being utilized.

#### Start Time

The starting time stamp of the activity.

#### End Time

The ending time stamp of the activity.

#### Queued CPU

This provides the % CPU utilization for when the task was initiated.

#### Start CPU

This provides the % CPU utilization when the job began from the queue.

## Logging

---

For more details, Task Queuing, see Data Management in "Application Tools" on page 727.

Within the grid, there are two icons on the left side of the row. If highlighted, there is the ability to drill down by clicking on them.



The first icon shows child steps within a particular task that has run. The second shows detailed information of the error when present.

## Error Logs

Administrators can use the following to evaluate errors on System > Logging.



### **Clear error log for current user**

Clears all logs for a user.



### **Clear error log for all users**

Clears all logs for all users.

This grid will display:

#### **Description**

Displays a brief description of the error.

#### **Error Time**

Indicates when the error occurred.

#### **Error Level**

Displays the error type such as Unknown, Fatal, Warning etc.

#### **User**

Displays the user ID.

#### **Application**

Identifies the application a user is logged onto.

#### **Tier**

Displays the application tier such as App Server, Web Server or Client.

#### **App Server**

Identifies the application server to which a user was connected when they encountered an error.

# System Tools

There are a variety of system tools that can be used to manage the OneStream application. These tools include system business rules, database, tools that allow you to check the overall health of the application environment, and the File Explorer to name a few. In this section you will learn how to use these and other tools to manage the application system.

## System Business Rules

You can use System Extender Business Rules with Azure Server Sets for enhanced scalability at the Azure Database and Server Sets level. You can perform Server and eDTU scaling manually or using System Business Rules. If System Business Rules is selected as a Scaling Type, then a user-defined System Extender Business Rule determines if scaling is needed. You must implement the scaling function and return the proper scaling object by adding a System Extender Business Rule and assigning it appropriately.

Under each Case statement, use these rules and related Args and BRApis to check the current Server Set capacity, query metrics about a Server Set or Azure Database, and identify the impact of the Server Sets volume or level of Azure Database deployed. See "Azure Database Connection Setting and Server Sets" in the *Installation and Configuration Guide*.

Example starting point of empty System Extender Business Rule upon creation:

```
Namespace OneStream.BusinessRule.SystemExtender.ServerSet2
| Public Class MainClass
|   Public Function Main(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Object, ByVal args As SystemExtenderArgs) As Object
|     Try
|       Select Case args.FunctionType
|         Case Is = SystemExtenderFunctionType.Unknown
|         Case Is = SystemExtenderFunctionType.GetDesiredServerSetCapacity
|         Case Is = SystemExtenderFunctionType.GetDesiredElasticDatabasePoolCapacity
|         Case Is = SystemExtenderFunctionType.GetDesiredExternalServerSetCapacity
|       End Select
|     Return Nothing
|     Catch ex As Exception
|       Throw ErrorHandler.LogError(si, New XFEException(si, ex))
|     End Try
|   End Function
| End Class
End Namespace
```

# Sample System Business Rule

Metric are passed to this function to help users determine if a server or database needs scaling. For server scaling, Environment metrics and Scale Set metrics are passed. For database scaling, Environment metrics and SQL Server Elastic Pool metrics are passed.

```
Select Case args.FunctionType

Case Is = SystemExtenderFunctionType.Unknown

Case Is = SystemExtenderFunctionType.GetDesiredScaleSetCapacity
    Dim systemExtenderScaleSetResult As New SystemExtenderScaleSetResult
    systemExtenderScaleSetResult.Capacity = args.ScaleSetArgs.CurrentScaleSetCapacity

    If (args.ScaleSetArgs.ScaleSetMetricValues.AvgCPUUtilization > 50) Then
        systemExtenderScaleSetResult.Capacity = args.ScaleSetArgs.CurrentScaleSetCapacity + 1
    End If

    Return systemExtenderScaleSetResult

Case Is = SystemExtenderFunctionType.GetDesiredElasticDatabasePoolCapacity
    Dim systemExtenderSQLServerElasticPoolResult As New SystemExtenderSQLServerElasticPoolResult
    systemExtenderSQLServerElasticPoolResult.AzureElasticPoolDTU = args.SQLServerElasticPoolArgs.DatabaseAndEPoolDTU.AzureElasticPoolDTU

    If (args.SQLServerElasticPoolArgs.AzureElasticPoolLevelMetricValues.DTUConsumptionPercent > 90)
        systemExtenderSQLServerElasticPoolResult.AzureElasticPoolDTU = 1600
    End If

    Return systemExtenderSQLServerElasticPoolResult

Case Is = SystemExtenderFunctionType.GetDesiredExternalScaleSetCapacity

End Select
```

# Database

The Database screen allows System Administrators to view all of OneStream's database tables and provides tools for managing stored data and other information.

## Tables

This gives read-only access to all data tables in the database and can be used for tasks such as trying to debug issues without having access to the database, or deletion logging.

## Tools

Database Tools allow System Administrators to manage the database.

## Data Records

Enter a Member Filter in order to view data for the entire system.

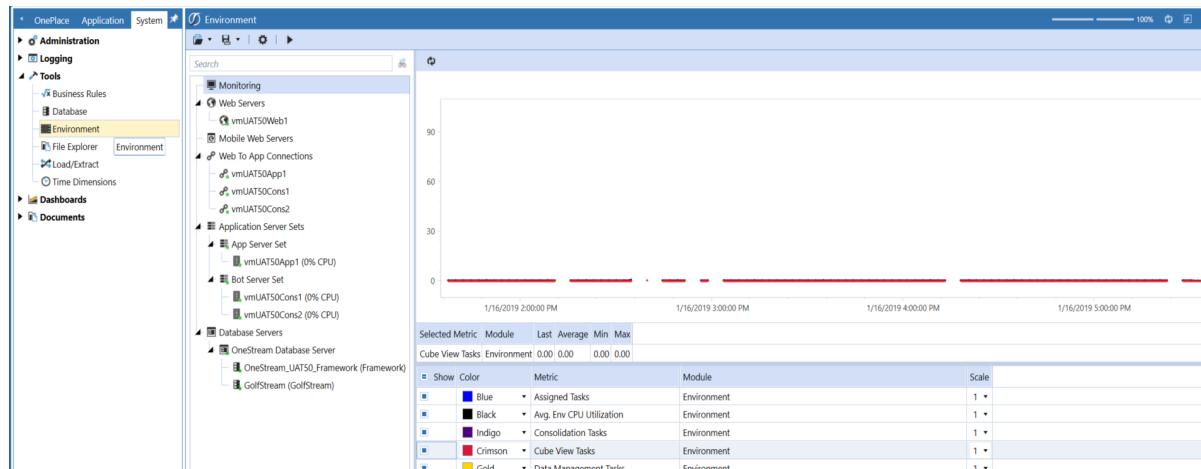
# Environment

This section can be used to check the overall health of the environment, which contains Web Servers, Mobile Web Servers, Application Server Sets and Database Servers. This will check the connection status as well as the configuration.

The Environment page is designed to give both IT and power business users a way to manage and optimize their applications and the environment that is running under. Using the Environment page, the user can monitor the environment, isolate bottlenecks, look at properties and configuration changes, and scale in/out application servers and database resources if needed. They can also customize what data to collect in log files, save collection metrics files, and replay collected performance data in many ways.

To access the Environment page, select **System > Tools > Environment**.

**NOTE:** The Environment page is only accessible via the OneStream Windows App.



# Monitoring Environments

Instead of logging onto the server to collect metrics, use the Monitoring page to access real time Key Performance Indicators (KPIs) across an environment. Click **System > Tools > Environment > Monitoring** to:

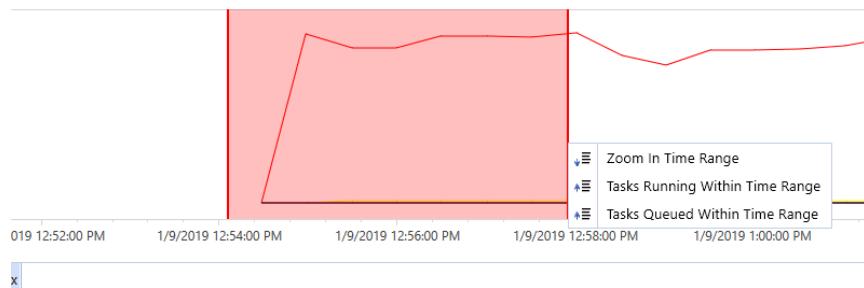
## System Tools

---

- Access user activity and interactive KPI graphs.
- Track all system components that affect stability and performance.

You can perform these tasks:

- **Open:** Access a metrics and configuration setting file from the File System or a local folder.
- **Save As:** Save metrics and configurations locally or in the File System.
- **Settings:** Specify these types of KPI metrics to monitor and the monitor frequency:
  - Environment
  - Application Servers
  - Database Servers
  - Server Sets
- Zoom into part of the chart to see running or queued activities.



- **Refresh Automatically Using a Timer:** Retrieve metrics based on the Play Update Frequency interval setting.

## Specifying General Settings

- **General Play Update Frequency (seconds):** How often to update Performance charts.
- **Metric and Task Time Range:** Indicate how much historical data to retrieve and depict on the Performance Chart. Applying a time range can help identify the cause of a server event or issue.

- **Y-Axis:** If Auto Range box is selected, the system sets the range for the Min and Max values on the Y-Axis. If cleared, you can set the Min and Max ranges on the Y-Axis.
- **Secondary Y-Axis:** Displays series of a different range of values, or different arguments (or values) in the same chart. Can be used when the numbers in a chart vary widely from data series to data series, or when there are mixed types of data.

## Specifying Monitoring Settings

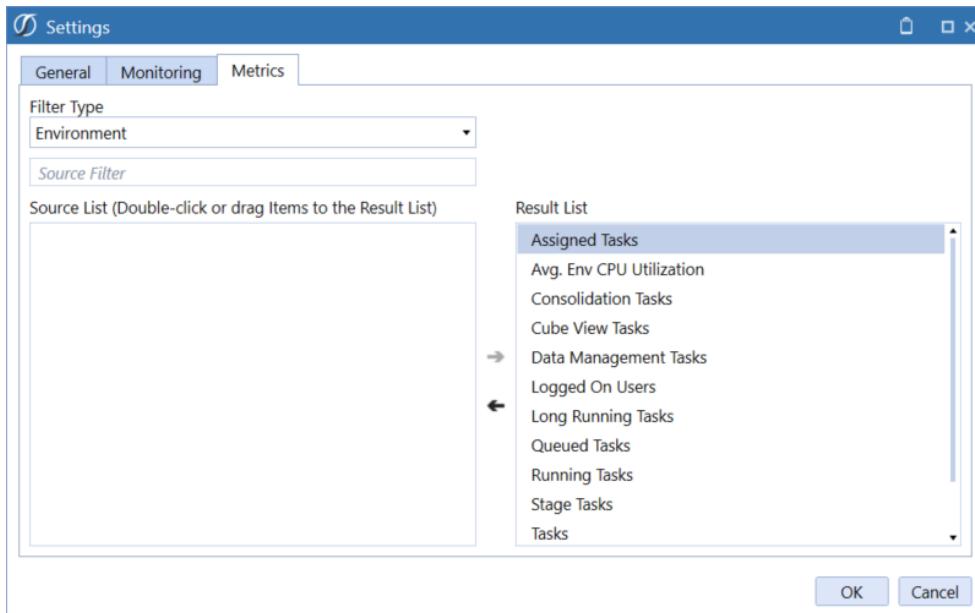
- **Monitoring (Server filter):** Identify specific application or database servers to evaluate.
- **Filter Type:** Specify the type of servers for which to collect metrics.
- **Source Filter:** Use to filter the Source List.
- **Source List:** Displays the servers that meet your criteria.
- **Result List:** Displays selected **Source List** items for which to collect metrics.

## Specifying Metrics Settings

Metrics are collected based on the Metric Update Interval (seconds) setting in the Application Server Configuration utility. To minimize database access and maximize performance, some metrics are collected on every iteration, but some will skip iterations based on count settings defined for a metric.

## System Tools

---



- **Filter Type:** Refine the types of application servers, database servers or server sets for which to collect metrics.
- **Source Filter:** Use to filter the Source List.
- **Source List:** Displays the list of filtered application servers, database servers or server sets.
- **Result List:** Displays the items for which to collect metrics.

These settings shown below in the Application Server Configuration File in Environment Monitoring, determine which metrics are collected and how often.

## System Tools

---

Application Server Configuration Settings	
OneStream XF Environment	
Environment Monitoring	
Number of Running Hours Before Automatic Recycle	24
Start Hour for Automatic Recycle (0 to 24 UTC)	5
End Hour for Automatic Recycle (0 to 24 UTC)	7
Maximum Number of Minutes to Pause Before Automatic Recycle	30
Active Check Update Interval (seconds)	60
Metric Update Interval (seconds)	30
Server Heartbeat Update Interval (seconds)	10
Collect Environment CPU Metrics	Always
Collect Environment Task Metrics	Always
Collect Environment Logon Metrics	Always
Collect Server Set CPU Metrics	Never
Collect Server Set Task Metrics	Never
Collect Server Disk Metrics	Never
Collect Server Memory Metrics	Never
Collect Server Network Card Metrics	Never
Collect SQL CPU Metrics	Never
Collect SQL Page Metrics	Never
Collect SQL Memory Metrics	Never
Collect SQL Connection Metrics	Never
Collect SQL Query Metrics	Never
Collect SQL File Metrics	Never
Collect SQL Elastic Pool CPU Metrics	Never
Collect SQL Elastic Pool DTU Metrics	Never
Collect SQL Elastic Pool Storage Metrics	Never
Collect SQL Elastic Pool Workload Metrics	Never
Number Past Metric Readings for SQL Blocking	5
Fragmentation Iteration Count	600
Fragmentation Percent Threshold	90
Detailed Logging	False
Number of Hours to Retain Offline Servers	1
Number of Days to Retain Server Config History	30
Number of Days to Retain Metrics	30

## Web Servers

This section lists all the web servers that are participating in that environment. Each web server will display its configuration and its audited setting.

## Configuration

Sample Web Server configuration settings:

## System Tools

---

The screenshot shows the System Tools interface with the 'Environment' tab selected. On the left, there's a navigation tree with categories like Monitoring, Web Servers, Mobile Web Servers, Web To App Connections, Application Server Sets, and Database Servers. Under 'Web Servers', 'vmUAT50Web1' is selected. On the right, there are two tabs: 'Configuration' (selected) and 'Audit'. The 'Configuration' tab displays detailed settings for 'vmUAT50Web1'. It includes sections for General (Identity Provider set to Azure, Silverlight Enabled set to True), Web To App Connection - vmUAT50App1 (Name vmUAT50App1, WCF Address https://vmUAT50App1.onestreamcloud.com:50002/OneStreamApp), and Behaviors (Used For General Access, Used For Stage Load, Used For Consolidation, Used For Data Management). Similar sections are shown for Web To App Connection - vmUAT50Cons1 and Web To App Connection - vmUAT50Cons2.

### Identity Provider

Displays single sign on identity provider if used.

### Server Heartbeat Update Interval (seconds)

Used to specify how often each server updates its record that it is alive and responding to user input.

### Name

This is the name of the server as it is defined in the web configuration file.

### WCF Address

The full URL address of the server.

### WCF Connection

This examines the status of the connection. Ok = connected

## System Tools

---

### Used for General Access

This examines the web configuration file to see if it has this server configured as a general server. Settings are True or False.

### Used for Stage Load

This examines the web configuration file to see if this server is configured as a Stage server. Settings are True or False.

### Used for Consolidation

This examines the web configuration file to see if this server is configured as a consolidation server. Settings are True or False.

### Used For Data Management

This examines the web configuration file to see if this server is configured as a Data Management server. Settings are True or False.

## Audit

Sample Web Server audit setting:

The screenshot shows the Audit tab of a management console. On the left is a navigation tree with categories like Monitoring, Web Servers, Mobile Web Servers, Web To App Connections, Application Server Sets, and Database Servers. Under Web Servers, there's a node for 'vmUAT50Web1'. The main pane displays a table of audit logs. The columns are: Property Type, Property, Item, Value From, Value To, Description From, Description To, Timestamp From, and Timestamp To. The table contains four rows of data, each corresponding to a change in the 'WebServerConfigSettings' section of the configuration file. The last row shows a complex XML configuration snippet.

Property Type	Property	Item	Value From	Value To	Description From	Description To	Timestamp From	Timestamp To
XFVersionInfo	XFClientPrerequisitesVersion			4.2.0.0				1/2/2019 11:09 AM
XFVersionInfo	XFCompatibleExcelAddinVersion							1/2/2019 11:09 AM
XFVersionInfo	XFVersion			5.0.0.9709	<WebServerConfigSettingsForDBStorage>	<AppServers>		1/10/2019 9:56 AM
XmlConfigSetting	WebServerConfigSettings				<AppServer name='vmUAT50App1' wcfAdd	<AppServer name='vmUAT50Cons1' wcfAdd		
					<AppServer name='vmUAT50Cons2' wcfAdd	</AppServers>		
					<SSOIdentityProviderType>Azure</SSOIdentity	<SilverlightEnabled>true</SilverlightEnabled		
					<ServerHeartbeatUpdateIntervalInSec>10</Se	</WebServerConfigSettingsForDBStorage>		
							1/2/2019 11:09 AM	

### Property Type

Type of server property in the hardware or software that was changed.

### Property Name

The name of the property in the hardware or software that was changed.

## System Tools

---

### **Value From**

Displays the original value of the property.

### **Value To**

Displays the new value of the property.

### **Description From**

Displays the original description of the setting if available.

### **Description To**

Displays the new description of the setting if available.

### **Timestamp From**

Displays the original date and time of the setting if available.

### **Timestamp To**

Displays the new date and time of the setting.

### **User From**

Displays the original user name if available.

### **User To**

Displays the user name of the person that made the change if available.

## Mobile Web Servers

Similar to web server, the mobile web servers list all the web servers that are included in that environment. It pulls the information from the ServerHeartBeat table in the Framework database.

# Sample Mobile Web Server Configuration

The screenshot shows the System Tools interface with the 'Configuration' tab selected. On the left, a tree view shows nodes like Monitoring, Web Servers, Mobile Web Servers, Web To App Connections, and Application Server Sets. Under Mobile Web Servers, 'vmUAT50Web1' is selected. The main pane displays configuration settings for 'vmUAT50Web1'. The 'General' section includes:

Identity Provider	Azure
Is Silverlight Enabled	True
Server Heartbeat Update Interval (seconds)	10

The 'Web To App Connection - vmUAT50App1' section includes:

Name	vmUAT50App1
WCF Address	https://vmUAT50App1.onestreamcloud.com:50002/OneStreamApp
WCF Custom Endpoint	

The 'Behaviors' section includes:

Used For General Access	True
Used For Stage Load	True
Used For Consolidation	False
Used For Data Management	False

# Sample Mobile Web Server Audit

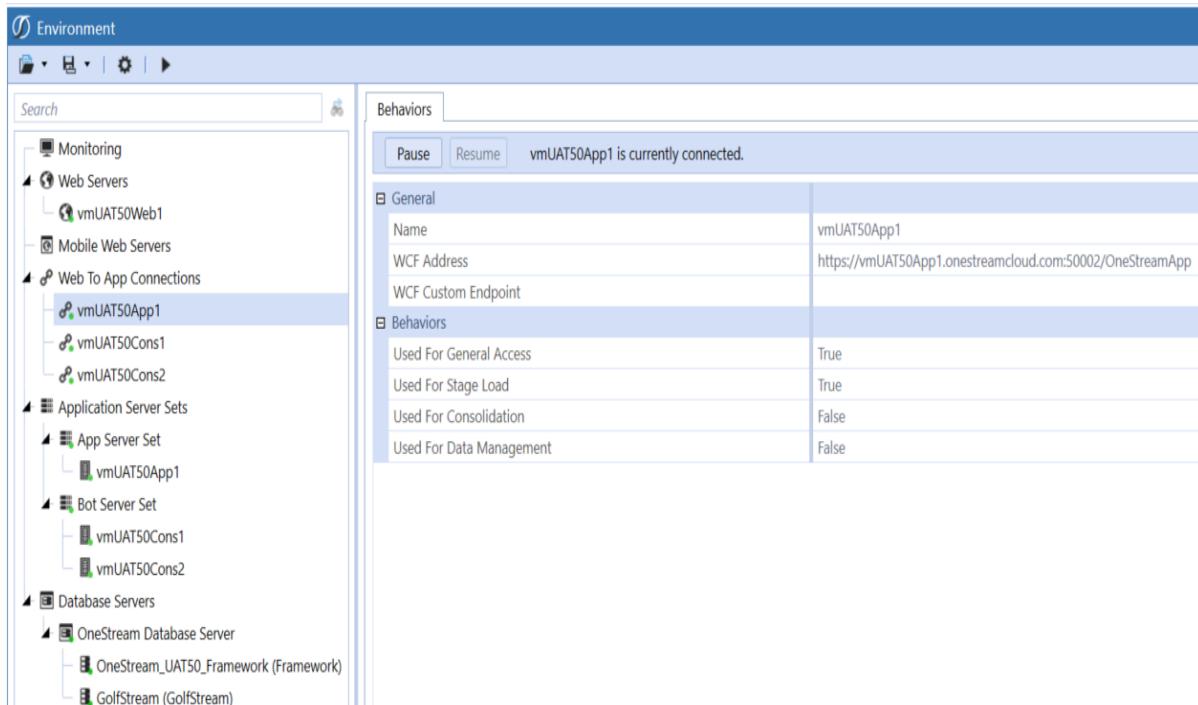
The screenshot shows the System Tools interface with the 'Audit' tab selected. The left pane is identical to the configuration view. The right pane shows an audit log from 1/21/2019 12:00 AM to 1/22/2019 5:11 PM. It lists changes made to 'vmUAT50Web1' properties:

Property Type	Property	Item	Value From	Value To	Description From	Description To
XFVersionInfo	XFClientPrerequisitesVersion			4.2.0.0		
XFVersionInfo	XFCompatibleExcelAddinVersion					
XFVersionInfo	XFVersion			5.0.0.9722		
XmlConfigSetting	WebServerConfigSettings			<WebServerConfigSettingsForDBStorage> <AppServers> <AppServer name="vmUAT50App1" wcfAddi <AppServer name="vmUAT50Cons1" wcfAdd <AppServer name="vmUAT50Cons2" wcfAdc </AppServers> <SSOIDentityProviderType>Azure </SSOIDentity <IsSilverlightEnabled>true </IsSilverlightEnable <ServerHeartbeatUpdateIntervalInSec>10 <Se </WebServerConfigSettingsForDBStorage>		

# WebTo App Server Connections

This section will list all the combined connections from the web server configuration files to all the application servers. From here the user can pause and then resume a specific connection to an application server or a load balancer (a load balancer could point to a multiple application server). See Installation and Configuration Guide.

Sample Web to App connection configuration:



### Pause

Used to pause any request to a WCF Address connection. This connection could be either an Application Sever or a Load Balancer. This can be set in the web configuration file.

### Resume

Used to Resume any request to a WCF Address connection. This connection could be either an Application Sever or a Load Balancer. This can be set in the web configuration file.

### Name

This is the name of the server as it is defined in the web configuration file.

### WCF Address

The full URL address of the server.

### **WCF Connection**

This examines the status of the connection. Ok = connected

### **Used for General Access**

This examines the web configuration file to see if it has this server configured as a general server. Settings are True or False.

### **Used for Stage Load**

This examines the web configuration file to see if this server is configured as a Stage server. Settings are True or False.

### **Used for Consolidation**

This examines the web configuration file to see if this server is configured as a consolidation server. Settings are True or False.

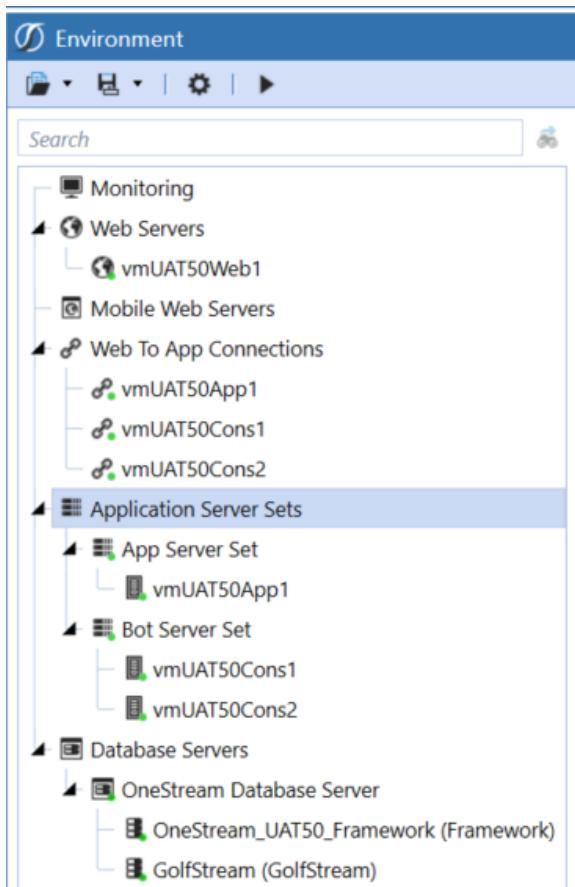
### **Used for Data Management**

This examines the web configuration file to see if this server is configured as a Data Management server. Settings are True or False.

## **Working With Application Server Sets**

Click **Application Server Sets** to view the server sets participating in an environment. Sets display based on the Server Set configuration in the Application Server Configuration Utility. A red "X" indicates offline servers. You can access configurations for each sever set and server in a server set. If a Server is hosted on Azure, you can Scale Out/ and Scale In the Scale Set manually or using a System Business Rule.

Sample Application Server Set:



## Accessing Server Sets Behaviors

Use the **Behavior** tab to view the capability of a Server Set. You can Scale Out (remove) and Scale In (add and configure) if the Scaling Type setting is set to Manual or ManualAndBusinessRule in the Server Configuration Utility file. This enables you to expand or contract server resources manually or automatically. For scaling with Azure, select Server Sets > Azure > Scaling Type. This feature will be available in a future release.

## System Tools

---

The screenshot shows the System Tools interface with the 'Configuration' tab selected for a 'Server Set'. The left pane displays a tree view of server components: Monitoring, Web Servers (vmUAT50Web1), Mobile Web Servers, Web To App Connections (vmUAT50App1, vmUAT50Cons1, vmUAT50Cons2), Application Server Sets (App Server Set containing vmUAT50App1, Bot Server Set containing vmUAT50Cons1, vmUAT50Cons2), and Database Servers (OneStream Database Server, OneStream\_UAT50\_Framework (Framework), GolfStream (GolfStream)). The right pane shows configuration settings:

General	
Name	App Server Set
Type	Standard
State	OK
Capacity	
Behaviors	
Process Queued Stage Tasks	True
Process Queued Consolidation Tasks	False
Process Queued Data Management Tasks	False
Queued Tasks Require Named Application Server	False

You can specify the following settings at the Server Set level using the Server Configuration Utility, but can override them at the individual Application Server level.

- **Process Queued Consolidation Tasks:** If true, the server can process consolidation tasks.
- **Process Queued Data Management Tasks:** If true, the server can process data management tasks.
- **Process Queued Stage Tasks:** If true, the server can process stage tasks.
- **Queued Tasks Require Named Application Server:** If true, the server only runs assigned tasks.

## Server Sets Configuration

Use the **Configuration** tab to view Server Set configurations defined with the Application Server Configuration Utility. For example:

## System Tools

---

The screenshot shows the System Tools interface with the 'Environment' tab selected. On the left is a navigation tree:

- Monitoring
- Web Servers
  - vmUAT50Web1
- Mobile Web Servers
- Web To App Connections
  - vmUAT50App1
  - vmUAT50Cons1
  - vmUAT50Cons2
- Application Server Sets
  - App Server Set
    - vmUAT50App1
  - Bot Server Set
    - vmUAT50Cons1
    - vmUAT50Cons2
- Database Servers
  - OneStream Database Server
    - OneStream\_UAT50\_Framework (Framework)
    - GolfStream (GolfStream)

On the right, there are four tabs: Behaviors, Configuration, Audit, and Performance. The Behaviors tab is selected, displaying a table of properties:

Property	Item	Value	Description
CanChangeQueueingOptionsOnServers		False	
CanPauseResumeServers		False	
CanRecycleAppPoolOnServers		False	
ProcessQueuedConsolidationTasks		False	
ProcessQueuedDataManagementTasks		False	
ProcessQueuedStageTasks		True	
QueuedTasksRequireNamedAppServer		False	
ServerNames		vmUAT50App*	
ServerSetProviderType		Standard	

- **Azure Resource Group Name:** (Azure Scale Set Only) The resource group name for the set.
- **Azure Scale Set Name:** (Azure Scale Set Only) The scale set name in the resource group.
- **Can Change Queuing Options On Servers:** If true, Administrators can change the queuing behavior - impacting stage, consolidation and data management tasks - of specific servers.
- **Can Pause or Resume Servers:** If set to true, then the user can pause and/or resume the server from the Environment page via the Pause and Resume buttons.
- **Can Stop or Start Servers** (Azure Scale Set Only): If true, users can stop / restart the server on the Environment page.
- **Maximum | MinimumCapacity** (Azure Scale Set Only): A fail-safe setting to identify the maximum | minimum number of servers.

## System Tools

---

- **Process Queued Consolidation Tasks:** If true, the server can process consolidation tasks.
- **Process Queued <feature> Tasks:** If true, the server can process tasks such as data management and state tasks.
- **Queued Tasks Require Named Application Server:** If true, the server only runs its assigned tasks.
- **Server Names for Standard Server Sets (Supports \*? Wildcards):** Specify server names if the standard server set type is used.
- **Server Set Provider Type:** Specify the external authentication provider used, such as Azure.
- **System Business Rule Name (Azure Scale Set Only):** The name of the extender business rule for Azure scale set and database scaling.

## Using Server Sets Audit

The **Audit** tab identifies all property changes made in **XFAppServerConfig.xml**. Changes display in yellow. For example:

The screenshot shows the Environment tool interface with the Audit tab selected. The left pane displays a tree view of server components: Monitoring, Web Servers (vmUTS0Web1), Mobile Web Servers, Web To App Connections (vmUTS0app1, vmUTS0Cons1, vmUTS0Cons2), Application Server Sets (App Server Set, Bot Server Set), and Database Servers (OneStream Database Server, OneStream\_UTS0\_Framework (Framework), GolfStream (Golfstream)). The right pane shows a table of audit logs. The table has columns: Property Type, Property, Item, Value From, Value To, Description From, Description To, Timestamp From, Timestamp To, User From, and User To. A single row is highlighted in yellow, representing a change in the 'CanRecycleAppPoolOnServers' property of an 'XmlConfigSetting' item from 'False' to 'True' at timestamp 1/10/2019 9:56 AM.

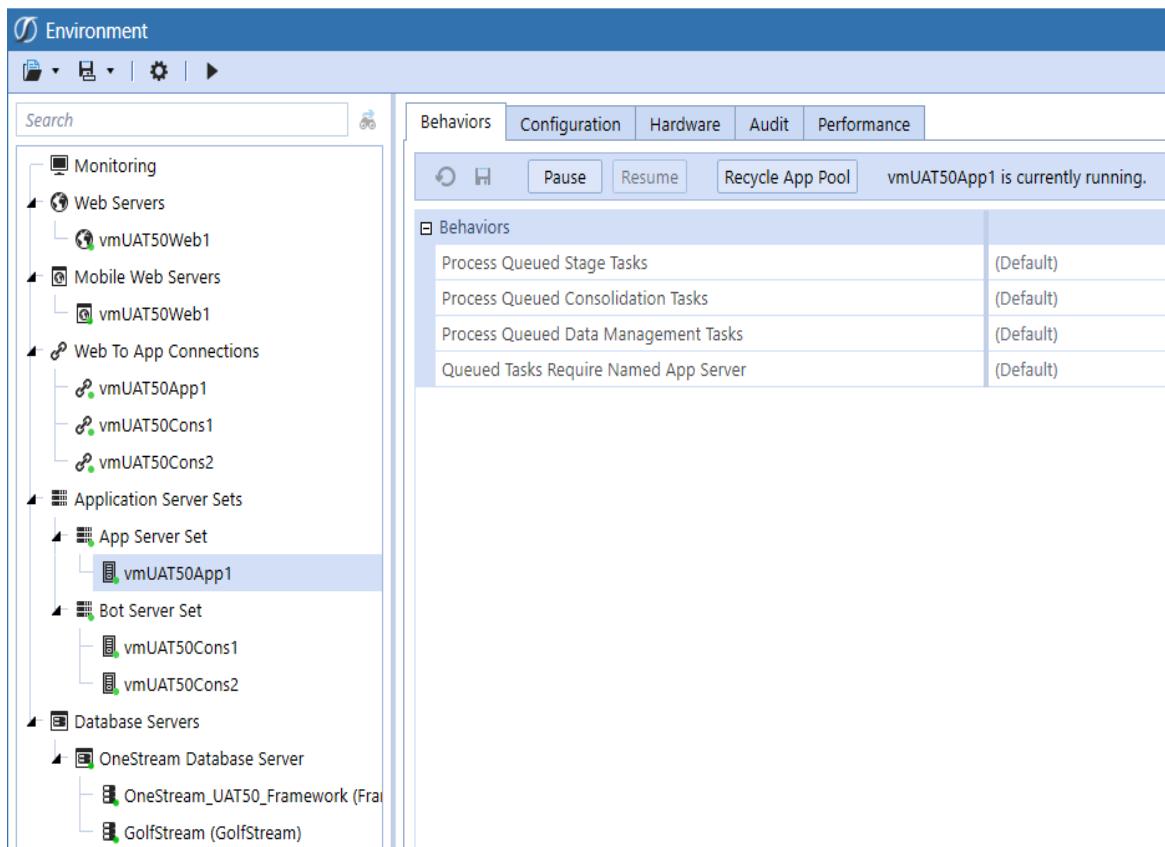
Property Type	Property	Item	Value From	Value To	Description From	Description To	Timestamp From	Timestamp To	User From	User To
XmlConfigSetting	CanRecycleAppPoolOnServers		False	True				1/10/2019 9:56 AM		

## Monitoring Server Sets Performance

Use the **Performance** tab to view scale set and environment metrics, highlighting sections to drill down to time-specific records.

# Application Server Behavior

When selected the user can pause and resume that server, so that none of the long duration tasks can be run on that particular server. Also, IIS can be recycled by selecting the Recycle App Pool button. Whether these buttons appear or not is based on settings in the OneStream Server Configuration Utility.



### Pause

Pause will stop the server from seeking more tasks to run from the queue but will let tasks that have already started to finish.

### Resume

Server will resume accepting tasks from the queue.

### Recycle App Pool

Used to recycle the specific server's application pool (reset IIS).

### Stop - (Azure Scale Set Only)

## System Tools

---

Stops the server, but while in this state will continue to incur Azure compute charges. The public and internal will be preserved.

### Stop (Deallocate)- (Azure Scale Set Only)

Stopping this way will mean the cost of virtual machine will not be charged, but the public and internal IP will be deleted.

## Application Server Configurations

This tab will display the server configurations from the OneStream Server Configuration Utility pertaining to that server.

The screenshot shows the OneStream Server Configuration Utility interface. The left sidebar contains a tree view of server components: Monitoring, Web Servers (vmUAT50Web1), Mobile Web Servers (vmUAT50Web1), Web To App Connections (vmUAT50App1, vmUAT50Cons1, vmUAT50Cons2), Application Server Sets (App Server Set, Bot Server Set), and Database Servers (OneStream Database Server, GfStream). The 'App Server Set' under Application Server Sets is selected. The main pane has tabs for Behaviors, Configuration, Hardware, Audit, and Performance. The Configuration tab is active, displaying a table of properties for the selected App Server Set. The table has columns for Property, Item, Value, and Description. The data is as follows:

Property	Item	Value	Description
App Server Set	CanChangeQueueingOptionsOnServers	False	
App Server Set	CanPauseResumeServers	True	
App Server Set	CanRecycleAppPoolOnServers	True	
App Server Set	ProcessQueuedConsolidationTasks	False	
App Server Set	ProcessQueuedDataManagementTasks	False	
App Server Set	ProcessQueuedStageTasks	True	
App Server Set	QueuedTasksRequireNamedAppServer	False	
App Server Set	ServerNames	vmUAT50App*	
App Server Set	ServerSetProviderType	Standard	
Bot Server Set	CanChangeQueueingOptionsOnServers	False	
Bot Server Set	CanPauseResumeServers	False	
Bot Server Set	CanRecycleAppPoolOnServers	False	
Bot Server Set	ProcessQueuedConsolidationTasks	True	
Bot Server Set	ProcessQueuedDataManagementTasks	True	
Bot Server Set	ProcessQueuedStageTasks	False	
Bot Server Set	QueuedTasksRequireNamedAppServer	False	
Bot Server Set	ServerNames	vmUAT50Cons*	
Bot Server Set	ServerSetProviderType	Standard	

## Application Server Hardware

This tab will show the machine hardware information based on settings in the OneStream Server Configuration Utility.

## System Tools

---

The screenshot shows a software interface titled "Environment". On the left is a tree view of server components: Monitoring, Web Servers (vmUAT50Web1), Mobile Web Servers (vmUAT50Web1), Web To App Connections (vmUAT50App1, vmUAT50Cons1, vmUAT50Cons2), Application Server Sets (App Server Set: vmUAT50App1; Bot Server Set: vmUAT50Cons1, vmUAT50Cons2), and Database Servers (OneStream Database Server: OneStream\_UAT50\_Framework (Frame), GolfStream (GolfStream)). On the right is a table with the "Audit" tab selected, displaying system properties like BIOS caption, manufacturer, and CPU benchmark.

Property	Value	Description
BIOScaption	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	BIOS caption
BIOSmaker	American Megatrends Inc.	BIOS maker
BoardMaker	Microsoft Corporation	Board Maker
BootDevice	\Device\HarddiskVolume1	BootDevice
BuildNumber	14393	BuildNumber
BuildType	Multiprocessor Free	BuildType
Caption	Microsoft Windows Server 2016 Datacenter	Caption
CodeSet	1252	CodeSet
ComputerName	vmUAT50App1	Computer Name
CountryCode	1	CountryCode
CPUBenchmark	57.99	CPU Benchmark
CPUCurrentClockSpeed	2295	CPU Current Clock Speed
CPUManufacturer	GenuineIntel	CPU Manufacturer
CpuSpeedinGHz	2.295	Cpu Speed In GHz
CreationClassName	Win32_OperatingSystem	CreationClassName
CSCreationClassName	Win32_ComputerSystem	CSCreationClassName
CSName	vmUAT50App1	CSName
CultureName	English (United States)	Culture Name
CultureNativeName	English (United States)	Culture Native Name
CurrentTimeZone	0	CurrentTimeZone
DataExecutionPrevention_32BitApplications	True	DataExecutionPrevention_32BitApplications
DataExecutionPrevention_Available	True	DataExecutionPrevention_Available
DataExecutionPrevention_Drivers	True	DataExecutionPrevention_Drivers
DataExecutionPrevention_SupportPolicy	3	DataExecutionPrevention_SupportPolicy
Debug	False	Debug
Description		Description
Distributed	False	Distributed

## Application Server Audit

The audit tab will keep track and display any hardware and configuration changes.

## System Tools

# Application Server Performance

This tab will display metric values pertaining to that server and the Environment.

The screenshot shows the OneStream Environment monitoring interface. The left sidebar lists various server categories: Monitoring, Web Servers (vmUAT50Web1), Mobile Web Servers (vmUAT50Web1), Web To App Connections (vmUAT50App1, vmUAT50Cons1, vmUAT50Cons2), Application Server Sets (App Server Set: vmUAT50App1 (1% CPU)), Bot Server Set (vmUAT50Cons1 (0% CPU), vmUAT50Cons2 (0% CPU)), and Database Servers (OneStream Database Server: OneStream\_UAT50\_Framework (Fram, GofStream (Gofstream))).

The main area displays a chart titled "Behaviors" showing CPU Utilization over time from 1/24/2019 1:00:00 PM to 1/24/2019 4:00:00 PM. The chart has two lines: a red line representing the average CPU utilization for the application server, which remains relatively flat around 55-60%, and a black line representing the average CPU utilization for the bot servers, which remains near 0%. A sharp drop in the red line occurs around 3:30 PM.

Selected Metric	Module	Last	Average	Min	Max
Show	Color	Metric	Module	Scale	
Black	Avg. Env CPU Utilization	Environment	1		
Purple	Tasks	Environment	1		
Crimson	% CPU Utilization	vmUAT50App1	1		
Gold	Cube View Tasks On Server	vmUAT50App1	1		
Navy	Tasks On Server	vmUAT50App1	1		

# Database Servers (Connection Items)

The Database Servers section will list all the database connections based on settings in the OneStream Server Configuration Utility and all the databases that each connection is pointing to.

From the database connection items, the user can expand the SQL server elastic pool if one is configured, view hardware and server properties, view audit info, look at SQL server metrics, and run some diagnostic commands to track down performance issues.

## Behaviors

Sample database connection Behavior tab:

The screenshot shows the OneStream Server Configuration Utility interface. The left pane displays a tree view of monitoring components, including Web Servers (vmUAT50Web1), Mobile Web Servers, Web To App Connections (vmUAT50App1, vmUAT50Cons1, vmUAT50Cons2), Application Server Sets (App Server Set, Bot Server Set), and Database Servers (OneStream Database Server, OneStream\_UAT50\_Framework, GolfStream). The 'Behaviors' tab is selected in the top navigation bar. The right pane shows configuration details for the selected database connection, specifically the 'Behaviors' section. The table lists the following settings:

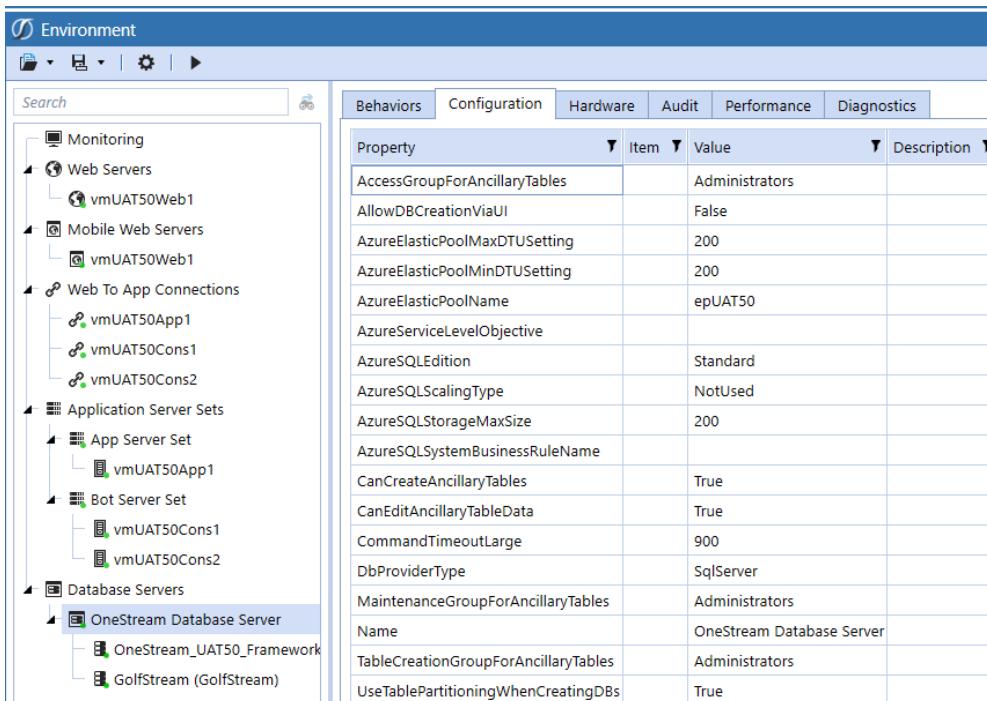
Setting	Value
Elastic Pool Name	epUAT50
Elastic Pool Edition	Standard
Elastic Pool DTU	200
Elastic Pool Database Min DTU	0
Elastic Pool Database Max DTU	200
Elastic Pool Database Storage Capacity	204800

The behavior will show up if SQL Server Azure and Elastic Pool are configured and used. Using the Behavior tab, you can increase or decrease the resources available to an Elastic Pool based on resource needs. When rescaling Elastic Pool DTUs, database connections are briefly dropped. This is the same behavior as occurs when rescaling Elastic Pool DTUs for a single database (not in a pool).

# Configuration

The configuration tab will display SQL server configuration properties.

Sample configuration properties



The screenshot shows the 'Configuration' tab of the System Tools interface. On the left is a tree view of server components under 'Environment'. The 'Database Servers' node is expanded, showing 'OneStream Database Server' and its sub-items: 'OneStream\_UAT50\_Framework' and 'GolfStream (GolfStream)'. On the right is a table listing configuration properties. The table has columns: Property, Item, Value, and Description. The data is as follows:

Property	Item	Value	Description
AccessGroupForAncillaryTables		Administrators	
AllowDBCreationViaUI		False	
AzureElasticPoolMaxDTUSetting		200	
AzureElasticPoolMinDTUSetting		200	
AzureElasticPoolName		ePUAT50	
AzureServiceLevelObjective			
AzureSQLEdition		Standard	
AzureSQLScalingType		NotUsed	
AzureSQLStorageMaxSize		200	
AzureSQLSystemBusinessRuleName			
CanCreateAncillaryTables		True	
CanEditAncillaryTableData		True	
CommandTimeoutLarge		900	
DbProviderType		SqlServer	
MaintenanceGroupForAncillaryTables		Administrators	
Name		OneStream Database Server	
TableCreationGroupForAncillaryTables		Administrators	
UseTablePartitioningWhenCreatingDBs		True	

# Hardware

The hardware tab will display hardware related information pertaining to SQL server. Sample hardware properties:

## System Tools

---

The screenshot shows the System Tools interface with the 'Configuration' tab selected. On the left, a tree view displays various system components: Monitoring, Web Servers (vmUAT50Web1), Mobile Web Servers (vmUAT50Web1), Web To App Connections (vmUAT50App1, vmUAT50Cons1, vmUAT50Cons2), Application Server Sets (App Server Set, vmUAT50App1), Bot Server Set (vmUAT50Cons1, vmUAT50Cons2), and Database Servers (OneStream Database Server, OneStream\_UAT50\_Framework, GolfStream). The 'OneStream Database Server' node is expanded. On the right, a table lists configuration properties with their values and descriptions.

Property	Value	Description
access check cache bucket count	0	Default hash bucket count for the access check result security cache
access check cache quota	0	Default quota for the access check result security cache
Ad Hoc Distributed Queries	0	Enable or disable Ad Hoc Distributed Queries
affinity I/O mask	0	affinity I/O mask
affinity mask	0	affinity mask
affinity64 I/O mask	0	affinity64 I/O mask
affinity64 mask	0	affinity64 mask
Agent XPs	0	Enable or disable Agent XPs
allow filesystem enumeration	0	Allow enumeration of filesystem
allow polybase export	0	Allow INSERT into a Hadoop external table
allow updates	0	Allow updates to system tables
automatic soft-NUMA disabled	0	Automatic soft-NUMA is enabled by default
backup checksum default	0	Enable checksum of backups by default
backup compression default	0	Enable compression of backups by default
blocked process threshold (s)	20	Blocked process reporting threshold
BuildClrVersion		
c2 audit mode	0	c2 audit mode
clr enabled	1	CLR user code execution enabled in the server
clr strict security	1	CLR strict security enabled in the server
Collation	SQL_Latin1_General_CI_AS	
CollationID	872468488	
column encryption enclave type	0	Type of enclave used for computations on encrypted columns
common criteria compliance enabled	0	Common Criteria compliance mode enabled
ComparisonStyle	196609	
ComputerNamePhysicalNetBIOS		
contained database authentication	0	Enables contained databases and contained authentication
cost threshold for parallelism	5	cost threshold for parallelism
cross db ownership chaining	0	Allow cross db ownership chaining
cursor threshold	-1	cursor threshold
Database Mail XPs	0	Enable or disable Database Mail XPs

## Audit

The audit tab will display any changes to the SQL Server properties. Sample configuration audit report:

## System Tools

---

The screenshot shows the System Tools interface with the 'Performance' tab selected. On the left is a navigation tree with categories like Monitoring, Web Servers, Mobile Web Servers, Web To App Connections, Application Server Sets, and Database Servers. The Database Servers section is expanded, showing OneStream Database Server, OneStream\_UAT50\_Framework, and GolfStream (GolfStream). The main area displays a table of performance properties. The table has columns for Property Type, Property, Item, Value From, Value To, and Description. A search bar at the top allows filtering by date range (From: 1/21/2019 12:00 AM, To: 1/22/2019 5:26 PM) and a checkbox for 'View only changed items'. The table includes rows for various database properties such as access check cache bucket count, affinity mask, and common criteria compliance enabled.

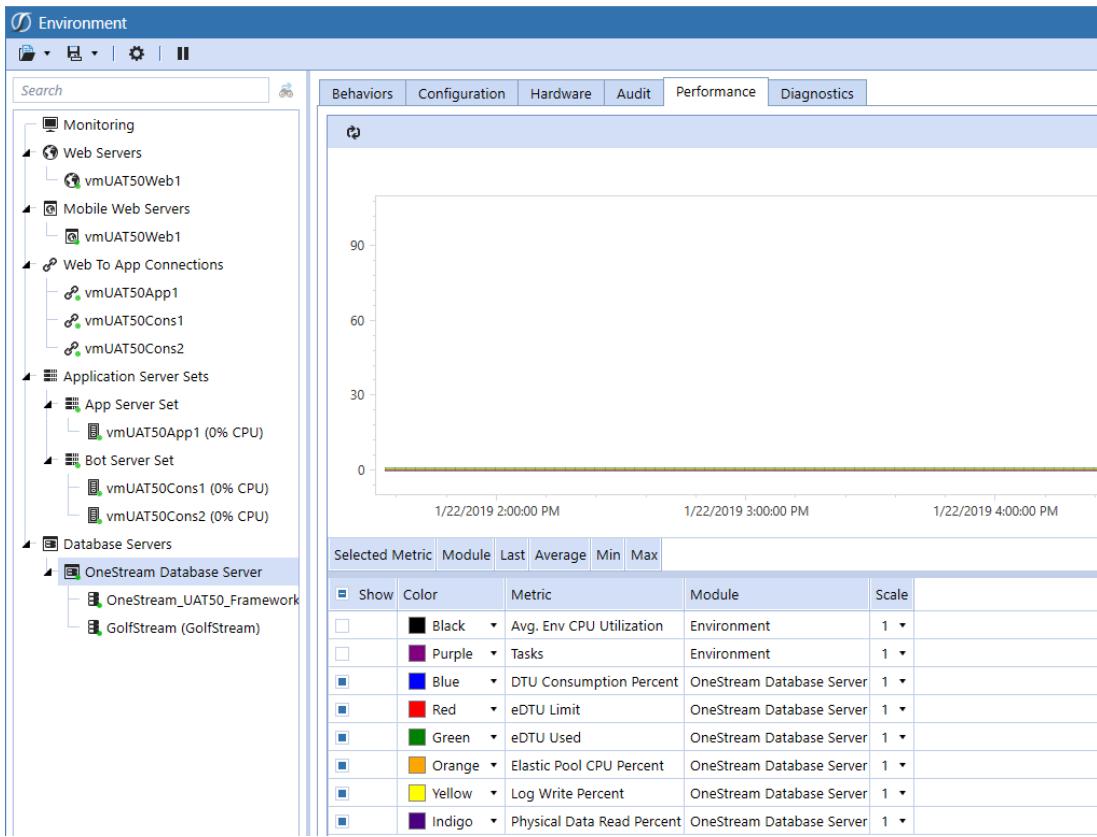
Property Type	Property	Item	Value From	Value To	Description
Database	access check cache bucket count	Instance Configuration	0	0	Default hash bucket count
Database	access check cache quota	Instance Configuration	0	0	Default quota for table access
Database	Ad Hoc Distributed Queries	Instance Configuration	0	0	Enable or disable Ad Hoc distributed queries
Database	affinity I/O mask	Instance Configuration	0	0	affinity I/O mask
Database	affinity mask	Instance Configuration	0	0	affinity mask
Database	affinity64 I/O mask	Instance Configuration	0	0	affinity64 I/O mask
Database	affinity64 mask	Instance Configuration	0	0	affinity64 mask
Database	Agent XPs	Instance Configuration	0	0	Enable or disable Agent XPs
Database	allow filesystem enumeration	Instance Configuration	0	0	Allow enumeration
Database	allow polybase export	Instance Configuration	0	0	Allow INSERT into polybase tables
Database	allow updates	Instance Configuration	0	0	Allow updates to system tables
Database	automatic soft-NUMA disabled	Instance Configuration	0	0	Automatic soft-NUMA disabled
Database	backup checksum default	Instance Configuration	0	0	Enable checksum on backup
Database	backup compression default	Instance Configuration	0	0	Enable compression
Database	blocked process threshold (s)	Instance Configuration	20	20	Blocked process reconnection threshold
Database	BuildClrVersion	Instance Property			
Database	c2 audit mode	Instance Configuration	0	0	c2 audit mode
Database	clr enabled	Instance Configuration	1	1	CLR user code execution
Database	clr strict security	Instance Configuration	1	1	CLR strict security
Database	Collation	Instance Property	SQL_Latin1_General_CI_AS	SQL_Latin1_General_CI_AS	Type of collation used
Database	CollationID	Instance Property	872468488	872468488	
Database	column encryption enclave type	Instance Configuration	0	0	
Database	common criteria compliance enabled	Instance Configuration	0	0	Common Criteria compliance

## Performance

The Performance tab will display metrics pertaining to SQL Server:

## System Tools

---

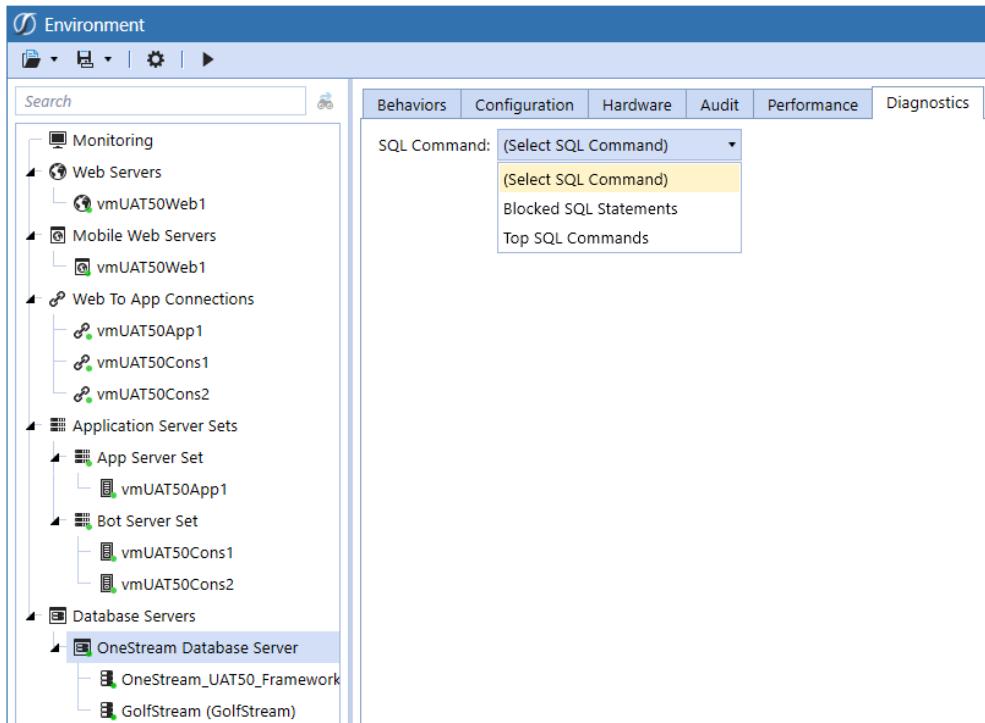


## Diagnostics

The Diagnostics tab allows the user to run SQL diagnostic commands to determine performance issue on the database instance.

## System Tools

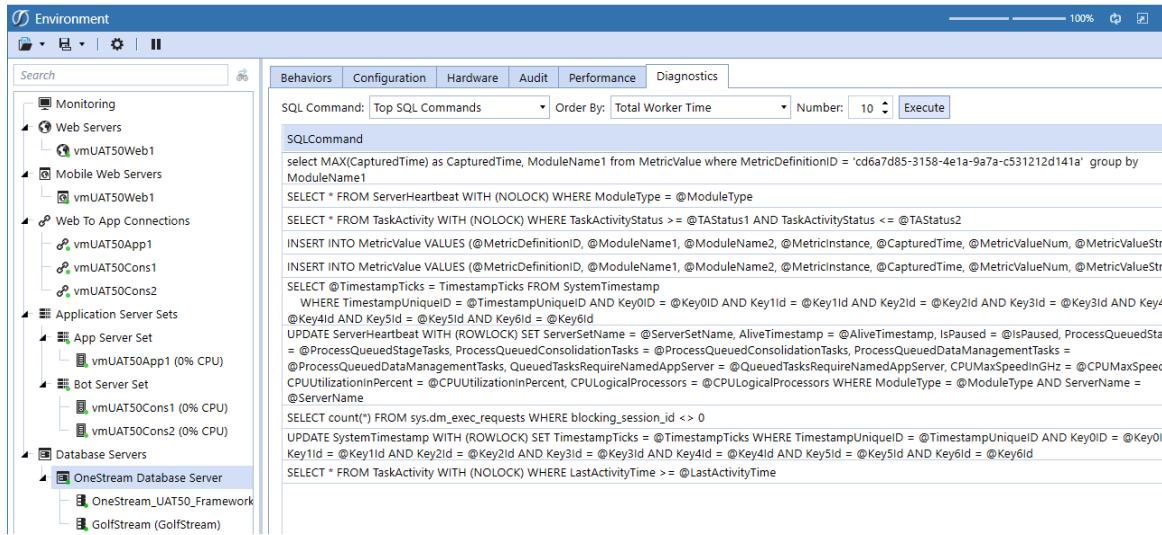
---



- SQL Deadlock information.
- The Deadlock SQL command will list out any deadlocks if any on the SQL server instance.
- Top SQL Commands will list out the top number of SQL using the order select by the user (Total Logical Reads, Total Logical Writes, Total Worker Time)

## System Tools

---



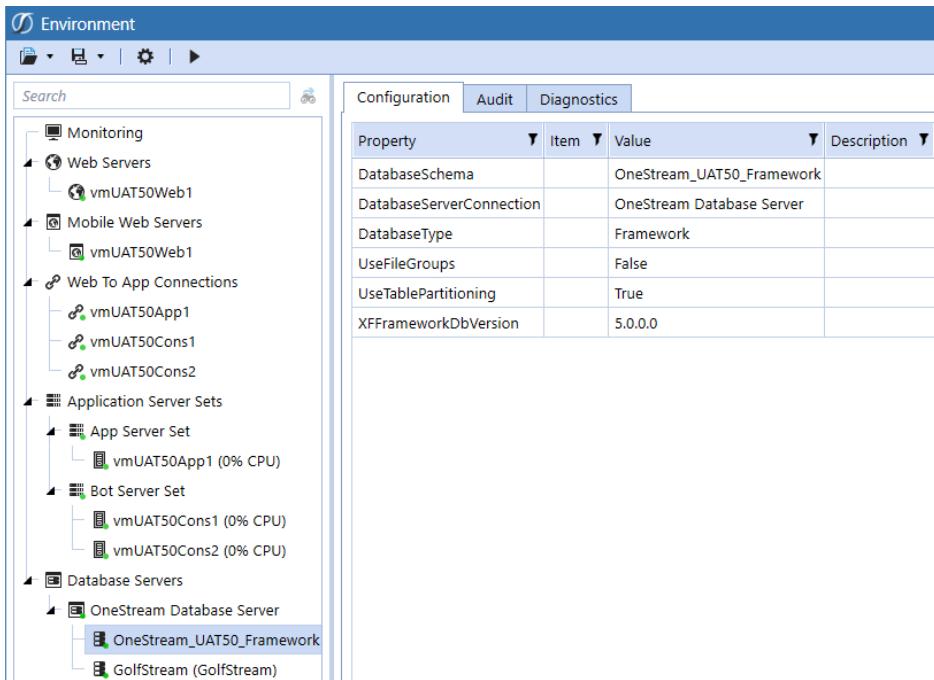
# OneStream Database Servers (Schema Items)

This section will list all the database schemas that this connection is pointing to. Each schema has its own configurations, audit and diagnostic tabs.

## Configuration

The Configuration tab contains the application-specific information:

## System Tools

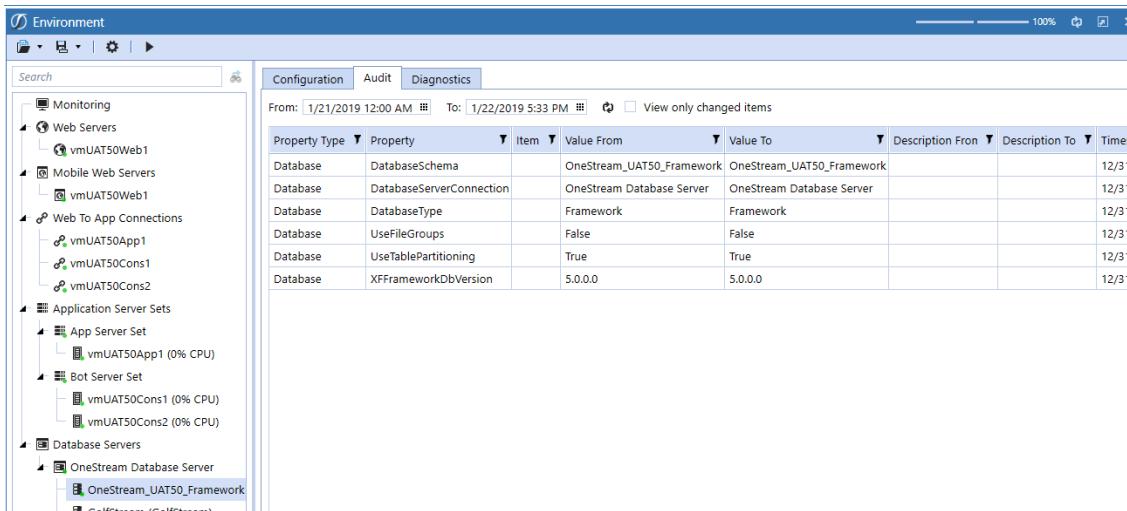


The screenshot shows the System Tools interface with the 'Configuration' tab selected. On the left is a tree view of application components, including Web Servers, Mobile Web Servers, Web To App Connections, Application Server Sets, and Database Servers. Under Database Servers, 'OneStream Database Server' is expanded to show 'OneStream\_UAT50\_Framework' and 'GolfStream (GolfStream)'. The main pane displays a table of configuration properties:

Property	Item	Value	Description
DatabaseSchema		OneStream_UAT50_Framework	
DatabaseServerConnection		OneStream Database Server	
DatabaseType		Framework	
UseFileGroups		False	
UseTablePartitioning		True	
XFFrameworkDbVersion		5.0.0	

## Audit

The audit tab will show if any of the application configuration has been changed.



The screenshot shows the System Tools interface with the 'Audit' tab selected. The date range is set from 1/21/2019 12:00 AM to 1/22/2019 5:33 PM. The main pane displays a table of audit entries for configuration changes:

Property Type	Property	Item	Value From	Value To	Description From	Description To	Time
Database	DatabaseSchema		OneStream_UAT50_Framework	OneStream_UAT50_Framework			12/3
Database	DatabaseServerConnection		OneStream Database Server	OneStream Database Server			12/3
Database	DatabaseType		Framework	Framework			12/3
Database	UseFileGroups		False	False			12/3
Database	UseTablePartitioning		True	True			12/3
Database	XFFrameworkDbVersion		5.0.0	5.0.0			12/3

# Diagnostic

Diagnostics tab will show a report of current schema table fragmentation. Sample of table fragmentation on the framework DB showing all tables beyond 70% fragmented:

The screenshot shows the 'Environment' section of the System Tools. On the left is a tree view of monitored resources under 'Monitoring'. Under 'Web Servers', there are 'vmUAT50Web1' and 'vmUAT50Web2'. Under 'Mobile Web Servers', there is 'vmUAT50Web1'. Under 'Web To App Connections', there are 'vmUAT50App1', 'vmUAT50Cons1', and 'vmUAT50Cons2'. Under 'Application Server Sets', there are 'App Server Set' (with 'vmUAT50App1 (0% CPU)'), 'Bot Server Set' (with 'vmUAT50Cons1 (0% CPU)' and 'vmUAT50Cons2 (0% CPU)'), and 'Database Servers' (with 'OneStream Database Server' containing 'OneStream\_UAT50\_Framework' and 'GolfStream (GolfStream)'). On the right, the 'Diagnostics' tab is selected. It contains a search bar, a dropdown for 'SQL Command' set to 'Table Fragmentation', a 'Fragmentation Threshold' set to '70', and an 'Execute' button. Below these are two tables: one for 'TableName' and 'IndexName' (e.g., MetricValue, TaskActivityStep, ServerConfigHistory, etc.) and another for 'IndexType' and 'PercentFragmented' (e.g., CLUSTERED INDEX, NONCLUSTERED INDEX, etc.). The fragmentation threshold of 70 is highlighted in blue.

TableName	IndexName	IndexType	PercentFragmented
MetricValue	PK_MetricValue	CLUSTERED INDEX	98.9310317523161
TaskActivityStep	PK_TaskActivityStep	CLUSTERED INDEX	98.0392156862745
ServerConfigHistory	PK_ServerConfigHistory	CLUSTERED INDEX	97.2602739726028
ServerConfigHistory	IX_ServerConfigHistory2	NONCLUSTERED INDEX	96.5909090909091
ServerConfig	IX_ServerConfig1	NONCLUSTERED INDEX	95.8333333333334
TaskActivity	PK_TaskActivity	CLUSTERED INDEX	95.1612903225806
ServerConfigHistory	IX_ServerConfigHistory1	NONCLUSTERED INDEX	95
TaskActivityStep	IX_TaskActivityStep0	NONCLUSTERED INDEX	94.4444444444444
UserLogonActivity	PK_UserLogonActivity	CLUSTERED INDEX	91.6666666666666
BRSnippet	IX_BRSnippet0	NONCLUSTERED INDEX	90.9090909090909
DashboardAdapter	PK_DashboardAdapter	CLUSTERED INDEX	90.9090909090909
ServerConfig	IX_ServerConfig0	NONCLUSTERED INDEX	90.9090909090909
TaskActivity	IX_TaskActivity1	NONCLUSTERED INDEX	90
TaskActivity	IX_TaskActivity3	NONCLUSTERED INDEX	87.5
ServerConfig	PK_ServerConfig	CLUSTERED INDEX	86.1111111111111
BRSnippet	PK_BRSnippet	CLUSTERED INDEX	85.7142857142857
AuditDashboardComponent	PK_AuditDashboardComponent	CLUSTERED INDEX	85
BRSnippet	IX_BRSnippet1	NONCLUSTERED INDEX	84.6153846153846
ServerConfigHistory	IX_ServerConfigHistory0	NONCLUSTERED INDEX	83.5051546391753
TaskActivity	IX_TaskActivity0	NONCLUSTERED INDEX	83.3333333333334
AuditDashboardAdapter	PK_AuditDashboardAdapter	CLUSTERED INDEX	80
DashboardComponent	PK_DashboardComponent	CLUSTERED INDEX	75
MetricDefinition	IX_MetricDefinition1	NONCLUSTERED INDEX	75

# Azure Configurations

(Azure Only or if Azure Elastic Pool Being Used)

## Azure Subscription Settings

The Azure Subscription Settings must be filled in, as they are used to login and retrieve Azure settings and data. This section will be populated when using Azure Elastic Pool or if using Scale Sets (which is a feature that will be available in a future release).

## System Tools

---

Application Server Azure Subscription Settings	
OneStream XF Azure	
Azure Subscription ID	015700f5-98b5-41b1-b739-1c7ab1
Azure Tenant ID	8deaecc4-beb7-482d-ae8c-d4bae
Azure Client ID	8173b323-cb92-42bc-a5cc-5eca0
Azure Secret Key	x5MXsCth3x8gV6L+X8s79fek7jU/
Azure App Insights Instrumentation Key	adcd11bc-37dd-4e4c-9a3a-da91e
Application Server Configuration Settings	

## Environment Monitoring

The Environment Monitoring section configures how and how often metrics are collected. What is available for metrics or monitoring is based on configurations made in the OneStream Server Configuration Utility. See Installation and Configuration Guide.

Environment Monitoring	
URL for the Automatic Recycle Service	<a href="https://*:50002/OneStreamMgmt/MgmtService.svc">https://*:50002/OneStreamMgmt/MgmtService.svc</a>
Number of Running Hours Before Automatic Recycle	24
Start Hour for Automatic Recycle (0 to 24 UTC)	5
End Hour for Automatic Recycle (0 to 24 UTC)	7
Maximum Number of Minutes to Pause Before Automatic Recycle	30
Active Check Update Interval (seconds)	60
SQL Blocking Timeout Interval (minutes)	5
Metric Update Interval (seconds)	30
Server Heartbeat Update Interval (seconds)	10
Collect Environment CPU Metrics	Always
Collect Environment Task Metrics	Always
Collect Environment Logon Metrics	Always
Collect Server Set CPU Metrics	Never
Collect Server Set Task Metrics	Never
Collect Server Disk Metrics	Never
Collect Server Memory Metrics	Never
Collect Server Network Card Metrics	Never
Collect SQL CPU Metrics	Never
Collect SQL Page Metrics	Never
Collect SQL Memory Metrics	Never
Collect SQL Connection Metrics	Never
Collect SQL Query Metrics	Never
Collect SQL File Metrics	Never
Collect SQL Elastic Pool CPU Metrics	Never
Collect SQL Elastic Pool DTU Metrics	Never
Collect SQL Elastic Pool Storage Metrics	Never
Collect SQL Elastic Pool Workload Metrics	Never
Number Past Metric Readings for SQL Blocking	5
Fragmentation Iteration Count	600
Fragmentation Percent Threshold	90
Detailed Logging	False
Number of Hours to Retain Offline Servers	1
Number of Days to Retain Server Config History	30
Number of Days to Retain Metrics	30

### URL for the Automatic Recycle Service

## System Tools

---

Used to specify the address of the recycle management service. The protocol for the address should be set to however the service is deployed (https or http) and the port (default is 50002). The asterisk will force the service to use the fully qualified domain name of the executing server.

### **Number of Running Hours Before Automatic Recycle**

Used to specify how often the app servers should be restarted. The default is 24.0. Use 0.0 to turn off auto recycling. Fractional numbers may be used in this setting.

### **Start Hour for Automatic Recycle (0 to 24 UTC)**

Used to specify the time period within each day that the server can be recycled. This setting is only used if Number of Running Hours Before Automatic Recycle is set to 24.0. The default setting is 5. With that value, the application servers will be recycled between 5:00am UTC and what the setting for the End Hour for Automatic Recycle is set to.

### **End Hour for Automatic Recycle (0 to 24 UTC)**

Used to specify the time period within each day that the server can be recycled. This setting is only used if Number of Running Hours Before Automatic Recycle is set to 24.0. The default setting is 7. With that value, the application servers will be recycled between 7:00am UTC and what the setting for the Start Hour for Automatic Recycle is set to.

### **Maximum Number of Minutes to Pause Before Automatic Recycle**

Used to specify the maximum number of minutes that the server should pause its ability to run newly queued tasks before recycling. The default setting is 30. This setting is the time period that the server will use to allow previously started tasks to finish. If there are no previously running tasks on the application server, the setting is ignored.

### **Active Check Update Interval (seconds)**

How often system checks for system monitoring (i.e.; deadlocks ...)

### **Metric Update Interval (seconds)**

How often system checks for metric updates.

### **Server Heartbeat Update Interval (seconds)**

How often system updates that this server is alive.

### **Collect Environment CPU Metrics**

How often to collect environment CPU metrics.

### **Collect Environment Task Metrics**

How often to collect environment task metrics (i.e.; running tasks, Queued Tasks ...)

## System Tools

---

### **Collect Environment Login Metrics**

How often to collect environment user login metrics.

### **Collect Server Set CPU Metrics**

How often to collect server set CPU metrics.

### **Collect Server Set Task Metrics**

How often to collect server set task metrics (i.e.; running tasks, Queued Tasks ...).

### **Collect Server Disk Metrics**

How often to collect server disk metrics (i.e.; Average Disk read/write per sec...).

### **Collect Server Memory Metrics**

How often to collect server memory metrics (i.e.; Available MBytes...).

### **Collect Server Network Card Metrics**

How often to collect server network card metrics.

### **Collect SQL CPU Metrics**

How often to collect SQL server CPU metrics.

### **Collect SQL Page Metrics**

How often to collect SQL server Page caching metrics (i.e.; Page Life Expectancy...).

### **Collect SQL Memory Metrics**

How often to collect SQL server CPU metrics.

### **Collect SQL Connection Metrics**

How often to collect SQL server connection metrics (i.e.; Number of connections...).

### **Collect SQL Query Metrics**

How often to collect SQL server Query metrics (i.e.; Number of Deletes/Inserts...).

### **Collect SQL File Metrics**

How often to collect SQL server File growth metrics.

### **Collect SQL Elastic Pool CPU Metrics**

How often to collect SQL server Elastic Pool CPU metrics (i.e.; Number of connections...).

### **Collect SQL Elastic Pool DTU Metrics**

How often to collect SQL server Elastic Pool DTU metrics (i.e.; Number of connections...).

### **Collect SQL Elastic Pool Storage Metrics**

How often to collect SQL server Elastic Pool Storage metrics (i.e.; disk storage usage...).

### **Collect SQL Elastic Pool Workload Metrics**

How often to collect SQL server Elastic Pool Workload metrics.

### **Number Past Metric Readings for SQL Blocking**

Used to select the past number of metric values for blocking analysis.

## System Tools

---

### Fragmentation Iteration Count

Used for fragmentation check, every 10 hours if the ActiveCheckUpdateIntervalInSec is set to 60.

### Fragmentation Percent Threshold

Used for fragmentation threshold check in percent.

### Detailed Logging

If true, then log whenever we enter and exit the metric collection and the Active System check.

### Number Hours To Retain Offline Servers

Remove offline servers from the heartbeat table after certain number of hours.

## Task Load Balancing

Task Load Balancing	
Maximum Queue Processing Interval (seconds)	10
Maximum Average CPU Utilization	80
Maximum Queued Time (minutes)	30
Number of Past Metric Readings for Average CPU	1
Task Logging	False
Detailed Logging	False

### Maximum Queued Processing Interval (seconds)

How often the queuing thread checks for new tasks to be run.

### Number Past Metric Reading For Analysis

How many metric readings to retrieve for analyzing server demand.

### Maximum Queued Time (minutes)

Maximum time to wait for a task to run before it is cancelled.

### Maximum Average CPU Utilization

Maximum average CPU utilization before we determine that a server can't take a task.

### Task Logging Only

If true will only logs picked up tasks.

### Detailed Logging

If true, then log whenever we enter and exit the Task Load Balancing function and what was run.

# Database Server Connection

The Azure section in the Database Connection Settings needs to be completed only if Azure database and/or Azure Elastic pool is used.

The screenshot shows the 'OneStream Database Azure Server properties' dialog. On the left, a tree view under 'Members:' shows three items: '0 OneStream Database Server', '1 SQL Server 2017', and '2 OneStream Database Azure Server'. The third item is selected and highlighted in blue. On the right, the 'Azure Database Connection Settings' section is expanded, displaying the following configuration:

Azure Elastic Pool Max DTU Setting	1600
Azure Elastic Pool Min DTU Setting	200
Azure Elastic Pool Name	TM_DB_ElasticPool
Azure Resource Group	TM_DB_Resource
Azure Service Level Objective	S2
Azure SQL Edition	Standard
Azure SQL Scaling Type	Manual
Azure SQL Server Name	tmdbsrv
Azure SQL Storage Max Size	10
Azure SQL System Business Rule Name	testme

Sample Azure database connection settings:

### Azure Elastic Pool Max DTU Setting

This is a fail-safe setting that the user can't set the DTU setting above this point.

### Azure Elastic Pool Min DTU Setting

This is a fail-safe setting that the user can't set the DTU setting below this point.

### Elastic Pool Name

The name of the elastic pool used with this database connection.

### Azure Resource Group

The resource group name that the elastic pool is in.

### Azure Service Level Objective

The service level used. This setting is used to create application on Azure.

### Azure SQL Edition

The Azure SQL Server edition used.

### Azure SQL Scaling Type

Manual, Business Rule or ManualAndBusiness Rule. This is used to set the type of scaling that is used to Scale Out or Scale In the SQL Server eDTUs. The default setting should be set to Not Used.

### Azure SQL Server Name

The name of the SQL Server. This setting is used to create application on Azure.

### Azure SQL Storage Max Size

This is used to specify the database storage size when creating a database on AZURE.

### Azure SQL System Business Rule Name

If SQL Scaling Type is set to BusinessRule, this setting must be set to a business rule that is used to scale in/out. The Environment metrics and the database metrics are passed to this rule to properly determine the eDTU scaling.

## Azure Server Sets Settings

This setting is used to set up the used scale sets for the different application servers:

Members:	Standard properties:																																																	
0 Standard	<table border="1"><thead><tr><th> </th><th>A Z</th><th> </th></tr></thead><tbody><tr><td colspan="3">Azure</td></tr><tr><td>Azure Resource Group Name</td><td>False</td></tr><tr><td>Azure Scale Set Name</td><td>10</td></tr><tr><td>Can Start or Stop Servers</td><td>1</td></tr><tr><td>Maximum Capacity</td><td>Not Used</td></tr><tr><td>Minimum Capacity</td><td></td></tr><tr><td>Scaling Type</td><td></td></tr><tr><td>System Business Rule Name</td><td></td></tr><tr><td colspan="3">Behaviors</td></tr><tr><td>Process Queued Consolidation Tasks</td><td>True</td></tr><tr><td>Process Queued Data Management Tasks</td><td>True</td></tr><tr><td>Process Queued Stage Tasks</td><td>True</td></tr><tr><td>Queued Tasks Require Named Application Server</td><td>False</td></tr><tr><td colspan="3">General</td></tr><tr><td>Name</td><td>Standard</td></tr><tr><td>Server Names for Standard Server Sets (Supports *? wildcards)</td><td>(Collection)</td></tr><tr><td>Server Set Provider</td><td>Standard</td></tr><tr><td colspan="3">Processing</td></tr><tr><td>Can Change Queueing Options on Servers</td><td>True</td></tr><tr><td>Can Pause or Resume Servers</td><td>True</td></tr><tr><td>Can Recycle App Pool on Servers</td><td>True</td></tr></tbody></table>		A Z		Azure			Azure Resource Group Name	False	Azure Scale Set Name	10	Can Start or Stop Servers	1	Maximum Capacity	Not Used	Minimum Capacity		Scaling Type		System Business Rule Name		Behaviors			Process Queued Consolidation Tasks	True	Process Queued Data Management Tasks	True	Process Queued Stage Tasks	True	Queued Tasks Require Named Application Server	False	General			Name	Standard	Server Names for Standard Server Sets (Supports *? wildcards)	(Collection)	Server Set Provider	Standard	Processing			Can Change Queueing Options on Servers	True	Can Pause or Resume Servers	True	Can Recycle App Pool on Servers	True
	A Z																																																	
Azure																																																		
Azure Resource Group Name	False																																																	
Azure Scale Set Name	10																																																	
Can Start or Stop Servers	1																																																	
Maximum Capacity	Not Used																																																	
Minimum Capacity																																																		
Scaling Type																																																		
System Business Rule Name																																																		
Behaviors																																																		
Process Queued Consolidation Tasks	True																																																	
Process Queued Data Management Tasks	True																																																	
Process Queued Stage Tasks	True																																																	
Queued Tasks Require Named Application Server	False																																																	
General																																																		
Name	Standard																																																	
Server Names for Standard Server Sets (Supports *? wildcards)	(Collection)																																																	
Server Set Provider	Standard																																																	
Processing																																																		
Can Change Queueing Options on Servers	True																																																	
Can Pause or Resume Servers	True																																																	
Can Recycle App Pool on Servers	True																																																	

The **Azure** section is used to set the Azure resource group name and the scale set name as it exists on Azure. The Scaling options are used to set the scaling capacity and the Scaling Type (e.g. Manual or BusinessRule). The System Business Rule Name must be set if the Scaling Type is set to Business Rule.

The **Behaviors** section is used determine how this scale set is used. Whether it can be called by the web server or it is used as a processing server for consolidations and Stage Load.

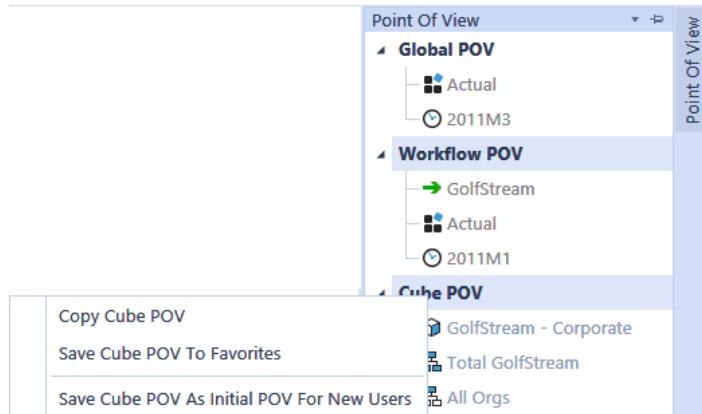
The **General** section is used to determine whether this scale set is used and who is its provider (i.e.; Azure, External ...).

# File Explorer

Documents and saved Point of View settings can be stored in the Application database.

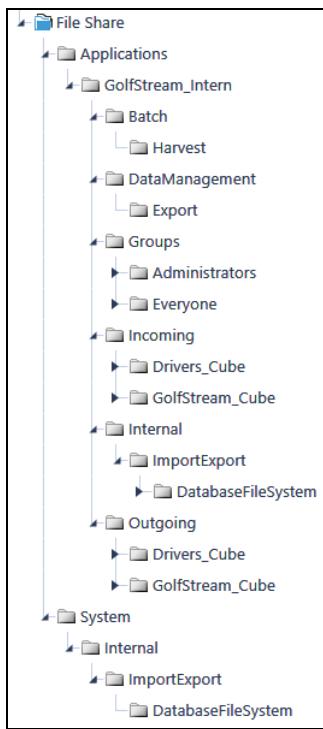
-  Create a new folder in the File Explorer
-  Upload a specific file into any folder. If a user has Administrator rights or is in the ManageFileShare security role, he/she can upload and delete files in the Harvest folder.
-  Import or Extract from the Application database and the System database. These are sent to the main product folder created during the initial configuration (e.g. C:\OneStream).

A common Cube Point of View can be stored for continued reuse by multiple users. The POV can be stored for all users in the Public area or for the specific user in the User directory. To create a saved, named POV, right click the Cube POV node under Point of View within the Context Pane:



# File Share

File Share is a self-service directory that temporarily stores files before they are imported into the Application or System Database. Files stored in an Import folder are accessible only through that Workflow Profile.



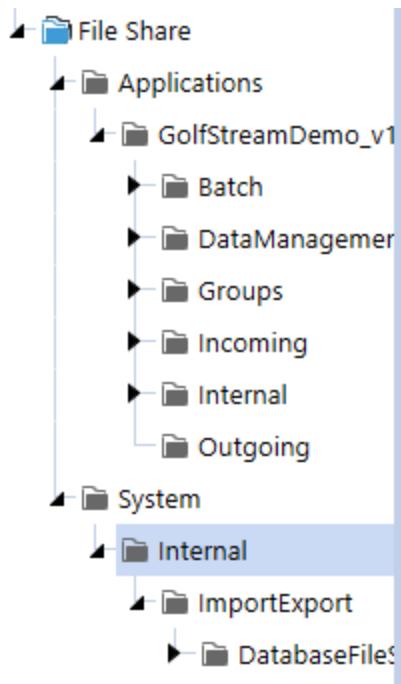
## Permissions

Administrators grant permissions for folders and individual files by adding a user to the associated group:

- **Access:** View content
- **Maintenance:** View, edit, and delete content

## Content Folder

Both the Application and System folders contain an auto-generated folder named Content, intended to store files larger than 300 MB.



## Permissions

The Application-specific and System Content folders are managed by the following system security roles:

- **Administrator** and **ManageFileShare** Roles have full rights.
- Non-Administrators can be assigned rights to modify, access or have limited rights to the Content folders.

## Uploads

By default, when this flag is set to **True**, authorized users can upload or edit files or folders in the File Share using the OneStream File Explorer (required to use Content folder). When set to **False**, no users will have upload, edit file or edit folder access to the File Share using the OneStream File Explorer. However, normal access rules will apply for browsing files and folders in the File System section.

Users will receive a Security error when attempting to write or edit the files and folders when attempting to use API or non File Explorer methods.

# Supported File Sizes

The OneStream File Explorer can be used as an application interface to the File Explorer's Content Folders. The supported file size varies by interface

### Windows Application

- Uploads (Applications and System): Up to 300 MB
- Uploads (Content): Up to 2 GB
- Downloads (Application and System): Up to 2 GB
- Downloads (Content): Up to 2 GB

# Whitelist File Extensions

Whitelist File Extensions provide a method of identifying which kinds of documents, according to their file extensions, are allowed to be saved into the OneStream application File Explorer. Whitelisting extensions helps to alleviate the risk of uploading malicious file types into the application.

## Overview

OneStream lets you save documents or upload documents from your device into the application. These documents are stored in one of three file storage root folders:

- Application Database: Displays stored documents for the current application only.
- System Database: Displays stored documents for the entire system without affecting the current application.
- File Share: This is a self-service directory that temporarily stores files before they are imported into the Application or System Database.

File Explorer is the product tool inside the Windows Application that lets you browse saved documents or upload new ones.

For example, a spreadsheet can be saved to any of the three root folders. A Cube POV can be saved to either the application or system folders.

## System Tools

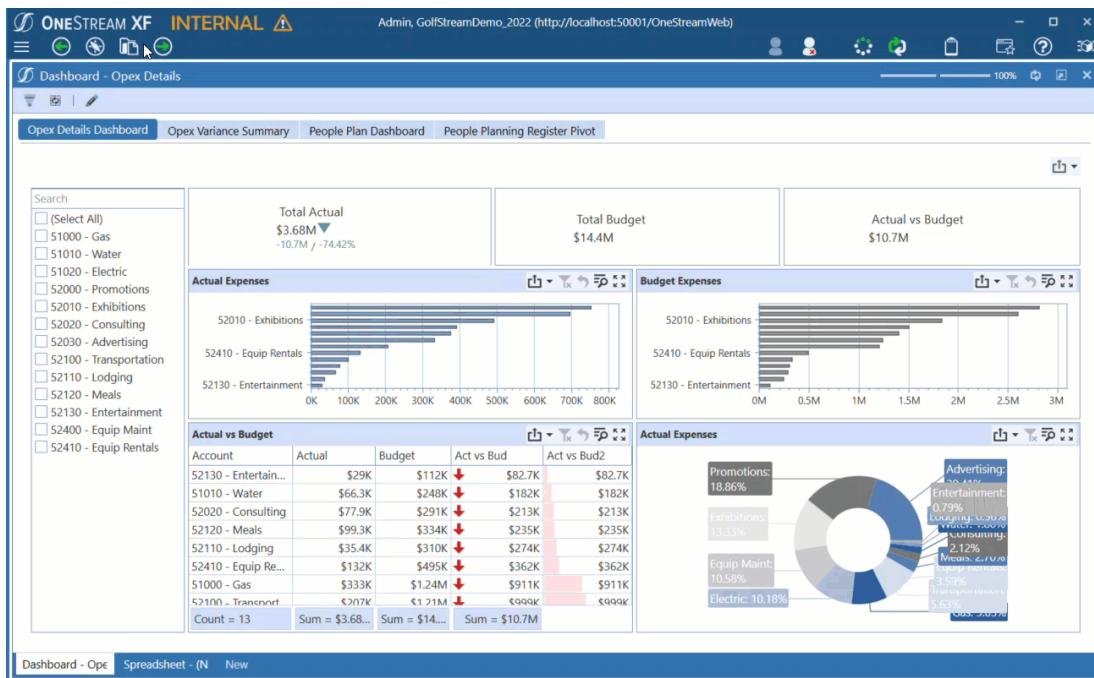
The screenshot shows the OneStream XF INTERNAL application's spreadsheet module. The ribbon menu includes File, Home, Insert, Page Layout, Formulas, Data, Review, View, and OneStream. The Home tab is selected. The main area displays a table with columns 'Location' and 'Count'. The 'Count' column for Mannheim, Frankfurt, and Heidelberg is 100, 200, and 300 respectively. The sidebar on the right contains sections for 'Quick View' (with options like Cube (Default POV), Entity (Default POV), Parent (Default POV), etc.), 'Column Dimensions', and 'Row Dimensions'.

A Cube POV can be saved to either the application or system folders.

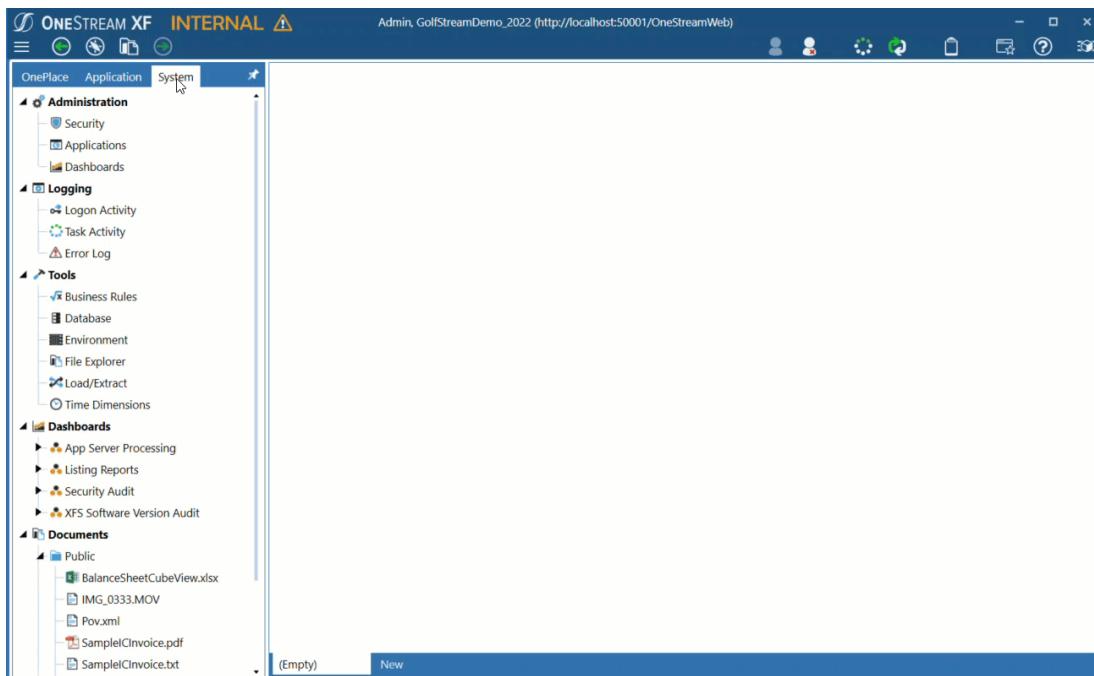
The screenshot shows the OneStream XF INTERNAL application's Opex Details Dashboard. The dashboard includes several visualizations: a bar chart for 'Actual Expenses' (Exhibitions, Equip Rentals, Entertainment), a bar chart for 'Budget Expenses' (Exhibitions, Equip Rentals, Entertainment), a table for 'Actual vs Budget' (listing accounts like 5210 - Exhibitions, 5240 - Equip Maint, etc.), and a donut chart for 'Actual Expenses' (Promotions, Advertising, Equip Maint, Electric, Meals, Lodging, Transportat, Exhibitions, Consulting). The sidebar on the right is titled 'Point Of View' and lists 'Global POV', 'Workflow POV', and 'Cube POV' (with 'GolfStream - Corporate' selected).

File Explorer is available from the main toolbar.

## System Tools



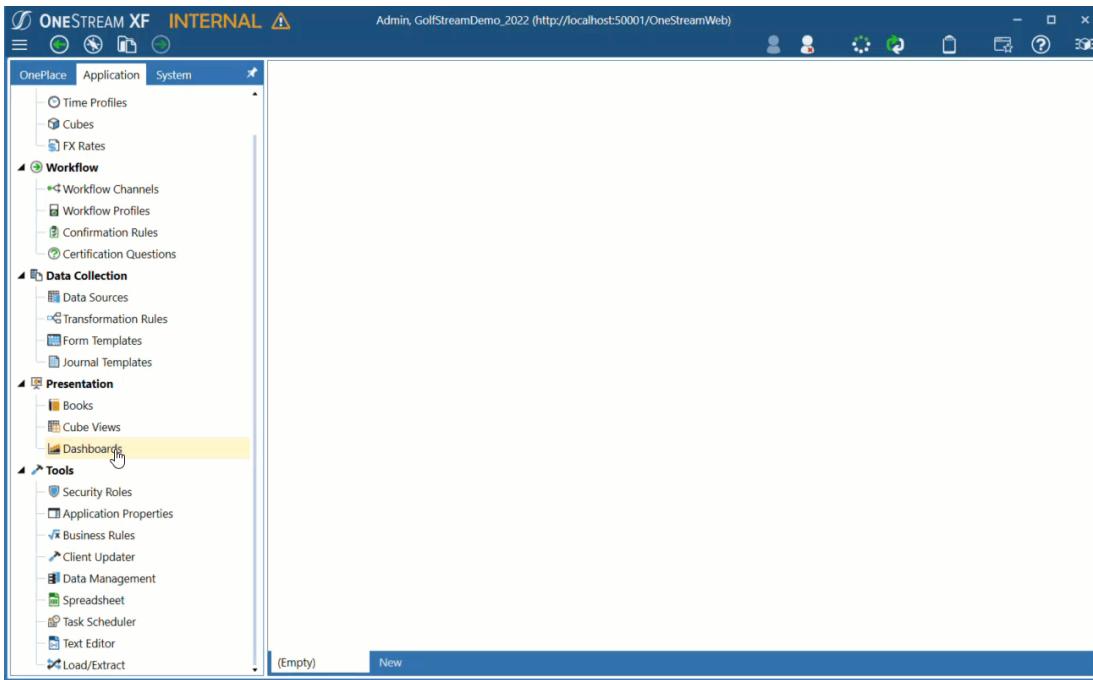
File Explorer is also available on the System tab.



The Dashboard component "File Viewer" also launches the File Explorer component.

## System Tools

---



Finally, there is a BRAPI used by marketplace solutions or custom-developed dashboards to upload documents.

## Whitelist Files Extensions Prerequisites

Standard File Explorer setup and security.

## Define the List of File Extensions

Whilelisted file extensions must be specified in the OneStream Application Server Configuration file.

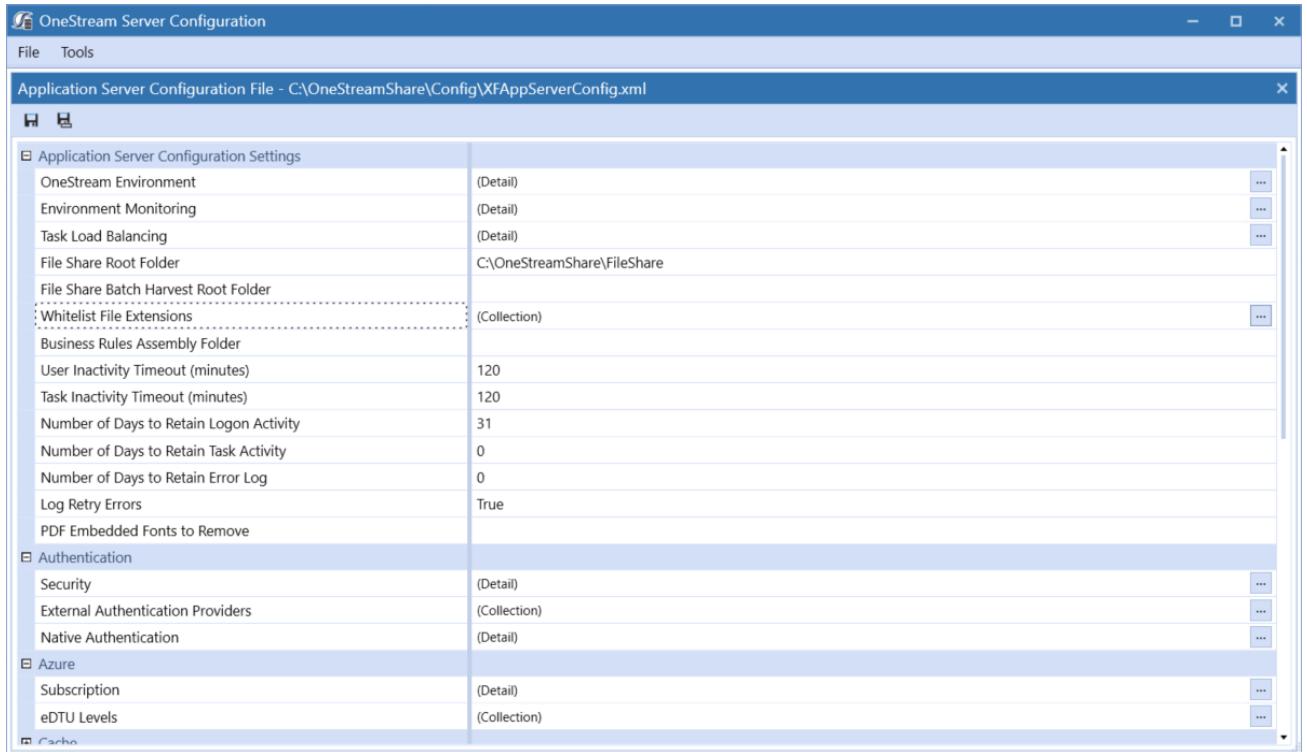
To enable the whitelist and define allowed file extensions:

1. Go to **Start > OneStream Software > OneStream Server Configuration Utility**, right-click and select **Run as Administrator**.
2. On the File menu, select **Open Application Server Configuration File**.
3. Browse to the location of your XFAppServerConfigFile (typically located at C:\OneStreamShare\Config), select the application config file, and click **Open**.

## System Tools

---

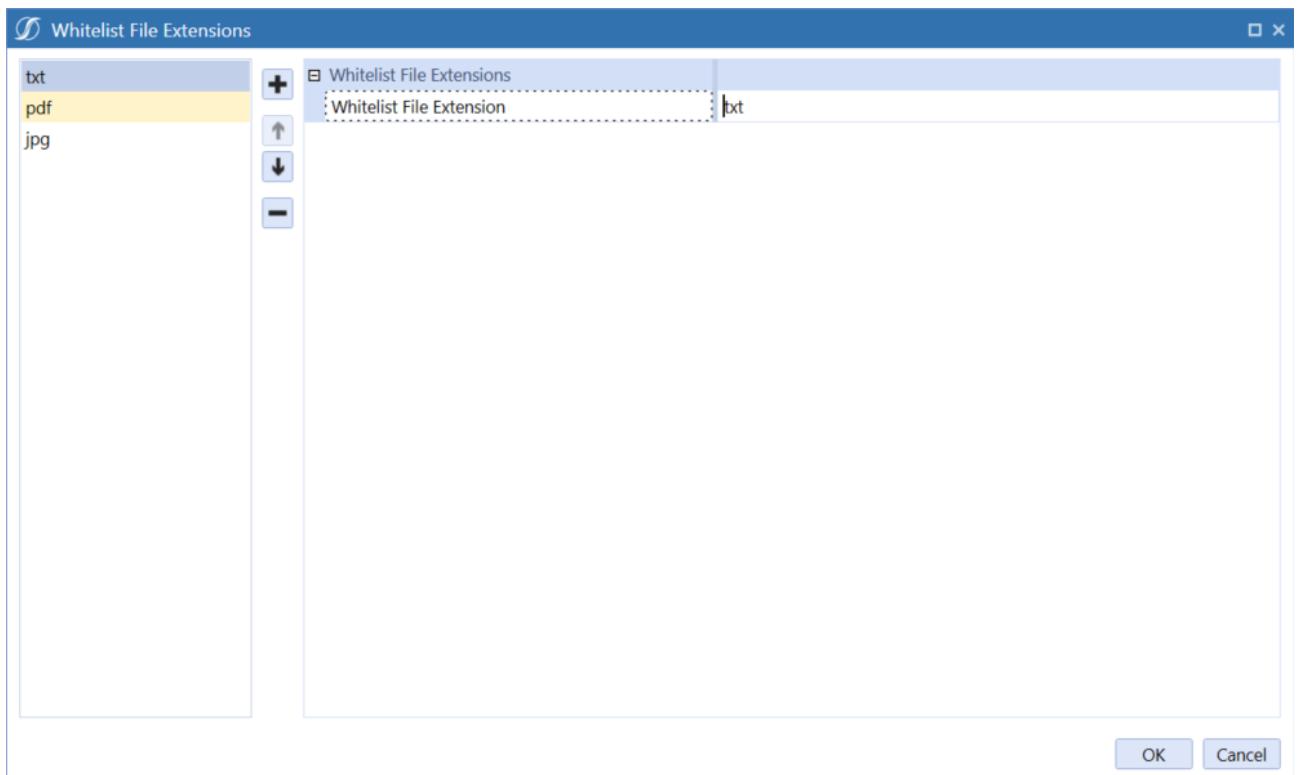
4. In the Application Server Configuration Settings section, locate Whitelist File Extensions and click the ellipses (...) on the far-right of the field.



5. In the dialog box, click the plus sign (+) then type the whitelisted file extension. For example, .txt.

## System Tools

---



6. Continue adding other whitelisted file extensions.
7. When you are finished, click **OK**.
8. Click **Save** to save the changes to the Application Server Configuration File.
9. Restart Internet Information Service (IIS).

**NOTE:** Cloud customers should contact support to have this configuration change made for them.

Also, if you enter any of the following characters, they will be automatically removed. For example, typing .txt (with a period) becomes txt.

/	^	=	'
(empty space)	&	\	[

!	*	?	]
@	(	<	.
#	)	>	
\$	-	~	
%	+	'	

## Add Documents using File Explorer

To add documents using File Explorer:

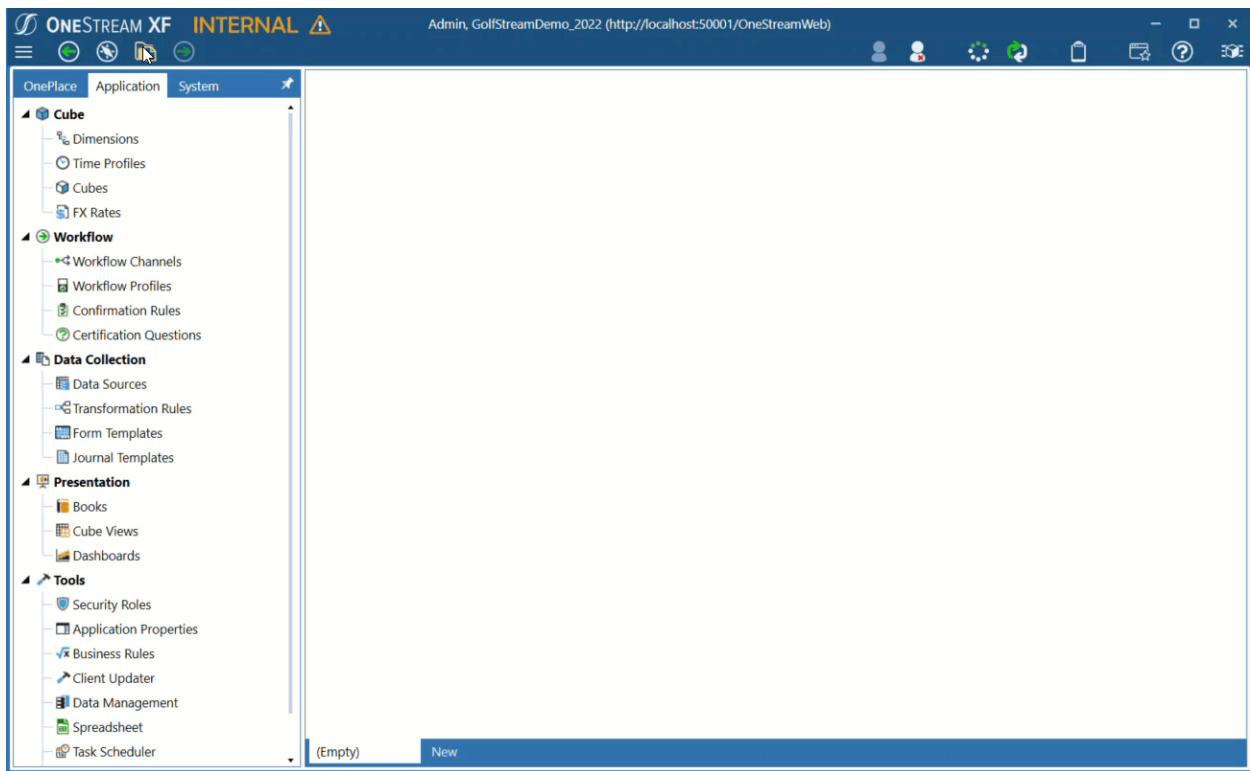
1. Navigate to any File Explorer component inside the Windows Application.
2. Select **Upload File**.
3. Browse to the file you would like to upload and click **OK**.

If the extension of the file is whitelisted, it will be uploaded. If not, you will receive an error

## System Tools

---

message.



## Troubleshooting

After a whitelist is defined, new files whose extension is not in the whitelist will be restricted from uploading.

If you upload a file that contains an exploit, malware, or a malicious script or macro, other users can download the file and infect their machine.

## Conclusion

Whitelisting a file extension provides a means of allowing only certain types of documents to be saved in the File Explorer. This helps to alleviate the possibility of introducing malicious file types into the system.

# Application Database/System Database

The Application Database displays stored documents for the current Application only. The System Database displays stored documents for the entire system without affecting the current application.

## Documents

### Public

Files available to everyone with access.

### Security Access

Enables a user to access an object and read the content.

### Maintenance (Group)

Enables a user to view an object and create, modify and delete objects in Groups. If in a Maintenance Group, a user does not need to be in the Access Group. The Maintenance Group also controls user profile contents.

### Users

Private documents only available to named users.

### Internal

Files used internally that cannot be modified.

# Load/Extract System Artifacts

You can import and export sections of the System using an XML Format.

**TIP:** Only System Administrators have access to this portion of the tool.



### Extract

Choose an item from the drop-down list, click the Extract icon to start the extract process, then name the output file.



### Load

After browsing to the file, click the Load icon to initiate the process.



### Extract and Edit

This option is available for all extracts and allows the ability for the end user to edit the XML file as needed.

The following items can be extracted:

#### Security

This covers System Roles, Users, Security Groups, and Exclusion Groups which can be found under the System Tab |Administration| Security. The screen will display Extract Unique IDs. If this box is checked, the unique IDs OneStream assigns to each user will be extracted with the security. When moving security changes from one OneStream environment to another, trying to load the security with the unique IDs into the destination environment can result in an error if some of the records already exist. If this is the case, uncheck this box and extract without the unique IDs. Items to Extract also displays and by default All is chosen. Choose specific items in each section or turn the sections on or off.

#### System Dashboards

This has multiple sections such as Maintenance Units and Profiles. Under Maintenance Units, the Groups, Dashboard Components, Dashboard Adapters, and Dashboard Parameters all go together. System Dashboards can be found under the System Tab| Administration| Dashboards. The screen will display Items To Extract and by default All is chosen. Choose specific items in each section or turn the sections on or off.

#### Error Log

This covers the Error Log found under System Tab| Logging| Error Log. The screen will display an Extract All Items check box along with a Start Time and End Time. If the check box is notchecked,

choose the Start and End Time by clicking . Click on the extract button and define where the file will be saved.

#### Task Activity

This covers the Task Activity found under System Tab > Logging > Task Activity. The screen will display an Extract All Items check box along with a Start Time and End Time. If the check box is

not checked, choose the Start and End Time by clicking . Click on the extract button and define where the file will be saved.

#### Logon Activity

This covers the Logon Activity found under System Tab| Logging| Logon Activity. The screen will display an Extract All Items check box along with a Start Time and End Time. If the check box is

not checked, choose the Start and End Time by clicking . Click on the extract button and define where the file will be saved.

# Time Dimensions

Applications can have a monthly or weekly Time Dimension. Time Dimension Types determine if an application uses a weekly, monthly or 12/13 period frequency, and the type of calendar (such as 445 or 454). In a new application, use a custom Time Dimension Type so users can specify the number of months in a quarter and the number of weeks in a month.

After you create a Time Dimension, an XML file is generated. New applications use this file to implement the desired Time Dimension. If a Time Dimension Type is not assigned to a new application database, the default Standard Type is used.

**NOTE:** Applications created prior to 4.1.0 can convert a monthly application to weekly application using the Database Configuration Utility. Contact Support for assistance.

## Time Dimension Types

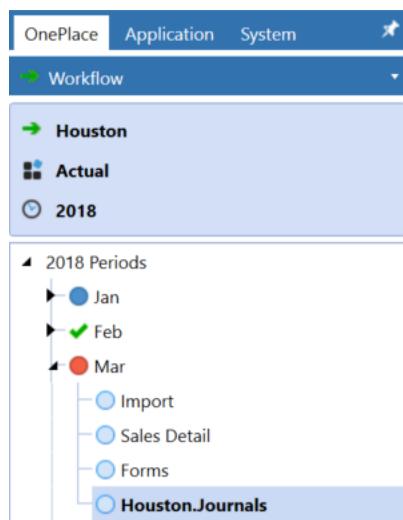
- **Time Dimension Type Standard:** Creates a Monthly Time Dimension and stores data by month in the data tables. Applications created prior to version 4.1.0 use this type.
- **StandardUsingBinaryData:** Creates a Monthly Time Dimension and stores data in a binary data table. Use this type if you may need to later convert an application to a Weekly Time Dimension.
- **M12\_3333\_W52\_445:** Creates a 12 Month, 4 quarter, 52 Week, 445 calendar Time Dimension.
- **M12\_3333\_W52\_454:** Creates a 12 Month, 4 quarter, 52 Week, 454 calendar Time Dimension.
- **M12\_3333\_W52\_544:** Creates a 12 Month, 4 quarter, 52 Week, 544 calendar Time Dimension.
- **M12\_3333\_W53\_445:** Creates a 12 Month, 4 quarter, 53 Week, 445 calendar Time Dimension.
- **M12\_3333\_W53\_454:** Creates a 12 Month, 4 quarter, 53 Week, 454 calendar Time Dimension.
- **M12\_3333\_W53\_544:** Creates a 12 Month, 4 quarter, 53 Week, 544 calendar Time Dimension.
- **Custom:** Creates a Time Dimension you can use to specify the months in a quarter and the in a month. This is only available for new applications.
- **Use Weeks:** If true, you can define a Custom Weekly Time Dimension. If False, you can create a Custom Monthly Time Dimension.
- **Vary Settings By Year:** If True, you can specify the number of weeks in a month for each year. If False, you can apply the number of weeks per month to each year.

- **M1- M16 Number of Weeks:** Specify the number of weeks per month. This coincides with the number of months per quarter, so any additional weeks required are enabled. If Vary Settings By Year property is True, select a year and customize the number of weeks in that year.

# Using OnePlace Workflow

The top of this slider configures the Workflow View for data loading: Workflow Profile, Scenario, and Year. Clicking on any of the three items displays a pop-up window to select Workflow Profile, Scenario, and Year.

A Workflow Unit is an individual period within the selected year and Scenario combination for a particular Workflow Profile. Twelve Workflow Units or periods are always displayed for selection, but only one can be selected and active at a given time. Each Workflow Origin step available must be completed as well as the collective data confirmation and certification process for a single Workflow process to be finalized.



**TIP:** The available tabs will be determined by the security level.

### Right-Click Options

Right-click a Workflow month, Workflow Input Type, or Dependent Status cell to display the following options. Not all options are available for every object.

### Status and Assigned Entities

Displays the Workflow Status of each Origin process (Import, Forms, and Journals). See Dependent Status under Certify for more details on this feature.

## Using OnePlace Workflow

---

### Audit Workflow Process

Right-click any Workflow channel. This provides an audit for every Workflow task's process including the date and time of the process, the user performing the process, how long it took for the process to complete, and any errors that occurred during the process. It also provides audit history through Lock History > Workflow Lock/Unlock.

### Lock/Lock Descendants

This will lock a particular Workflow period, or Origin process.

### Unlock/Unlock Descendants

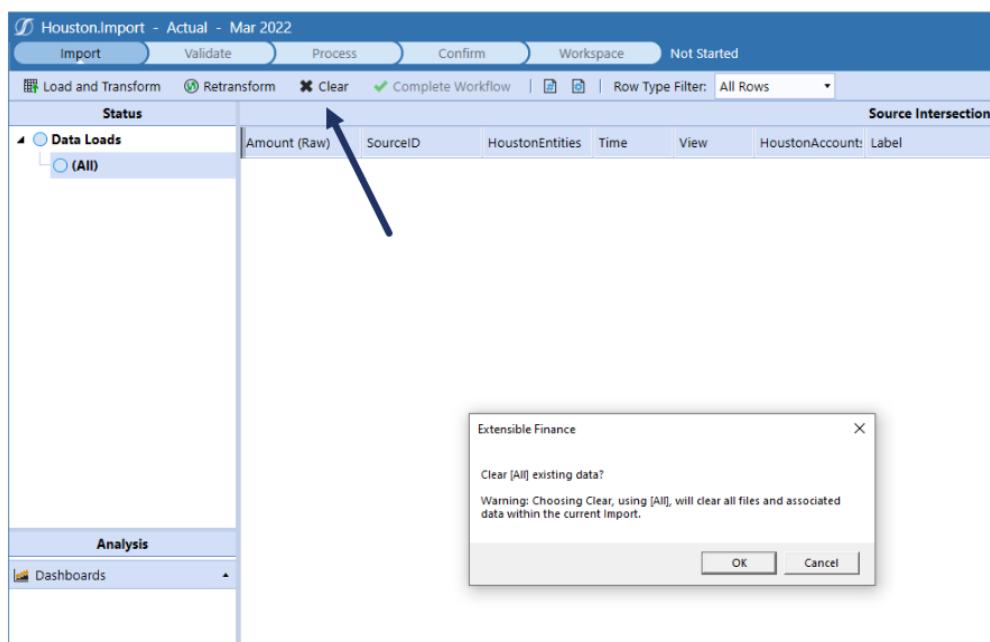
This will unlock a particular Workflow period, or Origin process.

### Edit Transformation Rules

This will navigate to the Transformation Rules screen and allow a user to fix any Transformation Rule errors that occur during the Workflow process.

### Clear All Data Loads

This will clear all files and data within the current import. If you select All, you will get a warning message before you are able to proceed.



### Clear All 'Import' Data From Cube

This will clear all imported data from the Cube, but the data still remains in the Stage.

## Using OnePlace Workflow

---

### Clear All ‘Forms’ Data From Cube

This will clear all Forms data from the Cube.

### Corporate Certification Management

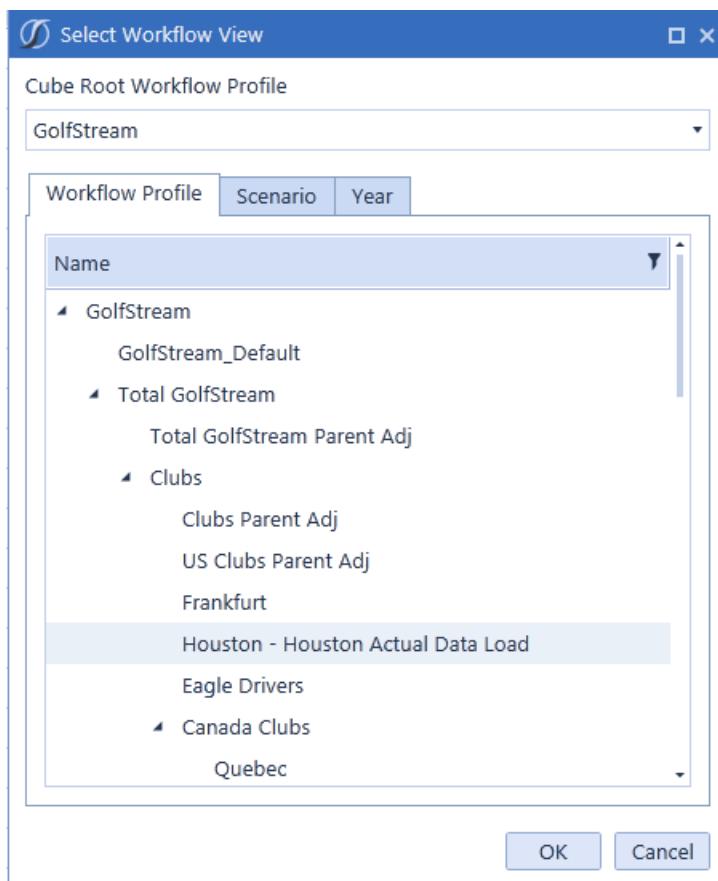
This allows a user to unlock and uncertify ancestors, or lock and certify descendants.

### Corporate Data Control

This allows a user to preserve data if changes need to be made and restore the preserved data if necessary.

### Workflow View Selection

A description can be added to a Workflow Profile during its design. Descriptions can only be added to Workflow Profiles using the Default Scenario Type. Scenario Type specific descriptions cannot be used.



# Workflow Tasks

This simplifies the data collection, consolidation, and certification process. Importing data, entering a form, making a journal entry, and signing off on reports is completed through the Workflow process.

## Import

This process step gives the end user the ability to import data into the system. This can be through a defined Data Source or a Data Connector. First click on the Import Workflow Origin located under the active month, then click Import in the task bar.



After a user clicks this icon, it will prompt him/her to either search for a file on the drive or initiate the Data Connector. The system will then import the data into the Stage engine. The file will be parsed into a clean tabular format with information on the Amounts, the Source ID, and each Dimension. Once the data is loaded successfully, the Import task will change from blue to green.

### Load Method

Upon clicking Load and Transform, a dialog will appear with four Load Options:

#### Replace

This will clear all data for the previous file that correlates with the specific Source ID and replace it with the new file's data. This can be done even if the previous data has already been loaded into the Cube. Once the file is re-loaded, the user will need to complete all Workflow Tasks and load the new data into the Cube.

#### Replace (All Time)

Replaces all Workflow Units in the selected Workflow View (if multi-period). Forces a replace of all time values in a multi-period workflow view

#### Replace Background (All Time, All Source IDs)

Replaces all Workflow Units in the selected Workflow View and all Source ID's in a background thread while the new file parse or connector execution is running. The delete is being performed while parse is being performed.

## Using OnePlace Workflow

---

**NOTE:** This load method always must always be used to delete ALL Source IDs. If the workflow uses multiple Source IDs for partial replacement during a load, this method cannot be used.

### Append

This is used when additional rows are added to a source file and need to be loaded into the Stage. This will not change any of the data already loaded for the source file, it will only add rows that were not included in the previous file load.



This is used after data has already been loaded, but some changes have been made and the calculations need to be repeated.



This will clear all loaded data from the Stage.



View Last Source File Processed for Current Workflow Profile



View Last Log File Processed for Current Workflow Profile



When this icon displays, the Global POV is enforced and data loading is limited to the Global POV. The Enforce Global POV setting is controlled under Application|Tools|Application Properties.



When this icon displays, data cannot be imported to time periods prior to the current Workflow year. The Allow Loads Before Workflow View Year setting is controlled under Application|Tools|Application Properties.



When this icon displays, data cannot be imported to time periods after the current Workflow year. The Allow Loads After Workflow View Year setting is controlled under Application|Tools|Application Properties.

# Right-Click Options

### **View Source Document**

This will open the source file imported into the application.

### **View Processing Log**

This will open a processing log and provide information on when and how the source file was imported.

### **View Transformation Rules**

This provides all mapping rules for the specific intersection

### **Drill Back**

Using a SQL Connector allows a user to drill back to a source system and show detailed records from a document, PDF, website, within the application. This option is only available if data was loaded using a Connector Data Source. For more details on this feature, see Drill Back in "Collecting Data" on page 132.

### **Export**

This will export the data into an Excel XML< CSV, Text, or HTML file

# Validate

After the data is loaded, the user will then validate the map and intersections. During the Validate step two specific actions happen. First, OneStream will check to make sure each piece of data has a map. Next, OneStream will check to make sure the combination of Dimensions can be loaded into the Cube based on the constraints defined in the system (e.g. using Intercompany). Click the green Validate task in the taskbar.



Click this icon to validate the data. If there are no errors, the Validate task will change from blue to green. If there are any errors, Validate will turn red and the errors will need to be fixed before proceeding.



### Transformation Errors

The errors will be listed by the Dimension with the error, such as a new account that was added, but not mapped. The user will be able to see the Source Value and the (Unassigned) Target Value. The system will already know the Dimension and pull up the possible targets. On the right, the user will be able to use the search box and click the funnel/filter button to find the correct account. The default will be a One-To-One Transformation Rule. For more information on these rules, see Transformation Rules in "Data Collection" on page 511.



Save any changes made and then click the Retransform icon, this will automatically re-validate the data. If there is an intersection validation error, this will now appear. If there are no intersection errors, the Validate Workflow step will change from blue to green.

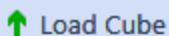
### Intersection Errors

Intersection Errors mean something is not correct with the entire intersection of data. For example, a Customer Dimension may be mapped to a Salary Grade Account. This does not make sense and the system will identify this prior to the load. To fix this error, click on the bad intersection and drill down to see the GL/ERP Account; (see Drill Down in "Using OnePlace Cube Views" on page 962 for more details). Right-click to View Transformation Rules and from there, investigate each rule for each Dimension in order to find which one does not make sense. Each Dimension will be on the left, and each Target Value Column needs to be investigated. To fix a Transformation Rule here, select the Target Value to correct, right-click and select Edit Rule. Choose the appropriate Target Value and fix the intersection error. Click Save and close the next two windows. Click the Validate icon again and if it was fixed correctly, the Validate task will change from blue to green.

See Right-Click Options under the Import Task for details on the right-click functions.

## Load

This simply loads the cleansed data from the Stage to the Analytic Engine. Click the Load task in the taskbar.

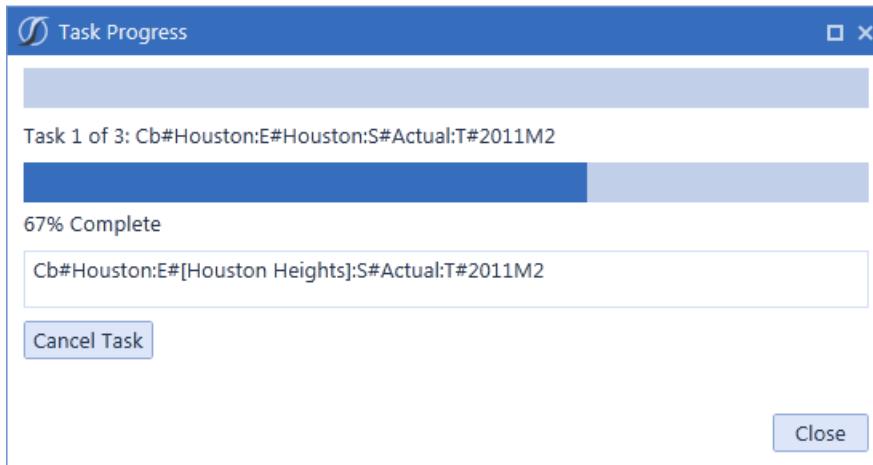


Next, click the Load Cube icon. This now loads the data into the Consolidation Engine and completes the Import process.

A Task Progress dialog will appear:

## Using OnePlace Workflow

---



A green check will appear next to the Import Origin Workflow Step.

See Right-Click Options under the Import Task for details on the right-click functions.

## Pre-Process

The Pre-Process task is the same as the normal Process task. This can be added as a first step in the Form channel process. Please refer to the Process description further down in this section for more details.

## Input Forms

Once in this step, there will be two options under Workflow Forms: Required and Optional. The required Forms will have to be completed before the end user can move on from this step. Click on the specific Form, and manually enter the required data.



### Open Excel Form

A user can export a Form derived from a Cube View to Excel, complete it in Excel, and then submit the data back to the application.



### Attachments

This allows the user to attach supplemental files to specific Forms.



### Import Form Cells Using Excel or Comma Separated Values File

Click this to load Form data via Excel template or CSV template. See Loading Form Data in "Collecting Data" on page 132 for more details on using these templates.

## Using OnePlace Workflow

---

### Import Cell Details Using Excel or Comma Separated Values File

Click this to load Cell Details via Excel template or CSV template. See Loading Cell Detail in "Collecting Data" on page 132 for more details on using these templates.



Once a Form is completed, click Save and then the Complete Form icon.

**NOTE:** A drop-down time period menu will display for a form with a defined time filter containing multiple periods. If a Form does not have a time filter defined, this will be a standard button.



This will re-open a Form and clear the data. Once the Form is updated, Complete Form must be clicked again.

**NOTE:** A drop-down time period menu will display for a form with a defined time filter containing multiple periods. If a Form does not have a time filter defined, this will be a standard button.



Checking this box enables all Forms to be completed/reverted in one step.



When each Form is completed, click Complete Workflow and a green check will appear for the Forms Origin Workflow Step.



This will re-open the Workflow. To make changes to the Form, Revert Form must also be clicked. Make any necessary changes to the Forms and click Complete Form/Workflow again.



This will run a consolidation/translation/calculation for a specific Form.



### Show Report

This shows the Form in a Cube View Report.

## Using OnePlace Workflow

---



### Export to Excel

This exports a copy of the Form into Excel.

### Cell Detail

Cell Detail can be entered on a Cube View used for a data entry form, or on a Cube View or Quick View in Excel. Cell Detail is available on any writeable O#Forms or O#BeforeAdj Member.

See Right-Click Options in "Using OnePlace Cube Views" on page 962 for details on the right-click options in Forms.

## Workflow Icons

The screenshot shows the OnePlace Workflow interface. At the top, there's a status bar with several buttons: 'Process' (green), 'Confirm' (red), 'Certify' (light blue), and 'Confirmation Error'. Numbered callouts point to specific elements: 1 points to the OneStream logo; 2 points to the rounded edges of the 'Process' and 'Confirm' buttons; 3 points to the up-arrow indicator on the 'Confirm' button; 4 points to the 'Confirmation Rules' section; 5 points to a green circle with a white checkmark; 6 points to a red circle with a white exclamation mark; 7 points to the 'US Standard' row; 8 points to the 'US Supplemental' row; and 9 points to the 'Form Validation' row. Below the status bar is a 'Status' table with columns for 'Status' and 'Rule Name'. At the bottom is a 'Summary Rule Analysis' table with columns for 'Status', 'Rule Name', and 'Rule Text'.

Status	Rule Name	Rule Text
Pass	Balance	Assets-Liabilities-Shareholders Equity must = 0
Pass	Cash Overdraft Reclass	Cash Deposits must be > 0, Overdraft must be reclassified to AP Trade.
Fail	Bank Statement Required	Bank Statement must be attached, if Cash Deposits are > 0
Pass	Headcount Import	Headcount must be > 0
Fail	Headcount Check Sum	Headcount Check Sum must = 0
Pass	Sales Detail Check	Sales Detail total must = Trial Balance Third Party Sales value.

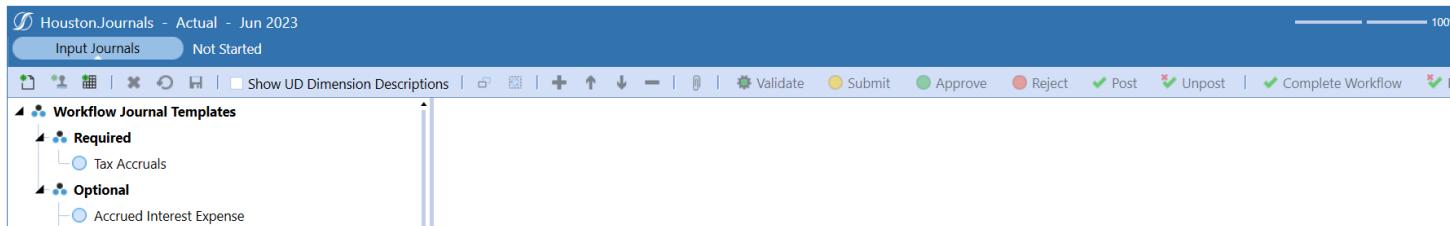
( All ) Summary Rule Analysis		
Status	Rule Name	Rule Text
Pass	Balance	Assets-Liabilities-Shareholders Equity must = 0
Pass	Cash Overdraft Reclass	Cash Deposits must be > 0, Overdraft must be reclassified to AP Trade.
Fail	Bank Statement Required	Bank Statement must be attached, if Cash Deposits are > 0
Pass	Headcount Import	Headcount must be > 0
Fail	Headcount Check Sum	Headcount Check Sum must = 0
Pass	Sales Detail Check	Sales Detail total must = Trial Balance Third Party Sales value.

1. OneStream logo is white.
2. Workflow Status bar steps have curved edges.
3. Failed task text is white.
4. Up-arrow indicator for selected task.
5. Stop sign has an X in the middle, and a flat design style.
6. Confirmation Rules; Pass = green circle white checkmark, flat design style.
7. Confirmation Rules; Fail = red circle white exclamation mark, flat design style.

8. Status; last step completed green circle.
9. Status; last step error red circle.

## Journals Toolbar

The Journals toolbar contains many tools for working with journals. This section describes each tool in detail.



## Input Journals

Once in this step, there will be two options under Workflow Forms: Required and Optional. The required Journals will have to be completed before the end user can move on from this step.



Click this to create a Journal if a template is not available. If this button is disabled, it is because the workflow profile is configured to only allow you to create journals based on existing templates.



Use this to create a Journal using the selected template. Manually enter the required Journal data and Click Save.



### Create Journal Using Excel or Comma Separated Values File

Click this to load a journal entry via Excel template or CSV template. See Loading Journal Data in "Collecting Data" on page 132 for more details on using these templates.

**NOTE:** After journals are copied, WfProfile\_WfUnit\_Scenario\_TimePeriod\_Index is appended to the journal name.

### \* Delete Selected Item

Delete one or more journals simultaneously. Journals in Working or Rejected status can be deleted. This saves time when you need to remove multiple journal entries quickly and efficiently. Select the boxes next to the journals you want to remove then click this button and confirm.



### Reapply Template Settings to Journal

This will clear the data from the selected Journal and return it to the original template.

Show UD Dimension Descriptions **Show UD Descriptions**

Check this checkbox to display descriptions for the eight UD Dimension Types. They will display next to the Dimension Type in the Journal POV and the line item headers. Descriptions are defined in the Application Properties, and can provide clarity on the purpose of the dimensions.

**NOTE:** This feature applies to the Journal page only. Reports, BRApis and Excel/CSV templates related to journals remain unchanged.

### Journal Entry Checkboxes

Select multiple journal entries in the same state (e.g., completed, posted, approved, etc.) in order to Submit, Post, Approve, or Quick Post them at the same time. Select multiple journal entries in either the Working or Rejected status to delete them at the same time.

- ▲  Journals
  - ✓ Accrued Interest Expense\_Houston.Journals\_Actual\_2011M1
  - ✓ Tax Accruals\_Houston.Journals\_Actual\_2011M1

### Validate

You can ensure that dimension members selected in the journal are valid cube intersections based on constraints configured in the dimension library. If validation errors exist, they will display in a Validation pane allowing you to view them as you make edits to the journal. This helps you save time by validating entries earlier in the process.

You must have the Journal Processing security role to use this feature and dimension member constraints must first be configured in the dimension library.

To use this feature:

1. Open a journal that is in Working status.
2. Click **Validate**.

## Using OnePlace Workflow

---

If invalid data is found, the Validation pane will display messages that detail each entry error.

### Quick Post

Once a Journal is completed, select Quick Post or Post. When all required Journals and any optional Journals are finished, click Complete Workflow and a green check will appear for the Journals Origin Workflow Step.



Depending on the security configuration, there are multiple options for Journals. If full security is in place, the end user will be able to create and submit. The approver will approve or reject, then the end user can post and complete the Workflow.

### Journal Line Items

Journal data is entered into line items. You can add line items to reflect the Dimensions they are adjusting. Any dimensions not in line items must be configured in the Journal POV.

To select a dimension member in journal line items:

1. Double-click the Entity, Account, Flow IC, or UD1-UD8 field.
2. If you know the member name, type it into the text field. Repeat for all fields as needed.
3. If you do not know the member name, use the Select Member dialog to locate the member.

**NOTE:** If the dimension members have a description, they are displayed in the journal line items. If dimension members contain language-specific descriptions, as determined by Culture, those descriptions display in the selected language. Descriptions are read-only, and can be set on journals and journal templates, reports, BR APIs, and Excel/CSV templates.

## Process

Once in this step, click Process Cube in order to process the loaded data.

### Process Cube

This has Calculation Definitions built behind it. This icon performs No Calculate and all the standard Calculate, Translate, and Consolidate options. Furthermore, this can be configured to filter by Entity for Reviewer level processes. Once the Cube is processed, the Process task will change from blue to green and move to the Confirm task.

# Confirm

### Run Confirmation

This step runs the Confirmation Rules defined for this particular Workflow. This will immediately inform users if they have passed or failed the data quality rules. The two types of statuses for this step are Warning or Error. Warning means the user is outside of the threshold, but it will not stop the process. Error means the user is outside of the threshold and this will stop the process and turn this step to red. If anything has failed, revisit one or many of the previous steps to make sure the data is accurate and complete. Once the data has passed all quality rules, the Confirm task will change from blue to green and move to the Certify task.

# Certify

This is typically the final step in the Workflow process. This certifies and completes the phase of the Workflow.

### Set Questionnaire Status

Some questions may need to be answered regarding the processes. Click each set of questions under the Questionnaires area. Answer the questions by clicking in the response cell and selecting the correct answer. Comments can also be added in order to explain the answers. The status will be displayed on the right. When this is completed, click on the Set Questionnaire Status icon and select Completed and then OK.

### Set Certification Status

Once each group of questions is completed, the Set Certification Status icon will be enabled, click this and select Certify in order to certify the data as complete and accurate. This will give the final green check for the month being processed and the data can now be trusted as complete and accurate by any stakeholder that is analyzing this information.

### Quick Certification

This is a one click option to expedite the Certify process. No questions need to be answered. This will give the final green check for the month being processed and the data can now be trusted as complete and accurate by any stakeholder that is analyzing this information.

This data can now be used for Consolidations by users or managers responsible for this Workflow. They can look at data as it moves up and perform their own top side adjustments, confirmations and certifications at as many levels as is appropriate for the organization.

## Using OnePlace Workflow

### Dependent Status

Click on Dependent Status to see the status of each required Workflow task to ensure they are all completed. This will display the Workflow Profile name and all input types, the Workflow Channel, the status of each input type, the last step completed for each input type, the percentage of each step that is OK, In Process, Not Started, and steps with errors, and a record of when the last activity took place for each step. See Right-Click Options for details on these right-click functions.

Dependent Status									
Name	Channel	Status	Last Step	% OK	% IP	% NS	% ERR	Last Activity Time	
Houston		Green	Confirmation Completed	100	0	0	0	11/25/2014 1:47 PM	
Houston.Import	Standard	Green	Completed					11/24/2014 4:55 PM	
Houston.Sales Detail	Standard	Green	Completed					11/24/2014 4:55 PM	
Houston.Forms	Standard	Green	Completed					11/12/2014 4:49 PM	
Houston.Journals	Standard	Green	Completed					9/23/2014 4:07 PM	

For more details on Workflow, see Workflow in "Workflow" on page 445.

Import (Stage Only), Import, Validate (Stage Only), Import, Validate, Certify (Stage Only)  
This setting in the Workflow limits the data load process to only the (Import), (Import, Validate), or (Import, Validate, Certify) step. When the POV is set to this Workflow, only these steps will be available. No data will be loaded to the Cube.

## Multi-Period Processing

Click on the Year in the Navigation Pane to enact Multi-Period Processing options:

The screenshot shows the OneStream XF interface with the 'Workflow' module selected in the navigation pane. The navigation pane displays a hierarchy: System > OnePlace > Workflow > Houston > Actual > 2011 > 2011 Periods. Under '2011 Periods', 'Jan' and 'Feb Houston' are listed, with 'Feb Houston' currently selected. The main window displays the 'Workflow Multi-Period Processing' dialog. The 'Execute Batch' button is at the top. Below it are three tabs: 'Select Workflow' (selected), 'Select Step', and 'Select Periods'. The 'Select Workflow' tab shows a tree view of workflow units: Houston (Import, Sales Detail, Forms, Journals), Workflow (Calculate Cube, Confirm Cube Data, Certify Workflow Unit), and Lock (Lock, Lock Descendants, Certify and Lock Descendants). The 'Select Step' tab shows a tree view of steps: Unlock (Unlock, Unlock Descendants, Unlock and Uncertify Ancestors), Workflow (Calculate Cube, Confirm Cube Data, Certify Workflow Unit), and Lock (Lock, Lock Descendants, Certify and Lock Descendants). The 'Select Periods' tab shows a list of periods: All, Jan (Completed, Closed), Feb (Process Cube Completed), Mar (Not Started), Apr (Not Started), May (Not Started), Jun (Not Started), Jul (Not Started), Aug (Not Started), Sep (Not Started), Oct (Not Started), Nov (Not Started), and Dec (Not Started).

From here, perform multiple Workflow tasks for one to many time periods, as shown above.

## Analysis Pane

In each Workflow step at the single period process within the monthly Process, Confirm and Certify tasks, there will be an Analysis Pane under the Status Pane.

## Cube Views and Dashboards for Analysis

This is where data can be viewed and analyzed in Cube Views and Dashboards. If there is a grid in any Dashboard or Cube View, these are available for further Drill Down or Annotation by right clicking the cell. See "Using OnePlace Cube Views" on page 962 and "Using OnePlace Dashboards" on page 997 for more details.

To view the Cube Calculation Status, click on Cube Views| Calculation Status to show a data grid presenting the Calculation Status of the current active Workflow POV. This will be available at the total Monthly (not an Origin process) Review, Process, or Certify tasks. See Calculation Status in "About the Financial Model" on page 2 for more details on this feature.

## Intercompany Matching

IC Matching which will show any Intercompany discrepancies. If the button is red in the status column, click the item to see the details. Each Intercompany counterparty will be visible and again in red will be counterparties with an Intercompany variance. By clicking on the counterparty, details including the Reporting Currency, Entity Currency and Partner Currency will be visible.

Select the Intercompany Partner to see the status of the Intercompany issue and any annotations the partner may have made. Select the Difference row to see both parties' statuses and annotations.

Right click on the Entity row to perform the following functions:

### Set IC Transaction Status

This allows the user to update the status by selecting Not Started, Loaded, Adjusting, Disputed, Finalized. The user may also include comments for the counterparty to see.

### Show Partner Workflow Status

This allows the user to see the Workflow status of the counterparty.

### Show/Hide Dimension Details

This allows the user to see the Dimension details for the Intercompany accounts.

### Drill Down

## **Using OnePlace Workflow**

---

This will allow the user to drill down on the Dimensions in order to get more information about the Intercompany data.

See Drill Down in "Using OnePlace Cube Views" on page 962 for more details on this feature.

See Intercompany Eliminations in "About the Financial Model" on page 2 for more details on Intercompany.

# Using OnePlace Cube Views

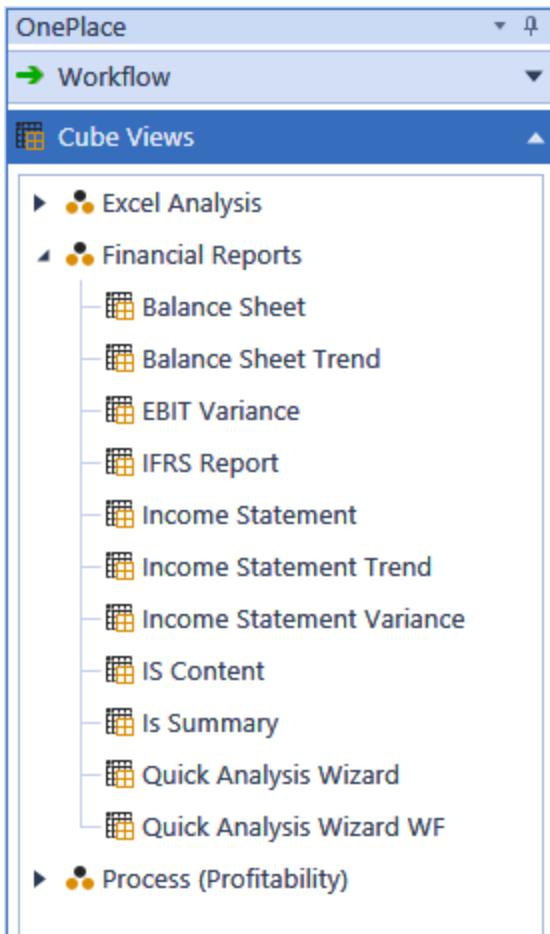
These Cube Views are complete and organized reports created specifically for the company's data in the Application section. A Cube View is used to query Cube data and present it to the user in variety of ways. They can be made read-only, used for editing data, and can be used as the Data Source for several different display mechanisms.

In order to copy Cube View cells from a Data Explorer Grid to an Excel spreadsheet, click CTRL, select the cells desired, and then click CTRL-C. Navigate to an Excel spreadsheet, select a cell, and click CTRL-V, this will paste the cells into Excel. This can also be done from an Excel spreadsheet into a Data Explorer Grid.

While viewing these reports, users can right click on any cell in order to learn more about any given number or piece of data. For more information on creating Cube Views, see Cube Views in "Presenting Data With Books, Cube Views and Dashboards" on page 547. For information on how to use Cube Views, or advanced uses, see Cube Views in "Presenting Data With Extensible Documents" on page 222.

## Using OnePlace Cube Views

---



## Toolbar

### Consolidate

This consolidates the data in the Cube View

### Translate

This translates the data in the Cube View

### Calculate

This performs a calculation on the Cube View data

### Row Suppression

### Use Default Suppression

This suppresses the data based on the Cube View's row suppression settings.

### Suppress Rows

This suppresses any data rows with zeroes, no data, etc. regardless of the Cube View suppression settings.

### Unsuppress Rows

This removes any data suppression set via the Cube View suppression settings and displays all data rows including those with no data, zeroes, etc.

**NOTE:** The actions above vary based on the user's security settings and restriction properties set on the Cube View.

### Show Report

This opens the Cube View in a polished, formatted report

### Export to Excel

This will open the fully formatted Cube View in Excel.

Users can open multiple excel exports using a Cube View without being prompted to rename or save the file. A version number will change with each export in sequence.

### Select Parameters

This allows the user to select new Parameters and view the Cube View data differently.

**NOTE:** This is based on the Parameters set for this Cube View. This feature will not apply to all Cube Views

### Edit Cube View

This opens the Cube View Application page where changes can be made to the Cube View design.

**NOTE:** This varies based on the user's security settings.

## Using OnePlace Cube Views

---



### Find Next Row

When viewing a Cube View with a large number of rows, use the search filter at the top of the screen in order to navigate to the desired row. Type a keyword into the search filter and click the Find Next Row icon in order to navigate to the preferred row. Continue clicking the icon in order to navigate to any row with the keyword included in the row header name. This also works when rows are collapsed.

Balance Sheet			
Assets	Jan 2011	Jan 2010	
10000 - Petty Cash	2,884,694	2,403,912	
10100 - Cash Deposits	51,667,478	43,056,232	
10300 - Marketable Securities	6,983,071	5,819,226	
10400 - Restricted Cash	6,409,952	5,341,627	
<b>10999 - Total Cash</b>	<b>67,945,196</b>	<b>56,620,997</b>	

### Copy or Paste Data to Multiple Cells

Use <Ctrl+C> or <Ctrl+V> to copy and paste values into Cube View data cells. Select the desired data cell, click <Ctrl+C> to copy the cell value and then select another cell to paste <Ctrl+V> the value. Hold down <Ctrl> to select multiple cells and paste the value into the selected cells simultaneously.

## Shortcuts, Scaling, and HotKeys

The Cube View Data Explorer allows the use of shortcuts, scaling, and hotkeys.

### Shortcuts

Use Shortcuts to calculate amounts within a cell. Click once in a cell, type in a shortcut, then click enter to run the calculation.

## Using OnePlace Cube Views

**Example:** Type "add2k" to add 2,000 to the cell value. Type "in10" to increase the cell value by 10%. Type "pow2" to calculate to the power of 2.

The figure consists of three screenshots of a Data Explorer interface, labeled 1, 2, and 3. In each screenshot, the 'Select Entity' dropdown is set to 'Frankfurt' and the 'Headcode' dropdown is set to 'New Hires'. The table contains four rows: 'MatMgt - Material Management', 'Logistics', 'Production', and a blank row. In screenshot 1, the 'Production' cell contains '100.0'. In screenshot 2, the 'Production' cell contains 'pow2'. In screenshot 3, the 'Production' cell contains '10,000.0'.

### Shortcuts

add	Add a number to the last value
sub	Subtract a number from last value
div	Divide last value of a number
mul	Multiply last value of a number
increase, in, gr	Increase last value by a percentage
decrease, de	Decrease last value by a percentage
percent, per	Calculate a percentage of last value
power, pow	Calculate a power of last value

## Scaling

When entering data in the Data Explorer, use scaling to quickly enter values.

**Example:** Type "1m" for a cell value of "1,000,000"; "1.5b" for a cell value of "1,250,000,000"

Scaling	
k	Scale value by 1,000
m	Scale value by 1,000,000
b	Scale value by 1,000,000,000
%	Divide by 100 (when using with a P number format on Cube Views)

## HotKeys

Use HotKeys to manage changes while entering data into the Data Explorer.

 **Example:** Press CTRL+S to save changes.

HotKeys	
CTRL + S	Save

## Right-Click Options

### Expand/Collapse

If nested rows are used in a Cube View, right-click on any row header in order to collapse or expand the selected row content. This feature works with the RowExpansionMode property in the Cube View designer and controls how users view Cube Views in the data grid. See Rows and Columns under Cube Views in "Presenting Data With Books, Cube Views and Dashboards" on page 547 for more details.

### Calculate/Translate/Consolidate

Cube Views can be set to enable processing of the Cube View data. This includes Calculate, Translate and Consolidate. There are also option to force these calculations and log the activity in the Task Activity. Each of these options can be enabled/disabled on an individual Cube View. For more details on these options, see Launching a Consolidation in "About the Financial Model" on page 2.

## Spreading

Spreading can be done from a Cube View while viewing it in the Data Explorer grid, in the Spreadsheet feature and Excel Add-in. User experience may differ slightly across these three user interfaces and more functions may be available in the Spreadsheet feature than others. This functionality provides the ability to spread values over selected cells.

**NOTE:** The Spreading Dialog can be left open while entering data. If the dialog is left open while users are spreading values on the grid, it will update to the respective spreading behavior.

## Spreading Type

### Fill

This fills each selected data cell with the value in the Amount to Spread property.

### Clear Data

This clears all data within the selected cells.

### Factor

This takes the selected cell's value and multiplies it by the rate specified.

### Accumulate

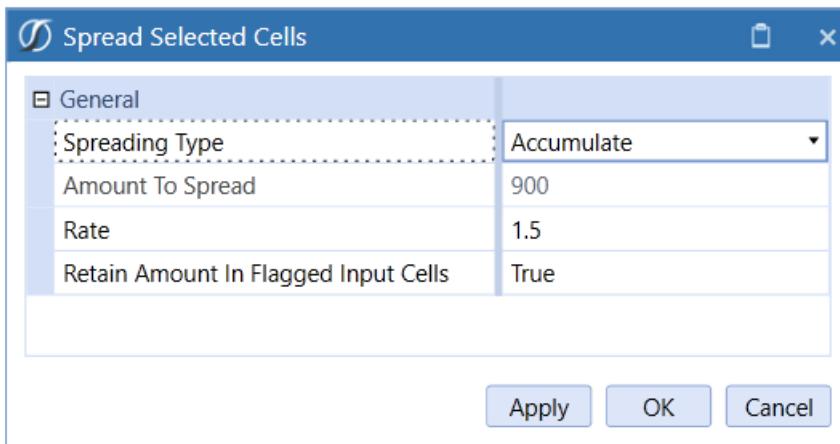
This takes the first selected cell's value and multiplies it by the rate specified. It then takes that value, multiplies it by the specified rate and places it in the second cell selected, and does this for all selected cells. For example, four cells are selected and the first cell has a value of 900.

Apr 2011	May 2011	Jun 2011	Jul 2011
909.00	14.00	622.50	33.75
900.00	0.00	0.00	0.00

The Accumulate Spreading is setup as follows with a rate of 1.5:

## Using OnePlace Cube Views

---



When the spreading is applied the outcome is as follows:

Apr 2011	May 2011	Jun 2011	Jul 2011
1,359.00	2,039.00	3,660.00	4,590.00
1,350.00	2,025.00	3,037.50	4,556.25

Each cell's value is a factor of the previous cell amount.

### Even Distribution

This takes the Amount to Spread and distributes it evenly across the selected cells.

### Proportional Distribution

This takes the selected cell's value, multiplies it by the specified Amount to Spread, and then divides it by the total sum of all selected cells. If all the cells have a zero value, the Amount to Spread will behave like an Even Distribution.

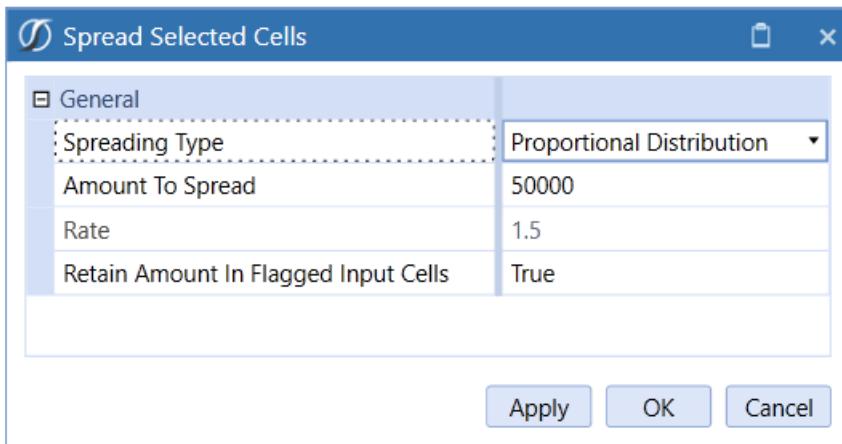
In the example below, four cells are selected:

Apr 2011	May 2011	Jun 2011	Jul 2011
1,359.00	2,039.00	3,660.00	4,590.00
1,350.00	2,025.00	3,037.50	4,556.25

A proportional amount of 50,000 is applied to the cells.

## Using OnePlace Cube Views

---



### Result

Apr 2011	May 2011	Jun 2011	Jul 2011
6,162.85	9,244.77	14,468.65	20,802.98
6,153.85	9,230.77	13,846.15	20,769.23

#### 445 Distribution

This takes the Amount to Spread and distributes it with a weight of 4 to the first two selected cells and a weight of 5 to the third cell.

#### 454 Distribution

This takes the Amount to Spread and distributes it with a weight of 4 to the first selected cell, a weight of 5 to the second cell and a weight of 4 to the third.

#### 544 Distribution

This takes the Amount to Spread and distributes it with a weight of 5 to the first selected cell and a weight of 4 to the second and third cells.

## Spreading Properties

### Amount to Spread

Specify the value to spread over the selected cells. The value defaults to the last cell selected. The way the amount in this field spreads varies by Spreading Type.

### Rate (Factor and Accumulate Spreading Types Only)

Enter a rate to multiply by a cell value.

### **Retain Amount in Flagged Input Cells**

Users can flag specific cells in order to retain the data within the cell. If this property is set to True, spreading will not apply to the selected flagged cells.

### **Include Flagged Read only Cells in Totals**

Set this to True to include locked base-level cell values when calculating spreading totals. True is the default.

### **Flag Selected Cells**

Flag cells so the original amount in the cell is retained during the spreading process.

### **Clear Flags**

Select this to clear any flagged cells.

### **Apply**

Enter spreading values in the dialog and select Apply to perform spreading without closing the dialog. The dialog closes upon clicking OK or Cancel.

### **Spread Selected Cells**

In order to select multiple data cells, hold the Control Key (Ctrl) on the keyboard, and click on all cells that apply. Spreading occurs across rows first and then down the columns. Once spreading has been performed using the Spreading Dialog, users can type values on specific Members without having to launch the dialog again. This spreading function applies the settings from the last time the Spreading Dialog was used, so all Spreading Types and flagged cells can be utilized. Even Distribution is the default Spreading Type if there is nothing currently set in the Spreading Dialog.

Users can select multiple Cube View data cells and type over the primary cell in order to apply spreading. If the primary selected cell is a Parent Member, a value can still be typed over it if multiple cells are selected. Any locked base-level data cells are automatically flagged when applying spreading.

When spreading across multiple time periods on one data row, users can select the Parent Time Member and automatically apply spreading to all its Base Time Members without having to manually select them.

For example, double-click on the Q1 Parent Member and the periods associated with Q1 will be selected.

## Using OnePlace Cube Views

---

The screenshot shows a software interface titled "OnePlace Cube Views". At the top, there are several icons: a magnifying glass, a dollar sign, a calculator, a refresh arrow, a file folder, and an Excel icon. Below the toolbar is a header row with four columns: "Jan", "Feb", "Mar", and "Q1". The "Q1" column has a blue circular icon with a clock symbol. The main data area contains eight rows of expense items, each with a small icon and a link-like text. The "Q1" column is highlighted with a light green background. The last cell in the "Q1" column contains the value "1302.0833333333".

	Jan	Feb	Mar	Q1
52299 - Total Facility Expense	63,371.57	103,698.94	87,717.86	254,788.37
52200 - Rent				
52210 - Property Ins			1,302.08	1302.0833333333
52220 - Property Taxes				
52230 - Building & Grounds Maint	34,595.67	56,611.10	47,175.91	138,382.68
52240 - Janitorial Exp	28,775.90	47,087.84	39,239.87	115,103.60
52250 - Business Fees				
52260 - Other Facility Exp				

Enter a value to spread on the Q1 data cell.

This screenshot shows the same software interface as the previous one, but with a value entered into the "Q1" cell for the "Property Ins" item. The value "1500" is typed into the cell, which is part of the "Q1" column. The rest of the data remains the same as in the first screenshot.

	Jan	Feb	Mar	Q1
52299 - Total Facility Expense	63,371.57	103,698.94	87,717.86	254,788.37
52200 - Rent				
52210 - Property Ins			1,302.08	1500
52220 - Property Taxes				
52230 - Building & Grounds Maint	34,595.67	56,611.10	47,175.91	138,382.68
52240 - Janitorial Exp	28,775.90	47,087.84	39,239.87	115,103.60
52250 - Business Fees				
52260 - Other Facility Exp				

Click <Enter> and it will automatically apply to the periods associated with the Quarter.

## Using OnePlace Cube Views

---

	Jan	Feb	Mar	Q1
52299 - Total Facility Expense	63,371.57	103,698.94	87,717.86	254,788.37
52200 - Rent				
52210 - Property Ins	500.00	500.00	500.00	1,302.08
52220 - Property Taxes				
52230 - Building & Grounds Maint	34,595.67	56,611.10	47,175.91	138,382.68
52240 - Janitorial Exp	28,775.90	47,087.84	39,239.87	115,103.60
52250 - Business Fees				
52260 - Other Facility Exp				

## Allocation

Form data Allocations can be completed on a Cube View in the Data Explorer Grid, a Cube View in Excel, and Quick Views.

### Allocation Type

#### Clear Data

This clears all Form data for the specified Destination POV.

**NOTE:** Users can clear data for Dimension Members that are not displayed on the Form.

For property definitions, see [Even Distribution](#).

### Even Distribution

This provides an even distribution across the Destination Members.

### Source POV

The Source POV determines the source intersection and applies the value from this intersection to the allocation. The Source POV defaults to the last cell selected. Users can also select a data cell and drag and drop the cell's POV into this property in order to update it. Any Members not included in the Source POV will default to the user's Cube POV.

### Source Amount or Calculation Script

Use this property to override the Source POV value. Drag and drop a value from the Cube View grid, specify a source amount (e.g., 1000), or use the Source POV to specify a different source amount (e.g., A#SourcePOV\*0.90).

### Destination POV

The Destination POV determines the intersection where the allocation will take place. A Member for each Dimension must be specified for the allocation to take place in the correct intersection. Users can select a data cell and drag and drop a POV into this field in order to display all Dimensions. If the Destination POV is similar to the Source POV, specify as many Dimensions as necessary and the remaining Members will come from the Source POV, or leave this field blank and all POV Members will be based on the Source POV. Any Members not included in the Source or Destination POV will default to the user's Cube POV.

### Dimension Type/Dimension Type 2

Allocations can be applied to Dimension Members not included in the Destination POV such as several periods or Accounts. These Members then override the Destination POV. Specify the Dimension in this property. For the Clear Data and Advanced Allocation Types, two Dimensions can be specified.

### Member Filter/Member Filter 2

Specify the Dimension Members to which the allocation will apply. The Members specified in this field override the Member in the Destination POV script. For the Clear Data and Advanced Allocation Types, two Dimension Member Filters can be specified.

### Save Zeroes as Not Data

Set this to True in order to suppress zeroes when saving allocation data. The default setting is True.

### 445 Distribution

A 445 Distribution takes the source amount and applies a weight of 4 to the first two specified destination intersections and then a weight of 5 to the third intersection. This applies the allocation across rows first and then moves down the column.

### 454 Distribution

A 454 Allocation takes the source amount and applies a weight of 4 to the first destination intersection, a weight of 5 to the second and then a weight of 4 to the third intersection. This applies the allocation across rows first and then moves down the column.

### 544 Distribution

A 544 Allocation takes the source amount and applies a weight of 5 to the first destination intersection and then a weight of 4 to the second and third intersections. This applies the allocation across rows first and then moves down the column.

For Source and Destination property definitions, see [Even Distribution](#) above.

### Weighted Distribution

This applies a weighted value to each specified Destination Member. The weights are determined in a Weight Calculation Script which uses the specified Dimension intersections' cell values.

For Source and Destination property definitions, see [Even Distribution](#) above.

## Using OnePlace Cube Views

---

### Weight Calculation Script

The weight calculation script calculates the value by determining

```
|SourceAmount| * (|Weight| / |TotalWeight|)
```

It then applies the weighted values to the intersections specified in the Destination POV/Member Filter combination.

These are system Substitution Variables and are determined by the following:

#### |Weight|

This is the weight value applied to each Member in the Destination POV.

#### |TotalWeight|

This multiplies the weight by number of allocated intersections. For example, if there is a weight of 5 being allocated across 12 intersections, the |TotalWeight| would be 60.

Identify specific Dimension Members separated by a colon in order to determine the |Weight| and |TotalWeight|. Any Dimensions not specified in this field will come from the Destination POV.

Users can drag and drop a POV from a data cell in order to jumpstart the calculation script. In order to apply the Members specified in the Member Filter property, delete that particular Dimension in the script.

## Example

### Destination POV

```
Cb#Houston:E#[Houston  
Heights]:C#USD:S#Actual:T#2011M5:V#Periodic:A#56000:F#None:O#Forms:I#None:U1#Non  
e:  
U2#None:U3#None:U4#None:U5#[IFRS Adj]:U6#None:U7#None:U8#None
```

### Member Filter

```
T#2011M7,T#2011M8,T#2011M9
```

## Weight Calculation Script Example 1

```
A#26000:O#Top
```

The remaining Dimensions are determined by the Destination POV and Member Filter.

## Weight Calculation Script Example 2

```
Cb#Houston:E#[Houston Heights]:C#USD:S#Actual:V#Periodic:A#56000:F#None:O#Forms:  
I#None:U1#None:U2#None:U3#None:U4#None:U5#[IFRS Adj]:U6#None:U7#None:U8#None
```

## Using OnePlace Cube Views

---

The Time Dimension was removed from the script in order to use the three Time Members in the Member Filter.

In the example above, the Weight Calculation Script is identifying three intersections of data. OneStream uses the sum of those three intersections as the `|TotalWeight|` and each individual intersection as the `|Weight|`. This determines how to spread the Source Amount amongst the Destination Members.

Enter a value in order to create an even distribution to all specified Destination Members.

### Advanced

Advanced Allocations are similar to Weighted Distributions, but allow users to override two destination Dimensions, control how the weights are calculated, and offset amounts for the Source and Destination Members. See Form Allocations in "Collecting Data" on page 132 for an Advanced Allocation example.

For Source and Destination property definitions, see Even Distribution above.

For Weight Calculation Script, see Weighted Distribution.

## Destination Calculation Script

This determines how to calculate the Weight Calculation Script. The default calculation is `|SourceAmount| * (|Weight|/|TotalWeight|)`.

Example:

```
(|SourceAmount| * (|Weight|/|TotalWeight|)) *1.5
```

This will calculate the weighted value for each specified intersection, multiply it by 1.5 and apply that value to the destination data cell.

## Translate Destination if Different Currency

Set this to True in order to translate the destination currency if it differs from the currency on the Form.

## Offset

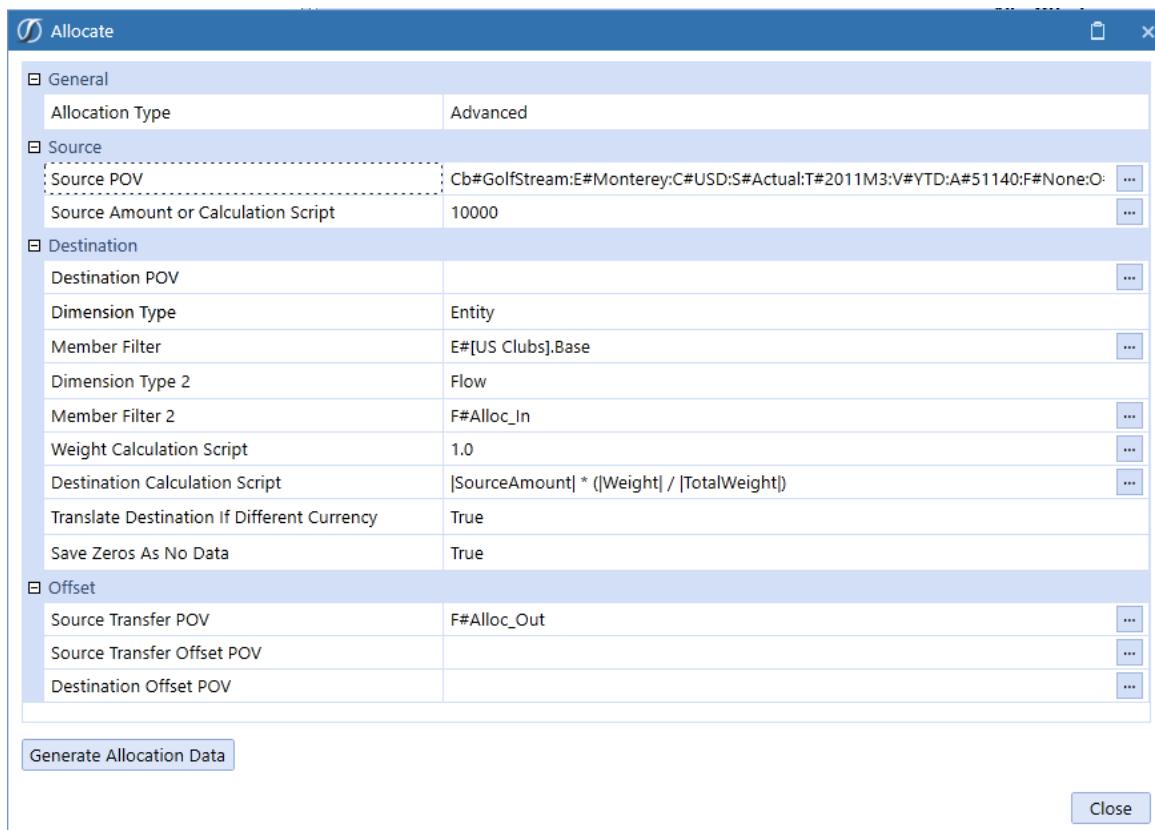
Specify specific Members to offset when doing an allocation. Transferring values out of the Source POV requires an entry in the Source Transfer POV while entries in Source Transfer Offset POV and Destination Offset POV will ensure that entries are balanced by updating a related Account.

## Using OnePlace Cube Views

		Monterey	Augusta	Carlsbad	Houston Heights	South Houston
51140 - IT Services Expense	TopFlow	0.00	2,500.00	2,500.00	2,500.00	2,500.00
	None	10,000.00				
	Allocations	-10,000.00	2,500.00	2,500.00	2,500.00	2,500.00
	Alloc_In		2,500.00	2,500.00	2,500.00	2,500.00
	Alloc_Out	-10,000.00				

In the example above, Monterey is transferring their IT Services Expense to four other Entities.

If a balanced entry is not required, the method below can be used by configuring the Destination and Source Transfer POV.



### Source Transfer POV

This is an optional field. Providing a Member Filter here will result in a transferring entry which will zero out the Source Amount. Using allocation members in the Flow Dimension in the example above, Monterey's original value is reduced to zero after netting F#None with F#Alloc\_Out, the Source Transfer POV. Note that the allocated values are targeted to the Alloc\_In Member in the Destination.

The screenshot shows the 'Allocation Results' dialog box. At the top, there is a 'Source POV' section containing a complex string of member filters. Below this, summary statistics are listed: 'Source POV Amount 10,000.00', '|SourceAmount| 10,000.00', '|NumDestinations| 4', and '|TotalWeight| 4.00'. A checkbox labeled 'Show All Dimensions' is present. The main area is a grid table with the following columns: Amount, Is NoData, |Weight|, Type, Account Type, Cube, Entity, Parent, Consolidation, and Flow. The data rows are:

Amount	Is NoData	Weight	Type	Account Type	Cube	Entity	Parent	Consolidation	Flow
-10,000.00	<input type="checkbox"/>	0.00	Source Transfer	Expense	GolfStream	Monterey		USD	Alloc_Out
2,500.00	<input type="checkbox"/>	1.00	Destination	Expense	GolfStream	Augusta		USD	Alloc_In
2,500.00	<input type="checkbox"/>	1.00	Destination	Expense	GolfStream	Carlsbad		USD	Alloc_In
2,500.00	<input type="checkbox"/>	1.00	Destination	Expense	Houston	Houston Heights		USD	Alloc_In
2,500.00	<input type="checkbox"/>	1.00	Destination	Expense	Houston	South Houston		USD	Alloc_In

At the bottom right of the dialog are 'Save Allocation Data' and 'Cancel' buttons.

### Source Transfer Offset POV

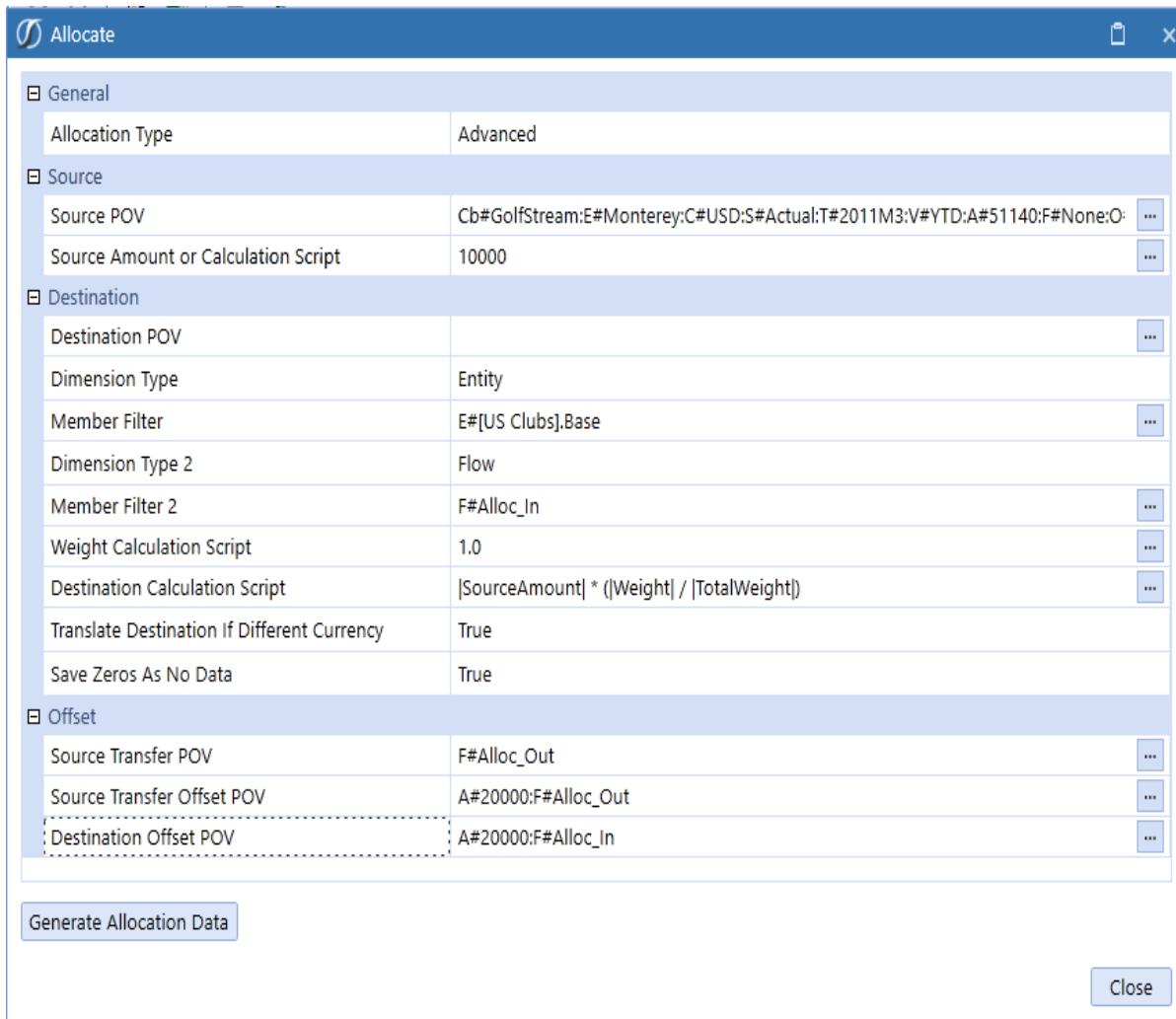
This is an optional field but required to create a balancing entry usually in a different Account in the Source. In the example below, the intersection is a different Account and results in a different intersection from the Source POV being updated to offset the Source Transfer POV update that occurs with the allocation. In the example below, an expense in the Source POV is being reduced so the offset is to a payable (Liability) Account.

### Destination Offset POV

This is an optional field but required to create a balancing entry usually in a different Account in the Destination. In the example below, the intersection is a different Account and results in a different intersection from the Destination POV or Member Filter being updated to offset the allocated value updates in the Destination. In the example below, an expense in the Destination is being increased so the offset is to a payable (Liability) Account. This could have been offset to an equity clearing Account instead, as this is typical.

## Using OnePlace Cube Views

---



This results in these entries:

## Using OnePlace Cube Views

---

**Allocation Results**

Source POV  
Cb#GolfStream:E#Monterey:P#?:C#USD:S#Actual:T#2011M3:V#YTD:A#51140:F#None:O#BeforeAdj:I#None:U1#None:U2#None:U3#None:U4#None:U5#None:U6#None:U7#None:U8#None

Source POV Amount 10,000.00

|SourceAmount| 10,000.00  
|NumDestinations| 4  
|TotalWeight| 4.00

Show All Dimensions

Amount	Is NoData	Weight	Type	Account Type	Cube	Entity	Parent	Consolidation	Scenario	Time	Is YTD	Account	Flow
-10,000.00		0.00	Source Transfer	Expense	GolfStream	Monterey		USD	Actual	2011M3	<input checked="" type="checkbox"/>	51140	Alloc_Out
-10,000.00		0.00	Source Transfer Offset	Liability	GolfStream	Monterey		USD	Actual	2011M3	<input checked="" type="checkbox"/>	20000	Alloc_Out
2,500.00		1.00	Destination	Expense	GolfStream	Augusta		USD	Actual	2011M3	<input checked="" type="checkbox"/>	51140	Alloc_In
2,500.00		0.00	Destination Offset	Liability	GolfStream	Augusta		USD	Actual	2011M3	<input checked="" type="checkbox"/>	20000	Alloc_In
2,500.00		1.00	Destination	Expense	GolfStream	Carlsbad		USD	Actual	2011M3	<input checked="" type="checkbox"/>	51140	Alloc_In
2,500.00		0.00	Destination Offset	Liability	GolfStream	Carlsbad		USD	Actual	2011M3	<input checked="" type="checkbox"/>	20000	Alloc_In
2,500.00		1.00	Destination	Expense	Houston	Houston Heights		USD	Actual	2011M3	<input checked="" type="checkbox"/>	51140	Alloc_In
2,500.00		0.00	Destination Offset	Liability	Houston	Houston Heights		USD	Actual	2011M3	<input checked="" type="checkbox"/>	20000	Alloc_In
2,500.00		1.00	Destination	Expense	Houston	South Houston		USD	Actual	2011M3	<input checked="" type="checkbox"/>	51140	Alloc_In
2,500.00		0.00	Destination Offset	Liability	Houston	South Houston		USD	Actual	2011M3	<input checked="" type="checkbox"/>	20000	Alloc_In

Save Allocation Data   

**NOTE:** Both the expense and liability have been removed from Monterey and transferred to the other four Entities:

AllocOffset						
			Monterey	Augusta	Carlsbad	Houston Heights
51140 - IT Services Expense	TopFlow	0.00	2,500.00	2,500.00	2,500.00	2,500.00
	None	10,000.00				
	Ending Balance					
	Allocations	-10,000.00	2,500.00	2,500.00	2,500.00	2,500.00
	Alloc_In		2,500.00	2,500.00	2,500.00	2,500.00
	Alloc_Out	-10,000.00				
20000 - Trade Accounts Payable	TopFlow	0.00	3,817,398.83	2,946,851.20	6,787,789.56	10,180,434.35
	None	10,000.00	3,814,898.83	2,944,351.20	6,785,289.56	10,177,934.35
	Ending Balance					
	Allocations	-10,000.00	2,500.00	2,500.00	2,500.00	2,500.00
	Alloc_In		2,500.00	2,500.00	2,500.00	2,500.00
	Alloc_Out	-10,000.00				

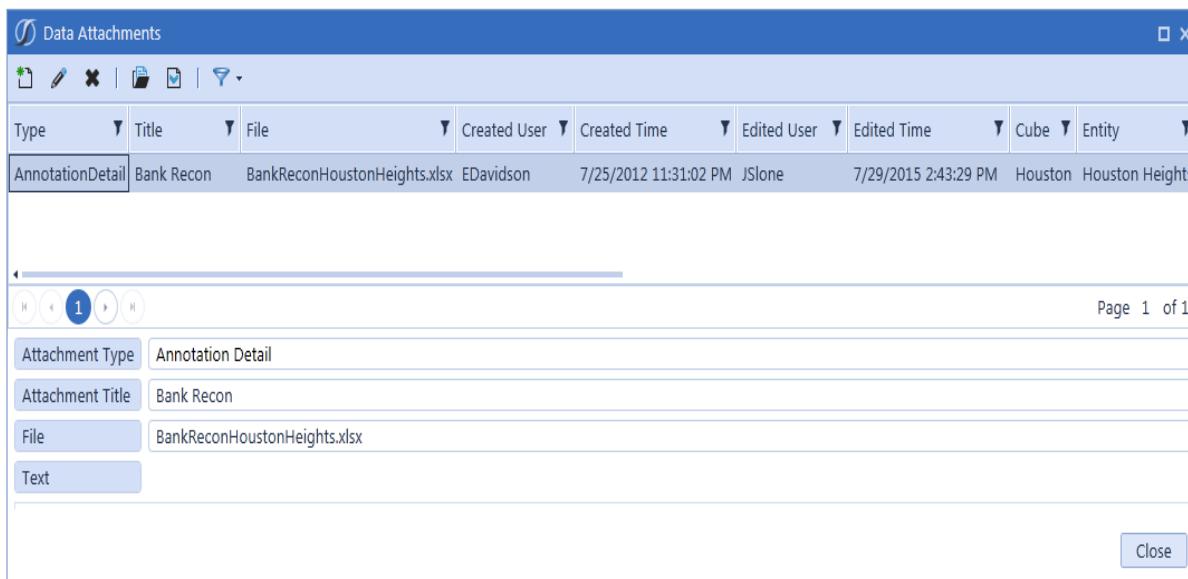
# Data Attachments for Selected Cell/Data Unit

Data Attachments can be added at the cell level or for the entire Data Unit and can be set up to show a red tick mark on cells that contain an attachment. These can be a textual comment, file attachment or both and with different types of attachments: Standard, Annotation, Assumptions, Audit Comment, Footnote or Variance Explanation. There can be many Standard attachments per cell or Data Unit, but only one of the other types. The other types of Data Attachments can be viewed in a row or column in a Cube View by selecting their related View Dimension Member. Attachment Types are part of the View Dimension which makes them available for a variety of reporting and explanation needed within the information delivery process. This allows them to be included within Cube View results.

### Data Attachments for Selected Cell/Data Attachments for Selected Data Unit

Any cell in any grid can contain a data attachment. To attach a file, right click on any cell in a data grid and select Data Attachment for Selected Cell. Click the upper left Create Data Attachment

icon  and select an Attachment Type from the drop-down menu, give it a title, select a file to attach or simply type in text, and click OK. The attachment will now appear in the Data Attachments box.



**TIP:** This attachment can be viewed from any data grid that employs a Workflow Unit, Scenario, and Period. To view all data attachments for any Workflow Unit, Scenario and Period, right click on any cell and select Data Attachments For Selected Data Units. This ensures that all data attachments become valuable analysis tools. This is part of the View Dimension because each is given a Type and the data attachments can be reported on as a whole and included in other types of reports.

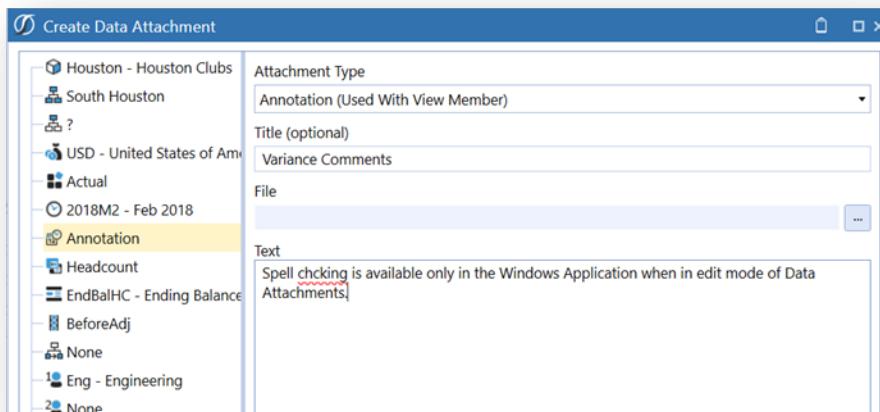
## Spell Checking in Data Attachments

A text spell check feature is available only when using the Windows Application. This feature is set as a default to be inactive. To enable the Spell Check feature, users must have access to the Application / Tools / Text Editor. In the Text Editor Tool, the Review ribbon will allow the user to activate Spell Check using the Spell Check button.

The Spell Check feature is enabled for English Culture only. The culture is determined by each user's culture assigned in OneStream User Security. The culture is assigned to the OneStream application on the Application Server Configuration Utility as "en-US". Users with cultures other than English (United States) will not have Spell Check available.

### Spell Check in Data Attachments

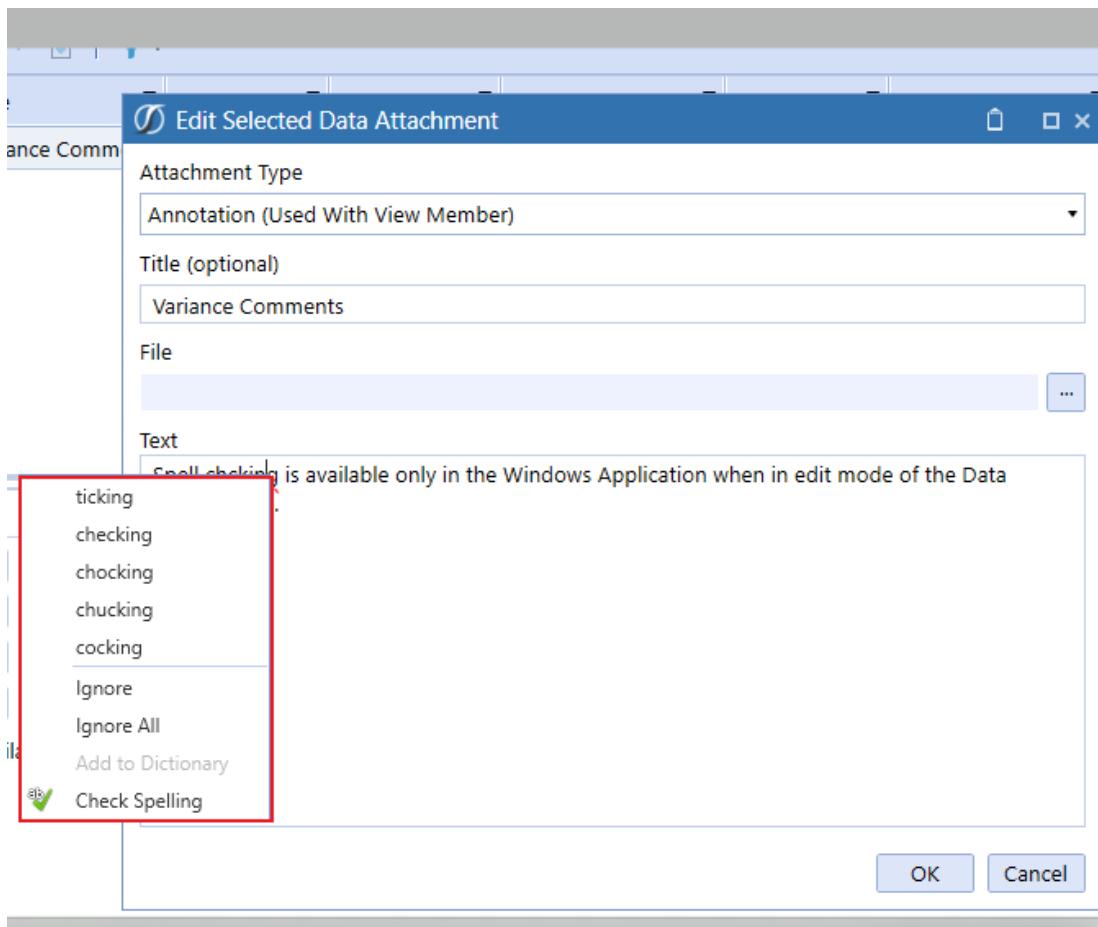
Spell Check will only be active when in the "new" or "edit" mode of Data Attachments. The Spell Check feature will actively check spelling within the selected Data Attachment type.



The user must right-click the triggered item to activate the Spell Check options where suggested solutions are presented. Choosing the "Ignore" option will only be retained for the current session. Closing and re-opening to the edit mode will re-check any previously ignored items.

## Using OnePlace Cube Views

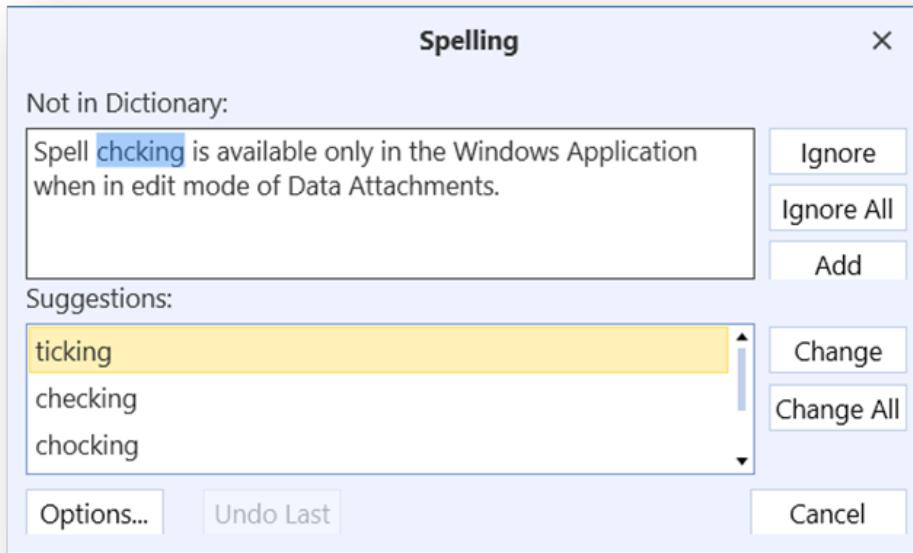
---



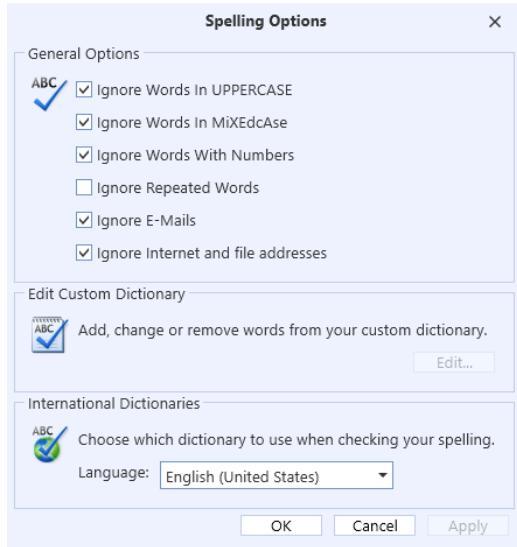
Choosing Check Spelling will allow the checking of spelling within content of the text box from any starting point.

## Using OnePlace Cube Views

---



The Options button will allow the user to modify the Spell Check behavior within the current task session. The settings are not persisted as user preference.



### Cell Detail

Cell Detail can be entered on a Cube View used for a data entry form, or on a Cube View or Quick View in Excel. Cell Detail can also be loaded via an Excel or CSV template. See Loading Cell Detail in "Collecting Data" on page 132 for more details on creating these templates. Cell Detail is available on any writeable O#Forms or O#BeforeAdj Member. In order to disable this by account or any other intersection, use a Conditional Input Business Rule. See Finance Business Rules in "Application Tools" on page 727 for an example of this rule.

The screenshot shows the 'Cell Detail' dialog box. At the top, there is a toolbar with a checkbox for 'Use Cell Detail' (1), radio buttons for 'YTD' and 'Periodic' (2), and buttons for adding (+), deleting (-), moving up (^), and moving down (^) (3). There is also a 'Clear All' button and a timestamp 'Last Edited by JSOne on 8/20/2015'. Below the toolbar is a table with four columns: 'Amount' (1), 'Aggregation Weight' (2), 'Classification' (3), and 'Description' (4). The table contains two rows of data:

Amount (1)	Aggregation Weight (2)	Classification (3)	Description (4)
10.11	1.00	Misc1	Miscellaneous Supply Expenses
10.00	-1.00	Misc1	Rebates Received on Supplies

Below the table, a message states 'O#Forms = 0.11'. Further down, a formula is shown: 'O#BeforeAdj (0.11) = O#Import (NoData) + O#Forms (0.11)' (5). At the bottom are buttons for 'Apply Import Offset' (6) and 'Remove Import Offset' (7), along with 'OK' and 'Cancel' buttons.

#### Cell Detail Toolbar

##### Use Cell Detail

Check this box in order to enable Cell Detail.

### YTD/Periodic

This determines how the data is being entered in the Form. This only applies to Income Statement Accounts. If a YTD Form is being used and a Periodic line item is entered, the Form will calculate the YTD value and store it accordingly



These icons add, delete, and move Cell Detail records in the grid.

### Clear All

This removes all Cell Detail records in the grid but does not remove any stored numerical data from the Forms Member in the Cube.

#### 1. Amount

Enter the amount for the Cell Detail Record Or Records.

#### 2. Aggregation Weight

Enter an aggregation weight in order to calculate a cell item using simple multiplication. For example, entering a -1 will reverse the value in the amount column.

#### 3. Classification

The user can select a Classification from a drop list on each line item, where the list of Classifications is defined using a Dashboard Parameter specifically named as CellDetailClassifications. Once this Dashboard Parameter is created, Cell Detail will recognize the classifications without having to assign the Parameter to any Cube View or Form.

Note: Users can add additional value items to a Parameter, or change existing items, however, this will not change any classification assigned and stored to a line item in a form.

#### 4. Description

Enter text information about the Cell Detail.

#### 5. This displays the values that are going to be stored in the Origin Members based on what was stored in the Cell Detail form.

#### 6. Apply Import Offset

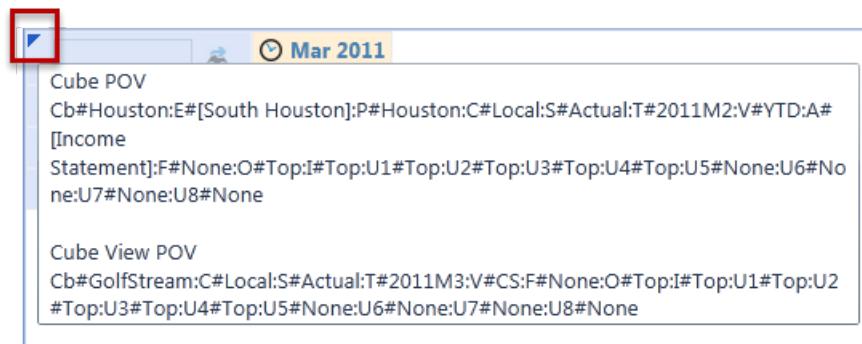
This is used mainly for Budgeting purposes. This applies the reverse amount of what is in the Import Origin Member allowing users to enter the total amount for the cell intersection without having to determine what has already been imported.

#### 7. Remove Import Offset

This removes an Import Offset Adjustment previously made via the Cell Detail form.

# Cell POV Information

By hovering over the tick mark on a previewed Cube View, defaulted Members from this Cube View's Point of View and the Cube's Point of View settings will display in the form of a tool tip.



Right click on any cell and select Cell POV Information in order to see a detailed summary of the selected Members related to this intersection. All the major properties of these Members can be seen from this dialog. The full Member Script and formula syntax to get to this value is also displayed. Users can use the clipboards to copy and paste the retrieve functions. The XFGGetCell formula can be copied in this dialog and used in an Excel file to retrieve this specific data in Excel. The XFCell formula can be copied in this dialog and used in a text file, such as Word or PowerPoint, to retrieve this specific data in an Extensible Document. See Extensible Document Framework in "Presenting Data With Extensible Documents" on page 222 for more details on this retrieve function.

## Using OnePlace Cube Views

The screenshot shows the 'Cell POV Information' dialog box. On the left is a tree view of dimensions: Cube [GolfStream - Corp], Entity [Houston], Parent [All Orgs], Cons [Local], Scenario [Actual], Time [2011M2 - Feb 2011], View [YTD], Account [10000 - Petty C], Flow [None], and Origin [Top]. The main area has two sections: 'General' and 'Properties'. Under General, there are four rows: Dimension Type (Cube), Id (5), Name (GolfStream), and Description (Corporate). Under Properties, there are five rows: CubeType (Standard), IsTopLevelCubeForWorkflow (True), TopLevelCubeWFPSuffixForActual (empty), TopLevelCubeWFPSuffixForBudget (empty), and TopLevelCubeWFPSuffixForFlash (empty). Below the dialog is a section titled 'Member Script And Formula Syntax' containing three examples of XFLR syntax.

Member Script And Formula Syntax

```
Cb#GolfStream:E#Houston:P#[All  
Orgs]:C#Local:S#Actual:T#2011M2:V#YTD:A#10000:F#None:O#Top:I#Top:U1#Top:U2#Top:U3#Top:U4#Top:U5#Non  
e:U6#None:U7#None:U8#None
```

```
=XFGGetCell(True, "GolfStream", "Houston", "All Orgs", "Local", "Actual", "2011M2", "YTD", "10000", "None", "Top",  
"Top", "Top", "Top", "Top", "None", "None", "None")
```

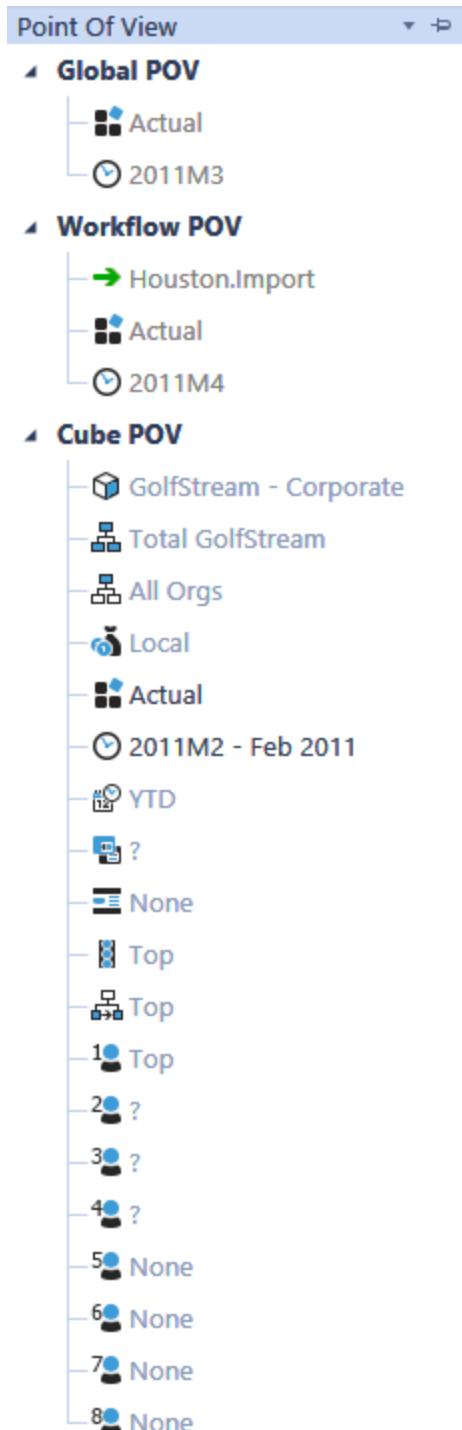
```
XFCCell(Cb#GolfStream:E#Houston:P#[All  
Orgs]:C#Local:S#Actual:T#2011M2:V#YTD:A#10000:F#None:O#Top:I#Top:U1#Top:U2#Top:U3#Top:U4#Top:U5#Non  
e:U6#None:U7#None:U8#None, Culture=User, NumberFormat=N0, DisplayNoDataAsZero=True, Scale=0,  
FlipSign=False, ShowPercentSign=False, UseApostrophePrefix=False)
```

Close

Interact with the Point of View in the Context Pane for the Cube View's bolded Dimensions. When these Dimensions are changed, the Cube View's results will change. In the example below the Cube View allows the Scenario and Time to change.

## Using OnePlace Cube Views

---



# Cell Status

Right click and select Cell Status in order to view status properties and Dimension information about a specific cell.



# Data Unit Statistics

Right click and select Data Unit Statistics in order to see details on the selected cell's Data Unit. (Number of: zero cells, real cells, derived cells, NODATA cells, calculated cells, consolidated cells, translated cells, journal cells, input cells, etc.)

D Data Unit Statistics	
Point Of View	
Cube	GolfStream
Entity	Clubs
Parent	
Consolidation	USD
Scenario	BudgetV1
Time	2011M3
General	
Total Number of Stored Records	1074
NODATA Status	
Number of NODATA Cells	48
Number of Zero Cells	9
Number of Real Cells	1017
Number of Derived Cells	0
Storage Type	
Number of No Activity Cells	0
Number of Input Cells	0
Number of Journal Cells	0
Number of Consolidated Cells	518
Number of Translated Cells	0
Number of Calculated Cells	556

[Close](#)

## Drill Down

When choosing Drill Down in the right click menu item in a Cube View or Quick View another tab will open with the Drill Down results. Drill down works the same whether a user is drilling from the data explorer grid in OneStream, from the Excel Add-In or OneStream Spreadsheet. An administrator might want to know what makes up Net Income for all Entities in Europe across all products groups, and with the Drill Down option this can be accomplished.

**TIP:** Drill into any cell on a data grid or in Excel; it does NOT need to be a base level number.

The resulting screen shows the drilled back intersection in the Drill Down History section. The white cells show base amounts, meaning drilling cannot go further. The green cells can continue to be drilled.

The screenshot shows the 'Drill Down' interface with two main sections: 'Drill Down History' and 'Results For Most Recent Drill Down'. Both sections have a header row with various columns: Amount, Cube, Entity, Parent, Consolidation, Scenario, Time, View, Account, Flow, Origin, IC, UD1, UD2, UD3, UD4, UD5, UD6, UD7, UD8, and Calculation. The 'Drill Down History' section contains three rows of data, where the first two rows have green-shaded 'Account' and 'Flow' columns, indicating they are drillable. The 'Results For Most Recent Drill Down' section contains three rows of data, with the first two rows having green-shaded 'Account' and 'Flow' columns.

Amount	Cube	Entity	Parent	Consolidation	Scenario	Time	View	Account	Flow	Origin	IC	UD1	UD2	UD3	UD4	UD5	UD6	UD7	UD8	Calculation
79,451,581.01	Houston - Houston Clubs	Houston		USD	Actual	2011M2 - Feb 2011 YTD		60999 - Net Sales	None	Top	Top	Top	Top	Top	Top	None	None	None	None	
79,451,581.01	Houston - Houston Clubs	Houston		USD	Actual	2011M2 - Feb 2011 YTD		60999 - Net Sales	None	Top	Top	Top	Top	Top	Top	None	None	None	None	
81,769,263.48	Houston - Houston Clubs	Houston		USD	Actual	2011M2 - Feb 2011 YTD		60000 - Operating Sales	None	Top	Top	Top	Top	Top	Top	None	None	None	None	
Results For Most Recent Drill Down																				
Amount	Cube	Entity	Parent	Consolidation	Scenario	Time	View	Account	Flow	Origin	IC	UD1	UD2	UD3	UD4	UD5	UD6	UD7	UD8	Calculation
30,567,141.71	Houston - Houston Clubs	Houston		USD	Actual	2011M2 - Feb 2011 YTD		2000_100 - Third Party Sales	None	Top	Top	Top	Woods	Top	Top	None	None	None	None	
33,693,380.33	Houston - Houston Clubs	Houston		USD	Actual	2011M2 - Feb 2011 YTD		2000_100 - Third Party Sales	None	Top	Top	Top	Irons	Top	Top	None	None	None	None	
11,702,548.09	Houston - Houston Clubs	Houston		USD	Actual	2011M2 - Feb 2011 YTD		2000_100 - Third Party Sales	None	Top	Top	Top	Putters	Top	Top	None	None	None	None	

### Show Data Unit Dimensions

When selected, the upper grid shows the Members for Cube, Entity, Parent, Consolidation, Scenario, Time and View.

### Download Results

These icons   allow a download of the drill results into either a CSV or Excel file. When downloading drill results into a CSV file from the Drill Down page, the list separator in the CSV file is determined by the culture set on your user profile. If the culture on your user profile matches that of the operating system's regional settings, the file opens correctly in Excel without first having to save the file and then importing it into Excel.

### Drill Down History

Every time an administrator drills further down a bread crumb trail of the last action is visible in the Selected Action column. This explains what happened in the drill down process to get to the most recent drill results on the bottom section of the screen. The user can always go back to this area and drill a different way into the data.

The following options become available when a user right-clicks on any field. Not all options will be available for every cell.

#### **Entity Children Contribution (Entity Only)**

This shows how each Entity contributed to the Parent.

#### **Local Currency (Consolidation Only)**

This will show the local currency Entities.

#### **Member Children and Base**

This provides two ways to drill further into a green cell in the top section. The first will drill just into the children of the Member while the other will drill down to the white base cells.

#### **All Aggregated Data Below Cell**

This drills to white cells in each Dimension.

#### **All Stored Data in Data Unit**

This drills into all of the base stored data within the full Data Unit (Cube, Entity, Parent, Consolidation, Scenario, Time and View), so this is a much broader result set than the initially drilled data.

#### **Copy POV from Data Cell**

Select this to copy the Point of View for the selected cell.

#### **Cell POV Information**

Select this to see the full Point of View for the cell.

#### **Cell Status**

Select this to see information about the cell such as if the Members have children, if it is calculated, the lock status, etc.

#### **Calculation Inputs**

This gives details on the formula source accounts for a specific account.

#### **Load Results for Imported Cell**

This will drill back to the source data for the imported cell.

#### **Audit History for Forms or Adjustment Cell**

This will drill back to the source data for a Form or Journal.

**TIP:** OneStream keeps a bread crumb trail of all drill actions in the right side of the top drill screen under the heading Selected Action. Users will never get lost in the data as they can always start over from the top and begin a drill again.

### Origin Audit Drill Down

Perform an Origin Audit Drill in order to see where data was delivered into the system. Right click on the Origin cell, typically Top and select Origin Base. This will reveal all the data that came from each Origin Member. This is now at the Origin Base and any of the following drill down processes can be performed.

#### Import Drill

Right click on the Origin Member Import and select All Aggregated Data. Right click again on the Origin Channel and click Load Results for Imported Cell and then select Navigate to Source Data to get down to the GL Account level. Right click on any cell in the source system data line to get down to the source document that created that line item.

#### Forms Drill

Right click on the Origin Channel Forms and select All Aggregated Data Below Cell. Right click again on the Origin Channel and click Audit History for Forms or Adjustment Cell. This is the specific line item that created the drilled-on value.

**TIP:** To see each line item from the Form used to create the drilled-on value, click the button View All Submitted Data Cells. This is now showing all the line items from the Form used to create the actively drilled line item.

#### AdjInput (Journals) Drill

Right click on the Origin Channel AdjInput and select All Aggregated Data Below Cell. Right click again on the Origin Channel AdjInput and click Audit History for Forms or Adjustment Cell. This is the specific line item that created the drilled-on value. This view shows all Journal entries affecting this cell including previous Journals, deleted Journals and the current Journal.

**TIP:** To see each line item from the Journal used to create the drilled-on value, click the button View All Submitted Data Cells. This is now showing all the line items from the Journal used to create the actively drilled line item.

#### User Defined Description – Drill-Down

The Drill Down window displays the custom User Defined Description as the default header. If no custom description is created, the dimension type label, such as UD1, will display. See Application Properties and then User Defined Dimensions (Descriptions) for more information.

## Using OnePlace Cube Views

---

The screenshot shows the OnePlace Cube View interface with a blue header bar containing a 'Drill Down' icon and two checkboxes: 'Show Data Unit Dimensions' and 'Suppress NoData'. Below the header is a toolbar with icons for 'New', 'Save', 'Print', and 'Close'. The main area is titled 'Drill Down History' and contains a grid with the following columns: Amount, Consolidation, Scenario, Time, View, Account, Flow, Origin, IC, Cost Centers, Product, Sales, Channel, and Customers. A red box highlights the 'Cost Centers' and 'Product' columns. The grid shows one row of data: Amount 2,927,034.61, Consolidation Actual, Scenario 2011M3, Time YTD, View 60000 - Operating Sales, Account None, Flow Top, Origin None, IC None, Cost Centers None, Product UD1 (Cost Centers), Sales None, Channel None, and Customers None.

## Show Cube View as a Report

To generate a smoothly formatted presentation of the Cube View, click this button while in preview mode and a report similar to what one would see in Dashboards will open. Control column widths and row heights from within the Cube View. See Application Tab| Application Properties in order to show a company name and logo on all reports. There are also several application-wide settings for these Data Explorer reports under Application Properties and under the Application Tab| Presentation| Cube Views.

The screenshot shows the Report Viewer window titled 'Report Viewer - Statement of Cash Flow with Adjustments'. The window has a toolbar with various icons for file operations, zoom, and navigation. The main content area displays a report titled 'Statement of Cash Flow with Adjustments' for 'GOLFSTREAM'. The report includes a logo of three stylized figures and a table with four columns: Statement Change, Cash Flow Activity, Reclass, and Adjustments. The table data is as follows:

	Statement Change	Cash Flow Activity	Reclass Adjustments
CashFlowStatement	0.00	0.00	0.00
Cash Flow Check	21,300.91	21,300.91	0.00
Change in Cash			
Net Increase (Decrease) in Cash and Cash Activity	348,671.68	3,657,095.24	-3,308,423.56

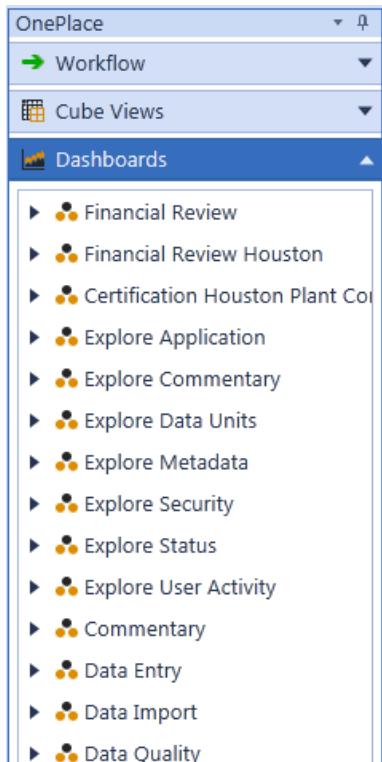
## **Using OnePlace Cube Views**

---

In order to print a Cube View, first generate a Report as shown above. Once the Report is generated, print and export that view to other Formats, such as PDF, HTML, RTF (for Word), CSV, Text, XPS, MHT or Excel.

# Using OnePlace Dashboards

This is the section where all pre-built Application Dashboards can be viewed.



Select a Dashboard and see a complete and organized series of reports, grids, charts, and graphs all combined. A user may be prompted to enter Parameters in order to make the Dashboard relevant. The Parameters are pre-defined when the Dashboard is created in the Application Tab.

**NOTE:** For OneStream Windows App, you must right-click the bookmark to jump to that location.

# Dashboard Toolbar



### Select Parameters

Use this to select specific Parameters when viewing a Dashboard



### Reset Parameter Selections and Refresh Dashboard

Use this to change the Parameters and view the data differently in the Dashboard



### Edit Dashboard

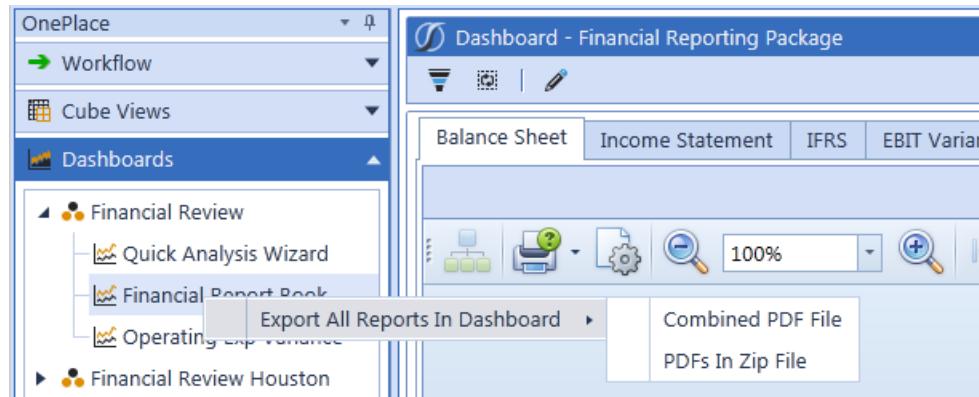
If security permits, select this to launch the Application Tab and make changes to the Dashboard properties and its components.

## Printing Options



Select to print a single Dashboard report from the web. Click the down arrow and select Click via PDF to turn the report into a PDF first and then print.

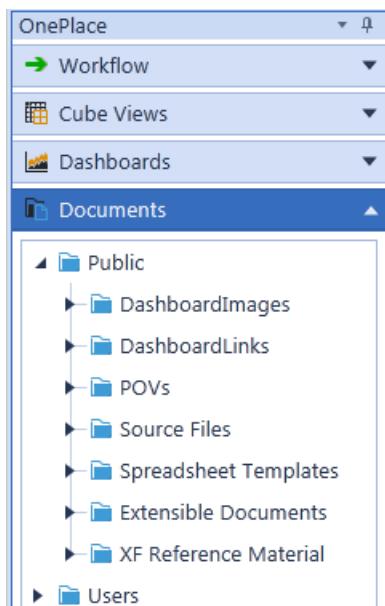
In order to save and print an entire Dashboard Book as one PDF, right-click on the Dashboard name and select Combined PDF File or PDFs In Zip File.



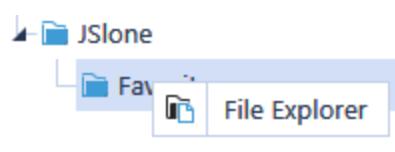
For more information on building Dashboards, see Application Dashboards in Presentation.

# Storing Documents

An administrator can save public documents or templates for users to access for their close process. These documents are saved in the Systems Tab|Documents where only administrators have access. However, a user can access these public documents in OnePlace.



**NOTE:** Right click on any document or folder and select File Explorer in order to upload files from the OnePlace or Systems tab. See File Explorer in "System Tools" on page 894 for more details on this feature.



**NOTE:** Files created with either the Spreadsheet or Text Editor feature are also visible here. These files can be opened by Right clicking the file and selecting one of the three options.

## Storing Documents

---

- Open in Text Editor Page/Open in Spreadsheet Page
- Open- opens file in its related Microsoft application, if loaded on the local PC
- Open With... this allows the user to select which application they want to use to open the file

**NOTE:** The File Explorer option is also available by right clicking a file. Files can also be uploaded or opened from this window.

# About Excel Add-In

The Excel Add-In is an alternative way to enter, update, manage, query and analyze application data, using Excel spreadsheets and workbooks.

You can use the Excel Add-In to:

- Query and submit data into the application.
- Change the cube point of view.
- Interact with forms.
- Leverage multiple cube views.
- Establish a live connection to cube views in the application.
- Drill through to source data.
- Create functions to return, amend, or submit application data.
- Refresh sheets and workbooks when changes occur in the application.

You can launch Excel documents from:

- The Navigation pane
  - The **OnePlace** tab: Through public and user-specific access to application functionality and documents.
  - The **Application** tab: Through the Spreadsheet tool.
  - The **System** tab: Using export features for system administrators.
- Workflow forms that have attached templates.
- Cell or data unit attachments.
- Dashboards

## About Excel Add-In

---

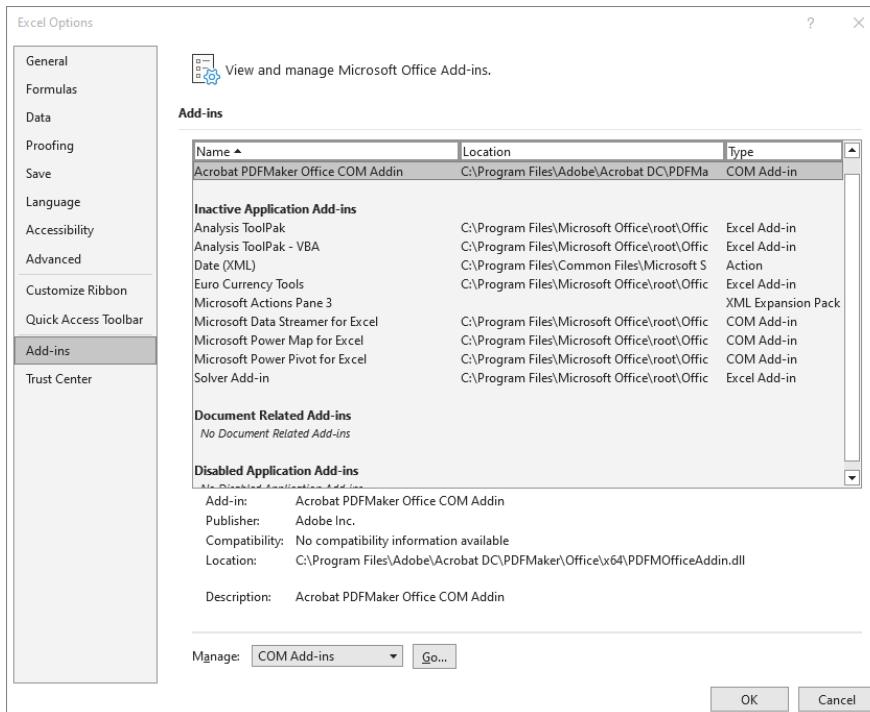
- Cube views
- File Explorer

# Get Started

Before you start using Excel Add-In, you must add the OneStream ribbon to your Excel program.

## Add the OneStream Ribbon

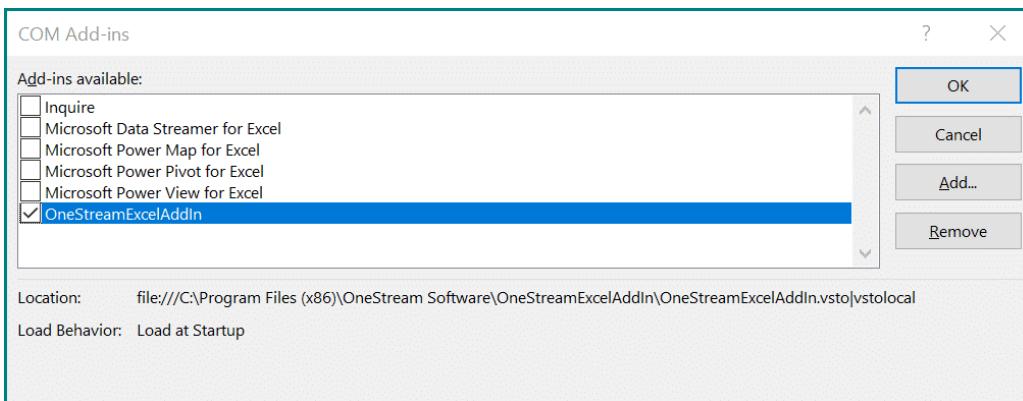
1. In an Excel workbook, navigate to **File > Options** to open the **Excel Options** dialog.
2. Select **Add-ins** from the menu in the left pane. Click the **Manage** drop down menu at the bottom of the dialog box and select **COM Add-ins** and then click **Go**.



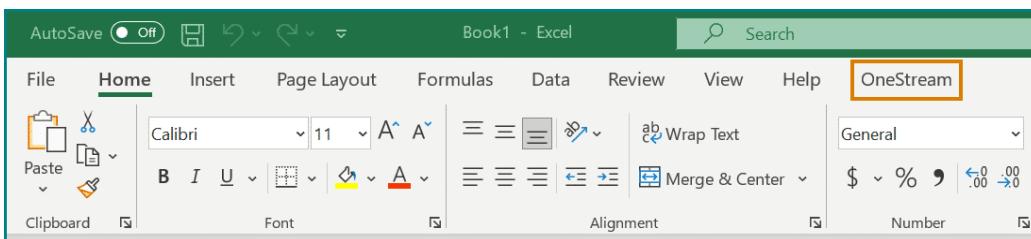
3. Click the box next to **OneStreamExcelAddIn** and then click **OK**.

## About Excel Add-In

---

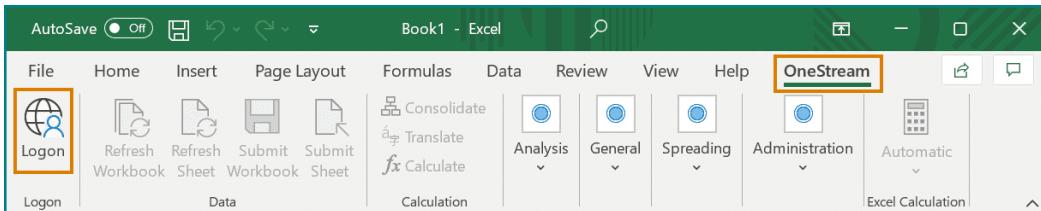


4. The OneStream ribbon is displayed at the top of your workbook page.



## Log Into the Excel Add-In

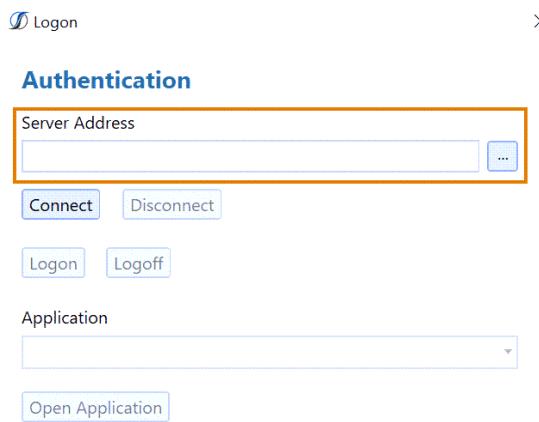
1. From the OneStream ribbon in Excel, click **Logon**.



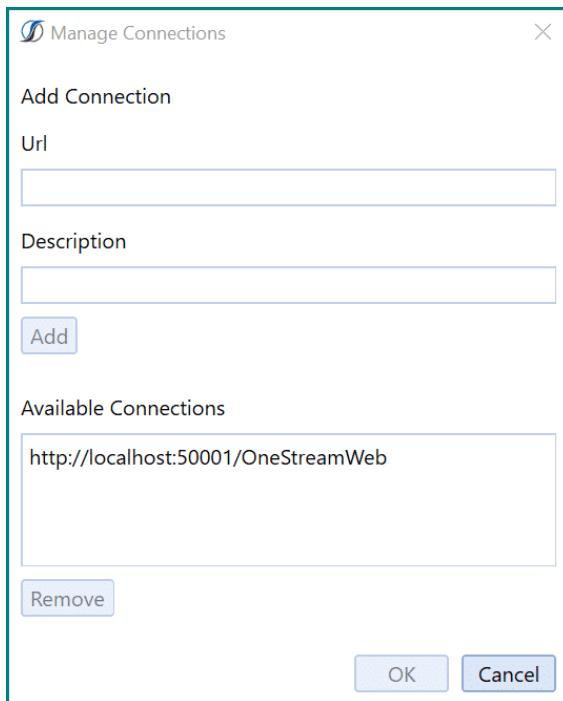
2. In the **Server Address** field, click the ellipsis to add or select the URL for the server.

## About Excel Add-In

---



3. Select an available connection or add a new connection by entering the URL and description and clicking **Add**. After you select a connection, click **OK**.



4. Click **Connect** to authenticate the server.
5. Enter your username and password and click **Logon**.
6. In the application field, select an application and click **Open Application**.

## About Excel Add-In

---

 Logon (7.0.1.13612)

### Authentication

Server Address  
 ...

Connect Disconnect

External Provider Sign In Sign Out

User Name

Password

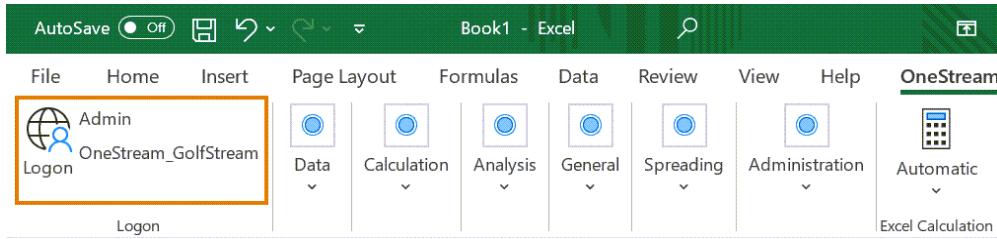
Remember User

Logon Logoff

Application  
 ▼

Open Application

After completing the authentication files and opening the application, your username and application name display in your Excel workbook, indicating that you are logged into the Excel Add-In.



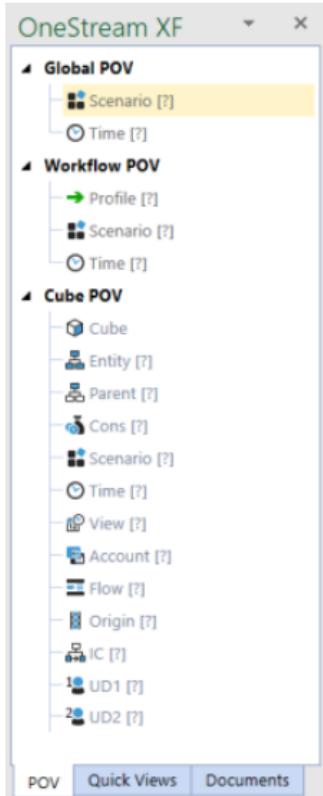
## Update the Excel Add-In

If you use the Excel Add-In provided with the Windows application, update the Excel Add-In as part of your client update. See "Client Updater" on page 768. Otherwise, update the Excel Add-In by uninstalling then reinstalling it as described in the *Installation Guide*.

## Excel Add-in Task Pane

The Excel Add-in **Task** pane includes the **POV**, **Quick Views** and **Documents** tabs as shown below.

- **POV:** Identifies the **Global POV**, **Workflow POV**, and **Cube POV**. This tab is linked to the POV in your application. From here, you can make updates to the cube POV.
- **Quick View:** Lets you build and edit quick views in the Excel Add-in. See "Quick Views" on the next page.
- **Documents:** Displays public or user-specific application or system documents. Access and manage documents with the "File Explorer" on page 1048.



## Quick Views

Quick views are ad hoc reports that allow you to pivot, drill back, create data sets, and design workbooks to analyze data. Quick views can also be used to submit data. You can create a quick view by:

- Entering members.
- Copying and modifying another quick view.

Quick views function similarly to cube views.

## Manage Quick Views

The **Quick View** tab displays your dimensions. You can move and replace the default Time and Account dimensions. This table identifies the tasks you can perform on the **Quick View** tab.

Button	Description
	Create a quick view. See "Quick Views" above.
	Rebuild a quick view created by direct data entry by selecting new or modified fields.
	Rename a quick view.
	Delete a quick view.
	Edit quick view properties.

## About Excel Add-In

---

Button	Description
	Refresh a quick view with the latest data set.
	Undo up to 100 changes.
	Restore deleted changes.
	Select members, identified by dimension type or dimension, to add to a quick view.
	Clear everything except selected items.
	Clear only a selected item.
	Go to the next level in the hierarchy.
	Go to the top of the hierarchy.
	Go to the parent member.

Button	Description
	Go to the immediate child members.
	Go to the branch base.

## Quick View POV

If you do not select a member, the quick view POV reflects your default POV settings. User-defined descriptions also display in the quick view POV. Hover over dimension labels to view dimension descriptions.

See "Application Properties" on page 731 and "User Defined Dimensions 1-8" on page 363

## Create and Modify Quick Views

This topic describes how to define and modify quick views.

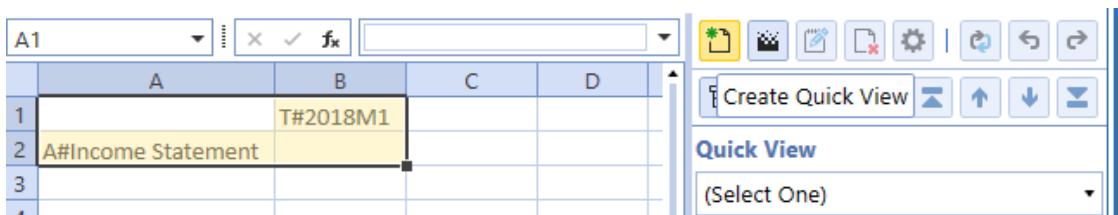
### Create Quick Views

Create a quick view using a sample income statement account. See "Find and Add Members to Quick Views" on page 1019 and "Manage Quick Views" on page 1007

1. In Cell **A2** specify the row: **DimensionToken#MemberName A#IncomeStatement**.
2. In Cell **B1** specify the column: **DimensionToken#MemberName T#2018M1**.
3. Highlight the dimensions you entered and click **Create Quick View**. If you are using the Excel Add-in, click the **Quick Views** tab.

## About Excel Add-In

---



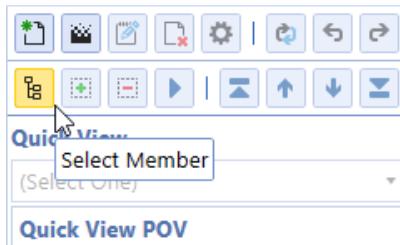
4. In the **Create Quick View** window, click **OK**.

You can create quick views using the Quick Views menu on the OneStream ribbon.

1. On the OneStream ribbon, click the **Quick Views** drop-down menu.
2. Click **Create Quick View** and one will populate in the cells below.

To add members to your new quick view, manually enter the dimension tokens into the cells or add them using the Quick View toolbar.

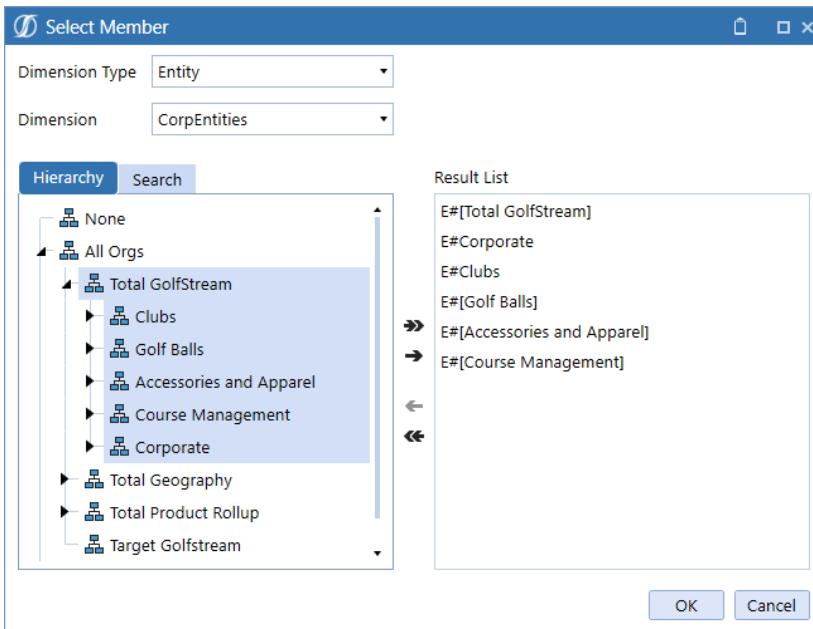
1. In cell **B1** specify the column: **DimensionToken# MemberName S#Actual**
2. In cell **C1** specify the column: **DimensionToken#Member Name S#BudgetV2**
3. Click on the cell where your quick view rows will begin. In this example, specify column **A2**.
4. Click **Select Member** from the quick view toolbar to the right of the screen.



5. From the Select Member dialog box, move dimensions to include in your quick view to the **Result List** using the single arrow. Click **OK**.

## About Excel Add-In

---



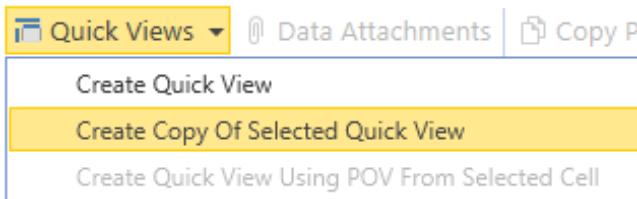
6. Highlight the cells with your data and click **Create Quick View**.

A	B	C
1	S#Actual	S#BudgetV2
2 E#None		
3 E#Atlanta		
4 E#AtlantaClothing		
5 E#AtlantaFootwear		
6 E#AtlantaEyewear		

You can create a quick view by copying an existing quick view or expanding on the data in a quick view cell.

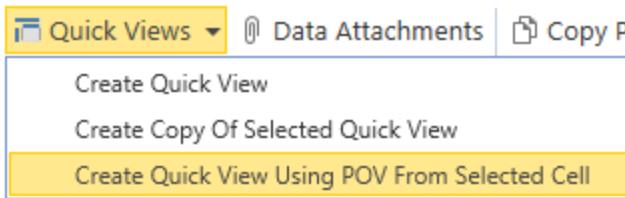
1. Select an existing quick view.
2. On the OneStream ribbon, click the **Quick Views** drop-down menu.

3. Select **Create Copy of Selected Quick View**.



To create a quick view based on the POV from a single cell in an existing quick view:

1. Select an intersection of data in an existing quick view.
2. On the OneStream ribbon, click the **Quick Views** drop-down menu.
3. Select **Create Quick View Using POV From Selected Cell**.



## Modify Quick Views

The following section outlines how to rebuild quick views after adding or removing dimensions and members.

Use a comma-separated member list in column and row cells to build the quick view. In this example, enter:

1. **DimensionToken#Root: A#60999,64000,63000,65200** in **A2**
2. **DimensionToken#Root: T#2018M1,2018M2,2018M3** in **B1**
3. Select the updated area and click **Create Quick View**.
4. Enter a quick view name and click **OK**.

## About Excel Add-In

---

The quick view is displayed:

A screenshot of Microsoft Excel with the ribbon menu open. The active sheet is titled "QuickView\_Quic". The data area contains a table with columns labeled "A", "B", "C", "D", and "E". Row 1 has entries "2018M1", "2018M2", and "2018M3". Rows 2 through 5 contain financial data: "60999 - Net Sales" (33,104,825.42), "64000 - Earnings Before Taxes" (5,448,448.66), "63000 - Earnings Before Interest and Taxes" (6,047,359.91), and "65200 - Extraordinary Income/Expense (net of tax)" (-88,375.00). The right side of the screen shows the OneStream XF ribbon bar with various icons and a toolbar for quick view operations.

Use the member expansion functions in cells to refine the quick view. For example, enter:

1. **DimensionToken#Root A#Root.Children in A2**
2. **DimensionToken#Root T#2018.Base in B1**
3. Select the updated area, click **Create Quick View** and specify a name.

A screenshot of Microsoft Excel showing the results of the member expansion functions. The data area now includes columns for months from "Jan 2018" to "Jul 2018". The first row contains the month names. Rows 2 through 4 show the expanded members: "None", "GAAP Account Structure", and "IFRS Account Structure". The right side of the screen shows the OneStream XF ribbon bar with various icons and a toolbar for quick view operations.

Use the member expansion functions to enter comma-separated members in cells to finalize the quick view. For example, enter:

1. **DimensionToken#Root A#Root.List(60999,64000,63000,65200) in A2**
2. **DimensionToken#Root T#2018M1 in B1**
3. Select the updated area and click **Create Quick View**.

The quick view updates:

A screenshot of Microsoft Excel showing the final refined quick view. The data area now displays specific values for each item: "60999 - Net Sales" (76,263,853.88), "64000 - PreTax Income" (10,534,767.03), "63000 - Earnings Before Interest and Taxes" (11,957,754.03), and "65200 - Extraordinary Income/Expense (net of tax)" (-212,100.00). The right side of the screen shows the OneStream XF ribbon bar with various icons and a toolbar for quick view operations.

**NOTE:** Check the suppression options if items do not display as expected.

### Extend a Quick View

To add a dimension or new dimension member to a quick view, enter # before the specified dimension or member names. If the dimension is already used, # is not required.

1. Specify the data to use. For example, enter:

Dimension type, such as **DimensionToken#MemberName S#Actual**

2. Select the area and click **Rebuild Quick View**.

The screenshot shows a Microsoft Excel spreadsheet with a table. The formula bar at the top has 'S#Actual' entered. The table has three rows: Row 1 contains 'Houston' and 'Income Statement'; Row 2 contains 'Jan 2018'; Row 3 contains 'S#Actual'. The 'Quick View' ribbon tab is selected, and the status bar at the bottom right says 'Quick View'.

The updated quick view is displayed:

The screenshot shows the same Microsoft Excel spreadsheet after rebuilding the quick view. The formula bar now shows 'Actual'. The table structure is identical to the previous screenshot, with rows for Houston, Income Statement, and Jan 2018, and the third row containing 'Actual'.

### Change Dimension Members

You can switch dimension members by entering another member name. You do not need to specify a dimension token if you switch to members in a dimension that is being used. For example:

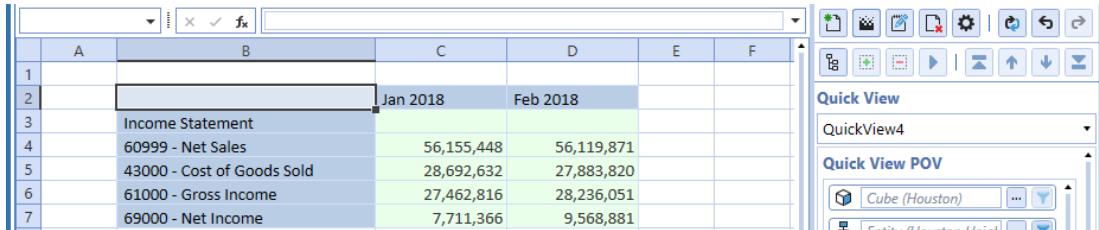
1. Enter these members: **60999**, **43000**, and **61000** .
2. Select the members and click **Rebuild Quick View**.

The screenshot shows a Microsoft Excel spreadsheet with a table. The formula bar at the top has '60999' entered. The table has several rows: Row 1 contains 'Houston' and 'Income Statement'; Row 2 contains 'Jan 2018' and 'Feb 2018'; Row 3 contains '69000 - Net Income'. The 'Quick View' ribbon tab is selected, and the status bar at the bottom right says 'Quick View'.

## About Excel Add-In

---

Members **60999**, **43000**, and **61000** replace **60000** and **54500** as shown in the updated quick view:

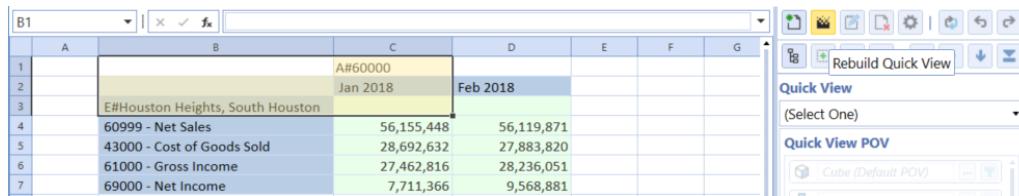


A	B	C	D	E	F
1					
2	Income Statement	Jan 2018	Feb 2018		
3	60999 - Net Sales	56,155,448	56,119,871		
4	43000 - Cost of Goods Sold	28,692,632	27,883,820		
5	61000 - Gross Income	27,462,816	28,236,051		
6	69000 - Net Income	7,711,366	9,568,881		
7					

You can change from one dimension to another by entering comma-separated members in the same cell.. In the example below, the Account dimension is changed to the Entity dimension.

1. Above the **Time** dimension enter the following, separated by commas:

- **E#HoustonHeights**
- **South Houston**
- **Operating Sales A#60000**



A	B	C	D	E	F	G
1		A#60000				
2		Jan 2018	Feb 2018			
3	E#Houston Heights, South Houston					
4	60999 - Net Sales	56,155,448	56,119,871			
5	43000 - Cost of Goods Sold	28,692,632	27,883,820			
6	61000 - Gross Income	27,462,816	28,236,051			
7	69000 - Net Income	7,711,366	9,568,881			

2. Select the area and click **Rebuild Quick View**.

**Operating Sales** columns are created for the Houston entities:



A	B	C	D	E	F	G
1						
2		Operating Sales	Operating Sales			
3		Jan 2018	Feb 2018			
4	Houston Heights	57,982,342	57,944,272			
5	South Houston	78,299,764	73,262,764			

You can add to an existing dimension's members by entering comma-separated member names in the same cell. If names are numeric or start with an apostrophe, insert a space after each comma.

## About Excel Add-In

---

1. In the same cell, enter the following members. Insert a space after each comma because the member names are numeric.

- **60999**
- **43000**
- **61000**

2. Select the area and click **Rebuild Quick View**.

A	B	C	D	E	F	G
1						
2		Jan 2018	Feb 2018			
3	Income Statement					
4	60000 - Operating Sales	34,070,526	81,769,263			
5	<b>60999, 43000, 61000</b>	15,590,644	37,417,545			
6	54500 - Total Operating Expenses	8,184,706	19,643,294			
7	69000 - Net Income	(2,732,060)	4,564,485			
8						
9						
10						

Members 60999, 43000, and 61000 are added to the existing member list:

A	B	C	D	E	F	G
1						
2		Jan 2018	Feb 2018			
3	Income Statement					
4	60000 - Operating Sales	34,070,526	81,769,263			
5	60999 - Net Sales	33,104,825	79,451,581			
6	43000 - Cost of Goods Sold	17,514,182	42,034,036			
7	61000 - Gross Income	15,590,644	37,417,545			
8	54500 - Total Operating Expenses	8,184,706	19,643,294			
9	69000 - Net Income	(2,732,060)	4,564,485			

## Change Dimensions

Dimensions can be replaced by modifying row and column headers with the updated dimension information and rebuilding the quick view.

1. Click the dimension to be changed, then specify the dimension to use. As shown in the example below, you could:

## About Excel Add-In

---

- a. Replace **Account** with an **Entity** such as **E#HoustonHeights**.
  - b. Add **OperatingSales A#60000** above **Time**.
2. Select the area and click **Rebuild Quick View**.

The screenshot shows the Microsoft Excel ribbon. The 'Quick View' tab is selected, and the 'Rebuild Quick View' button is highlighted with a yellow box. The main worksheet area displays a table with columns for Entity, Time, and Value. The first row has 'A#60000' in column B and 'Jan 2018' in column C. The second row has 'E#Houston Heights' in column B and 'Feb 2018' in column C. The third row contains the data for '60000 - Operating Sales' across Jan 2018 and Feb 2018. The fourth row contains '61000 - Gross Income'. The fifth row contains '54500 - Total Operating Expenses'. The sixth row contains '69000 - Net Income'.

The updated quick view will look like this:

The screenshot shows the Microsoft Excel ribbon. The 'Quick View' tab is selected, and the 'Rebuild Quick View' button is highlighted with a yellow box. The main worksheet area displays a table with columns for Entity, Time, and Value. The first row has 'Operating Sales' in column B and 'Operating Sales' in column C. The second row has 'Jan 2018' in column B and 'Feb 2018' in column C. The third row contains the data for 'Houston Heights' across Jan 2018 and Feb 2018. The fourth row contains '57,982,342' in column B and '57,944,272' in column C.

## Using "Keep Only" When Extending a Quick View

When modifying a quick view, select the rows and columns you want to remain unchanged using the **Keep Only** option. This allows you to clear a quick view without losing selected data.

1. In the row below an existing dimension, add these members: **60000**, **61000**, and **54500**.
2. Select the area and click **Keep Only**.

The screenshot shows the Microsoft Excel ribbon. The 'Quick View' tab is selected, and the 'Keep Only' button is highlighted with a yellow box. The main worksheet area displays a table with columns for Entity, Time, and Value. The first row has 'Actual' in column D and 'Actual' in column E. The second row has 'Jan 2018' in column D and 'Feb 2018' in column E. The third row contains the data for 'Houston' across Jan 2018 and Feb 2018. The fourth row contains 'Income Statement'. The fifth row contains '69000 - Net Income'. The sixth row contains '60000'. The seventh row contains '61000'. The eighth row contains '54500'.

## About Excel Add-In

---

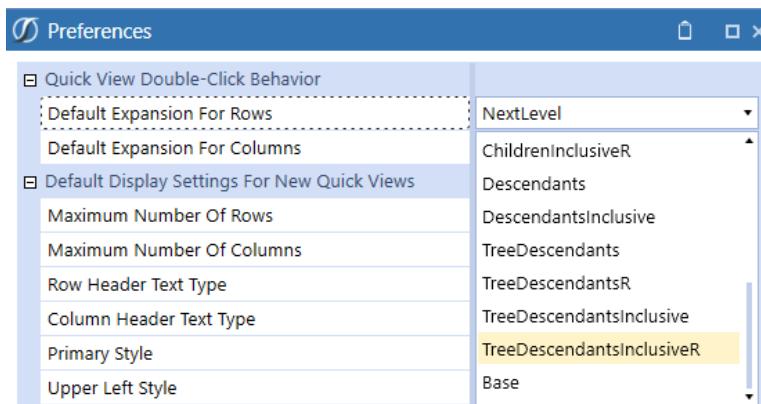
After updating, the quick view will look like this:

The screenshot shows the Microsoft Excel ribbon with the 'Quick View' tab selected. The ribbon bar includes tabs for Home, Insert, Page Layout, Formulas, Data, Page Break Preview, and Quick View. Below the ribbon is a worksheet area with data in columns A through F. To the right of the worksheet is a ribbon side panel titled 'Quick View' which displays 'QuickView4' and 'Quick View POV' sections. The 'Quick View POV' section shows 'Cube (Houston)' and 'Parent (Houston)'.

## Reverse Member Display Order

You can double-click to reverse the display order of members by specifying **Quick View Double-Click** preferences.

1. Select **Administration > Preferences**.
2. In **Quick View Double-Click**, set **Default Expansion for Rows** to **TreeDescendantsR**.



## Format and Suppress Data

Use the following to manage and customize your quick view display:

- **Row Header/Columns Header Text Types:** Reformat columns and rows.
- **Suppress Repeating Member Names:** Display or hide repeating member names. For example, if False, the repeating name displays in each row.
- **Data Style:** Apply a provided or custom data style. See "Styles" on page 1071.

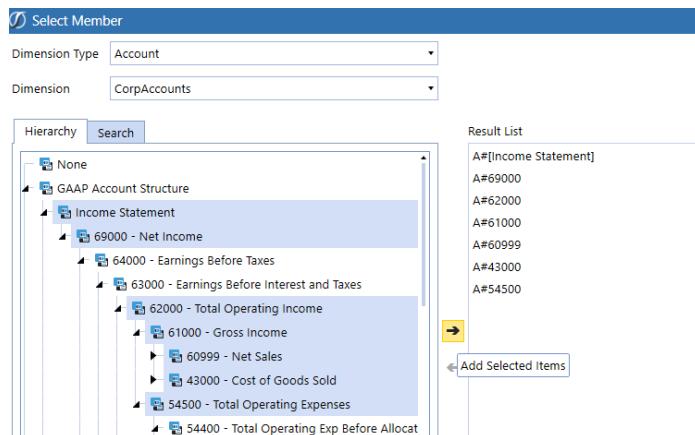
- **Suppress Invalid Rows/Columns:** Hide cells that have invalid data.
- **Zero Suppression Threshold:** Suppress all values that are less than a number you specify, for rounding purposes. For example, specifying 499.99 suppresses any value less than that, such as 499.19.

## Find and Add Members to Quick Views

You can add multiple members to a quick view, filtering and searching as necessary.

**NOTE:** To add members while preserving the current quick view column, select the columns to fill outside the current quick view range. After adding members, select the new column range and click **Rebuild Quick View**.

1. Select a dimension, then click **Select Member**.
2. On the **Select Members** dialog box, perform a task:
  - To find members not shown in the hierarchy, click the **Search** tab, enter the member name, select members, and then click **Add Selected Items**.
  - To select multiple members, press **Ctrl** or **Shift**, selecting the members, and then click **Add Selected Items**.



3. Select the area to which the members were added and click **Rebuild Quick View**.

# Cube Views in Spreadsheet

Cube views are the primary reporting tool in OneStream to display financial data for your organizational needs. You can also use cube view reporting in the Spreadsheet and Excel Add-in using cube view connections.

## Define Cube View Connections

Cube view connections are different from exporting a cube view to Excel because the connection remains live, meaning data can be refreshed and submitted.

Before defining cube view connections:

- Make sure you have the appropriate security settings to access your Cube View and the Spreadsheet page in OneStream.
- Make sure your cube view is in a group that is in a profile that has visibility set for Excel.
- If using the Excel Add-in, connect to OneStream. In Excel, from the OneStream menu, click Logon.

To add a cube view to Excel or Spreadsheet:

1. Select the cell that you want to place the cube view on. The uppermost left corner of your cube view will start on the cell you have selected.
2. Select **Cube Views > Cube View Connections > Add** from the OneStream ribbon.

## Configure Cube View Connections

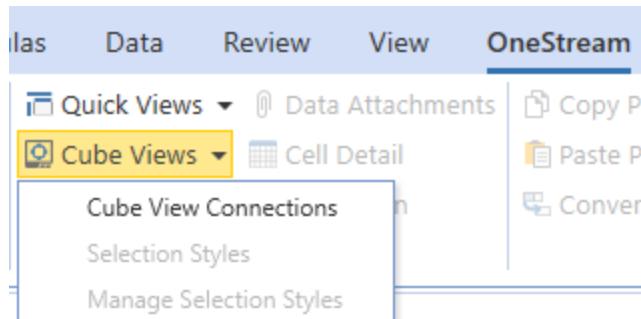
You can configure cube view connections in OneStream from the Spreadsheet functionality.

1. Click **OneStream** from the **Spreadsheet** ribbon.
2. From **Cube Views**, select **Cube View Connections** to open the cube view connections

## About Excel Add-In

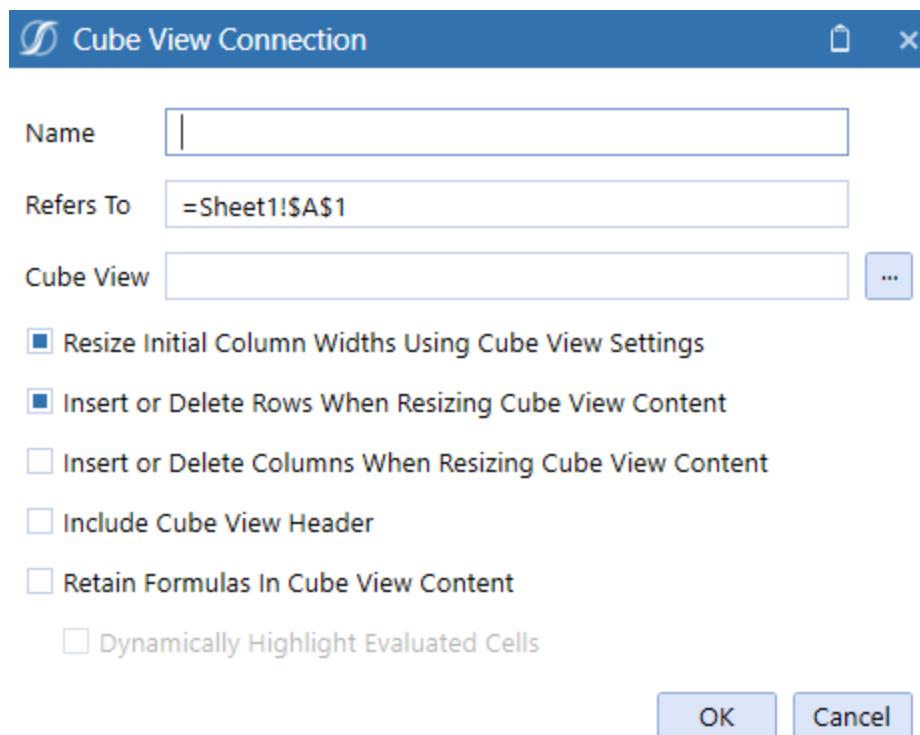
---

dialog box.



## Add a Cube View to Excel or Spreadsheet

1. Click **Add** to create a cube view connection.



2. Click the ellipsis to open the **Object Lookup** dialog box. Object Lookup enables you to search for an existing cube view.

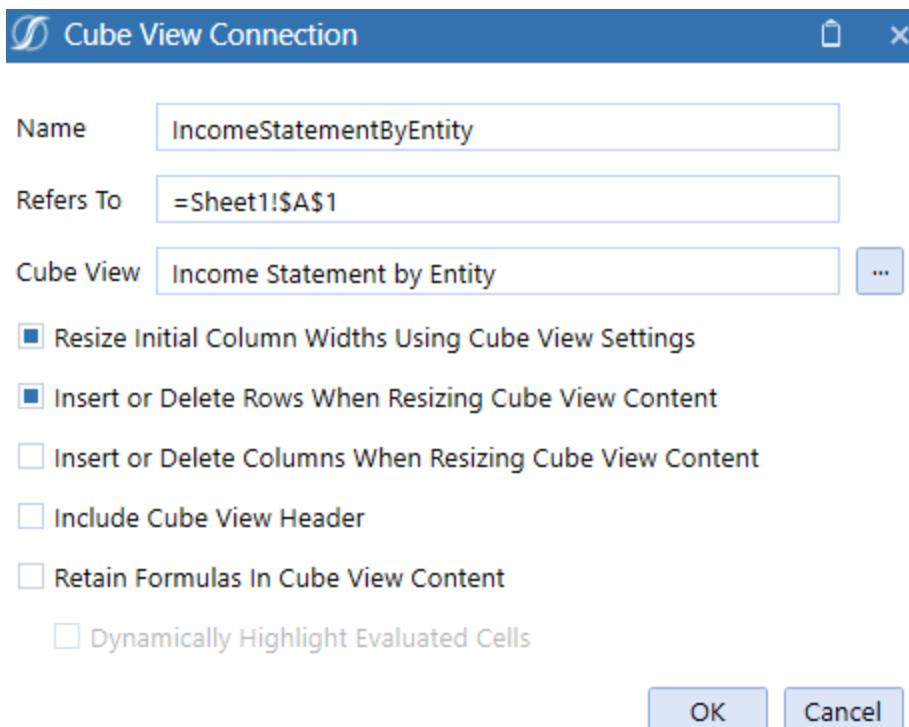
## About Excel Add-In

---

3. In this example, select **Spreadsheet > Income Statement by Entity Cube View**. The **Name** and **Refers To** fields automatically populate for the selected cube view.



4. Click **OK** to confirm the cube view selection.
5. Click **Close** to exit the dialog box. The Income Statement by Entity cube view is added to the spreadsheet.



**NOTE:** Refer to the table below for more context on these properties.

Property	Description
Name	The name is created

## About Excel Add-In

---

	for you if you select the cube view first. If you type a name first and then select the cube view, the typed name is retained.
Refers To	This setting refers to the initial cell in your spreadsheet that your cube view will start on. This becomes the default cell for the cube view connection.
Cube View	Click the ellipsis to look up and add a cube view to make a cube view connection with.
Resize Initial Column Widths Using Cube View Settings	Selected by default, this option automatically resizes column widths based on the settings from the cube view selected.
Insert or Delete Rows When Resizing Cube View Content	Select this option if you plan to stack everything vertically on your worksheet. This automatically adds or deletes rows as a cube view expands and

## About Excel Add-In

---

	contracts.
Insert or Delete Columns When Resizing Cube View Content	Select this option if you plan to stack everything horizontally on your worksheet. This automatically adds or deletes columns as a cube view expands and contracts.
Include Cube View Header	This option includes the header on the cube view.
Retain Formulas in Cube View Content	This option allows you to enter formulas in the cube view (Excel or Spreadsheet) and retain those formulas before and after submission of the sheet or workbook.
Dynamically Evaluate Highlighted Cells	This option is only available if the previous option is selected. This highlights a cell if the formula reference has changed without having to refresh a spreadsheet.

## Format Cube View Connections

In a spreadsheet, you can apply standard formatting to a cell, a group of cells, or to a row or column. However, for an applied cube view connection, the formatting reverts to the cube view default Row and Column settings after you refresh the sheet or workbook.

You can use Selection Styles and Named Ranges to format a cube view connection in a spreadsheet.

### Manual Format Example - Selection Styles

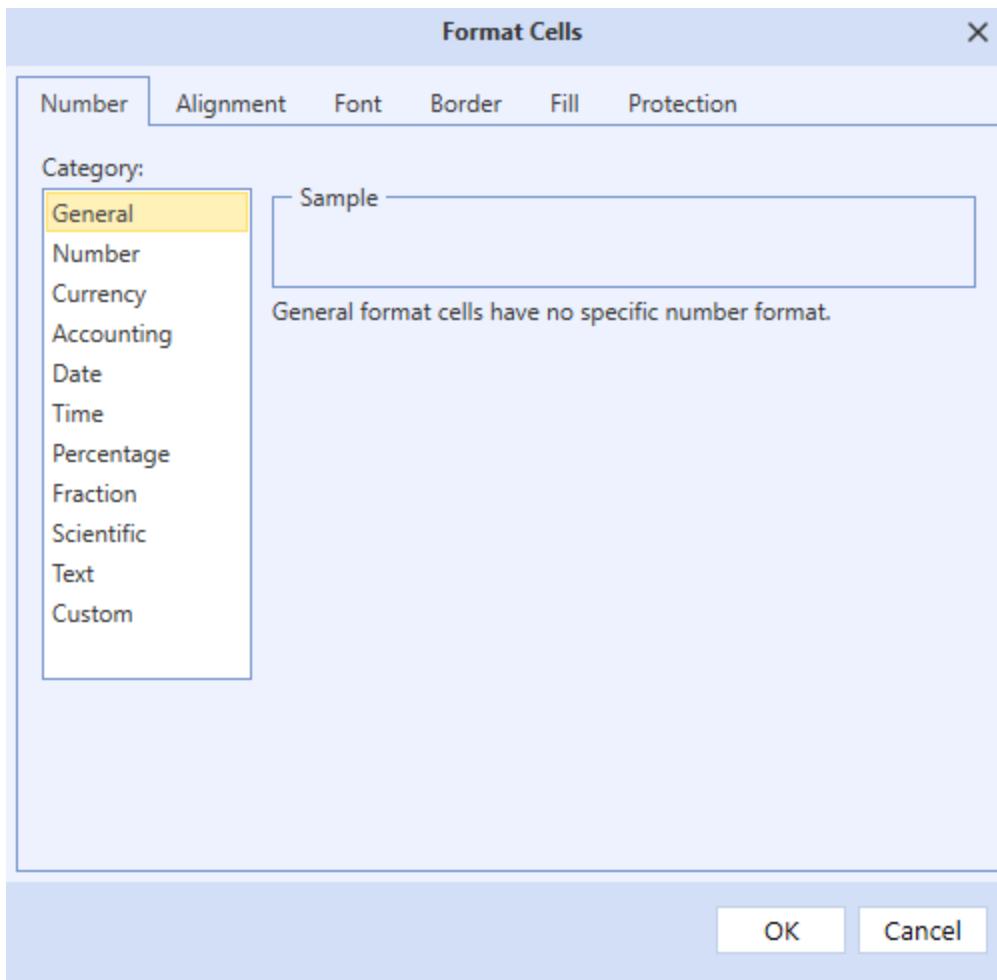
1. Highlight a section of a cube view. In this example, cells A2:D3 are highlighted.

	A	B	C	D
1		Detroit	Rochester	Cleveland
2	Sales	168764	188021	189577
3	Net Revenue	168764		

2. To change the background color, right-click on the highlighted cells and select **Format Cells**. The Format Cells dialog box is displayed.

## About Excel Add-In

---



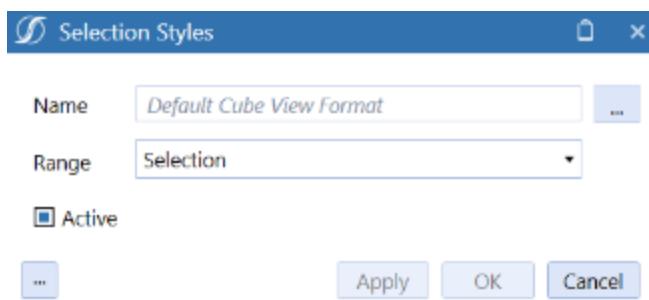
3. Click **Fill**.
4. Select a color and click **OK**.

	A	B	C	D
1		Detroit	Rochester	Cleveland
2	Sales	168764	188021	189577
3	Net Revenue	168764		

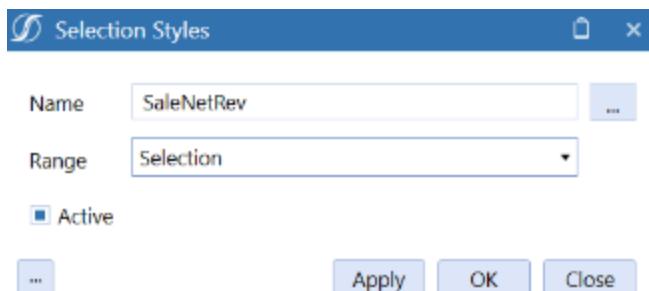
After the formatting displays, you can save this background color as a selection style. You can also add styles to apply changes to rows, columns, or cells, or a combination of existing styles with new styles. There are also options to apply conditional formatting.

## Creating a Selection Style

1. Format selected cells.
2. To save formatted cells for the cube view connection, select **Cube Views > Selection Styles**.
3. Name the style. In this example, the name is SaleNetRev.
4. From **Range**, choose **Selection**.



5. Click **OK** to save the SaleNetRev selection style.



You can apply this style to existing cells by selecting the style. You can enable and disable styles from the **Cube Views > Manage Selection Styles** drop-down.

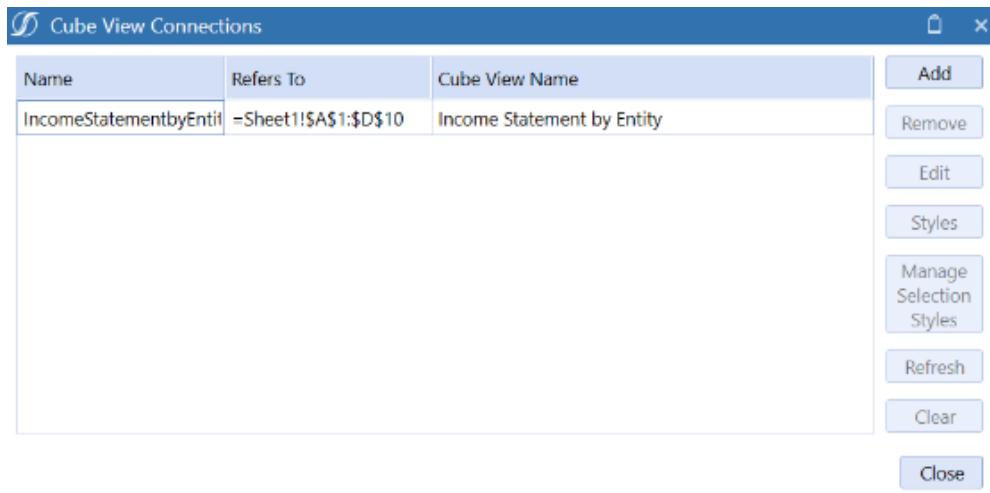
## Named Ranges Example

Styles can be updated for a specific named range on a cube view in Spreadsheet.

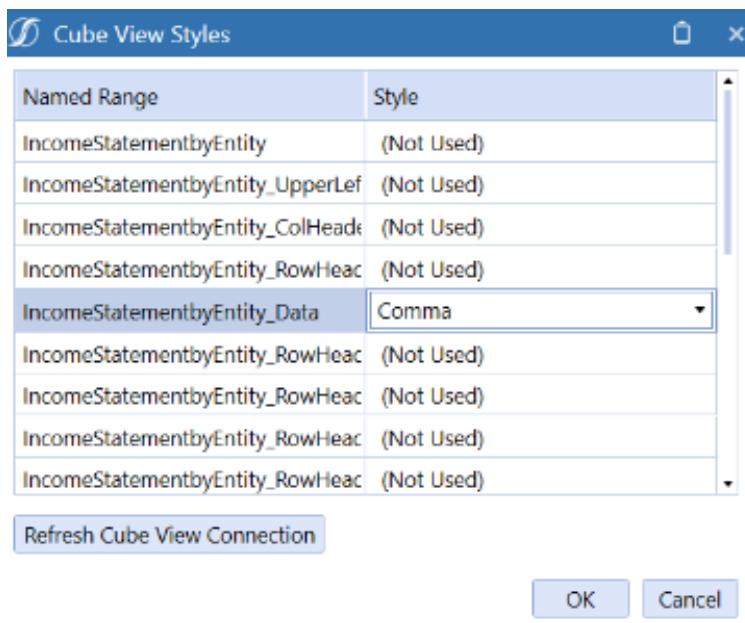
## About Excel Add-In

---

1. From **Cube Views**, select **Cube View Connections**.
2. Select **IncomeStatementbyEntity** and then click **Styles**.



3. In the Style cell for **IncomeStatementbyEntity\_Data**, select **Comma** and then click **OK**.

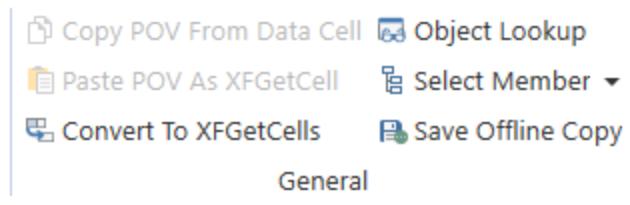


4. Click **Close**. In this example, the named range was changed to a comma style.

Sales	168,763,885.00	188,020,810.00	189,557,136.00
Net Revenue	168,763,885.00	188,020,810.00	189,557,136.00
Cost Of Goods Sold	99,121,875.00	87,033,882.00	115,427,342.00
IC Cost of Goods Sold	31,183,482.00	31,833,861.00	31,427,791.00
<b>Cost of Goods Sold</b>	<b>130,305,357.00</b>	<b>118,867,743.00</b>	<b>146,855,133.00</b>
Total Operating Expenses	36,108,044.25	36,710,676.75	36,305,982.00
Operating Expenses	36,108,044.25	36,710,676.75	36,305,982.00
Gross Profit	38,458,528.00	69,153,067.00	42,702,003.00
<b>Net Income Before Interest</b>	<b>2,350,483.75</b>	<b>32,442,390.25</b>	<b>6,396,021.00</b>

## XF Retrieves

In an existing quick view, you can convert to formula-based retrieves (XFGGetCell) on intersections of data using the **Convert To XFGGetCells** option in the top ribbon toolbar.



After selecting **Convert To XFGGetCells**, OneStream displays the following prompts:

- Are you sure you want to convert all the data in Quick View 'Name of the Quick View' to XFGGetCells?
- By clicking **OK**, the Quick View definition will be deleted and specific intersections of data will be converted to XFGGetCell formulas.

This is a great way to set up a lot of formulas at once. However, these will hardcode all the dimension members in your formulas. If the member name is being displayed, the rows and columns can be synchronized to the pre-made quick view and cube view rows and columns. Your quick view can be adjusted to display the member name instead of the description through the quick view options.

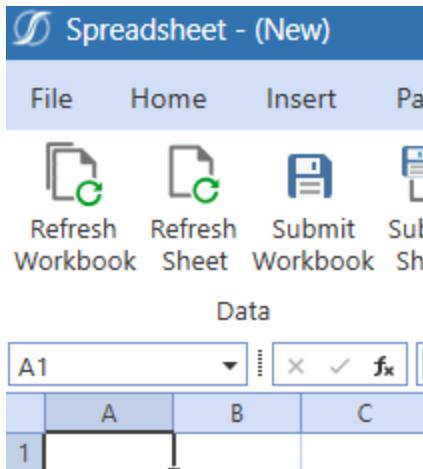
**NOTE:** Quick view data that has been converted to XFGGetCell cannot be converted back into a quick view. However, quick view content may be saved as a template before converting.

Additionally, XF Functions can be manually entered:

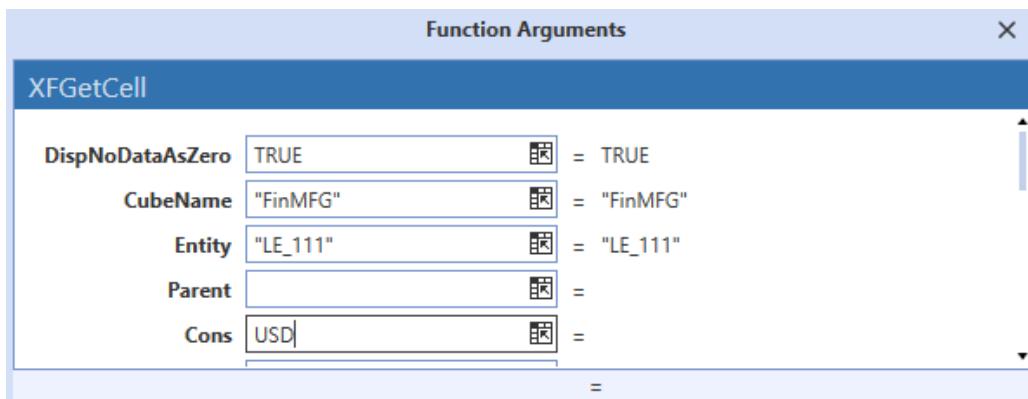
## About Excel Add-In

---

1. Select a cell and click on the insert function icon.
2. Select an XFGGet function from the pop-up menu.



3. Fill out the necessary dimensions to build your XF Retrieve. If you do not want to fill one out, place a "" in the property and it defaults to the data in your cube POV.



4. Click **OK** to save the dimensions entered and the selected cell data changes to #REFRESH.
5. If your query is in the correct syntax, data populates in the cell after refreshing the sheet.

**NOTE:** If a member name has been misspelled, the cell returns an error calling out the first misspelled member name.

## Absolute Cell References

Absolute cell references are a functionality in the Excel Add-in that allows an exact cell reference to be kept when you copy the formula to another cell. To do this, put a \$ in front of the portion of the cell reference that you want to keep.

For example, if you want to keep the entire A1 cell reference to copy to another cell you would change the cell reference to \$A\$1. If you only want to keep the column, the cell reference would be \$A1 and if you only want to keep the row it would be A\$1.

## Relative Cell References

Relative cell references are the default state of cells in an Excel workbook. The reference is relative to the location of the cell. Formulas copied to new cells retain the location information and apply it to the new set of cells.

For example, if you reference A1 in cell C1 and then copy the formula into D1, the results will be a reference to B1 in cell D1 because the formula is referencing 2 columns before.

## Table Views

A Table View definition for the Windows Application Spreadsheet Tool is defined in a Business Rule. The Administrator designing the rule can define the rows and columns which should be returned to the Spreadsheet from the source table presented in the Table View.

The Table View Business Rule can collect data from multiple data sources. For example, a single Spreadsheet worksheet can display a Table View which collects data from two or more sources.

The Administrator has full control over the write back “save” process through Business Rules. When designing the Table View Business Rule, the BRAPI Authorization functions should be designed into the Business Rule to control access to the viewing or modifying the data. This can be applied to the entire table or to specific cells. A workbook can contain multiple Table Views. These can be on the same worksheet or across worksheet pages.

A single Business Rule file can be used to define multiple Table Views by calling the Business Rule argument, TableViewName. Additionally, a single named range can be used to manage table data cells within the Spreadsheet using user defined named ranges (XFTV\_ \*).

**NOTE:** When browsing for Table Views Business Rule, only Spreadsheet Type Rules will be displayed in the Object Lookup dialog.

## Table Views Spreadsheet Control

Table Views is a OneStream Windows Application only spreadsheet control. The primary purpose of Table Views is to provide an easier method to creating Dashboards where relational data is required. The use of Table Views in Spreadsheet enables the designer to work in a more flexible environment to design a form or data collection tool.

Accessed via the OneStream Windows Application Spreadsheet, the Table Views provide a client-based tool to support Dashboard forms. Table Views are not intended as an alternative to other tools, such as the SQL Table Editor or Grid Viewer, Dashboard Components.

### Key Use

- Tool kit item for advanced Marketplace or dashboard designers
- Designed to collect records from relational tables, or other sources
- Present the information in the Spreadsheet format
- Utilize client-side functionality, found in the Spreadsheet tool, such as calculations and pick-list validation lists
- Update table records only

### Design Considerations

- The current functionality is designed to update records in target tables
- Does not perform inserts, to create new records
- Controlling elements must be designed into the Table View Business Rule by the creator to ensure data integrity, security and performance

### Table View Size Considerations

- Spreadsheet support of Table Views depends upon the number of rows and row content
- The Spreadsheet Control does not support paging, therefore all rows and content must be returned

## About Excel Add-In

---

- Performance testing and design expectations is to support approximately 8000 KB of data per Table View.

# Navigating the Excel Add-In

## OneStream Ribbon

After installing the OneStream Add-In, there will be a OneStream menu item and a ribbon in Excel.

The ribbon is organized as follows:

### Logon



This displays the current user and application. A user can logon to a different application by clicking this icon.

### Data

#### Refresh Workbook

This pulls down updated data from the server and refreshes the entire Excel workbook.

#### Refresh Worksheet

This pulls down updated data from the server and only refreshes the selected worksheet.

#### Submit Workbook

After editing data in Excel, click this icon to send it back to OneStream. This icon will send data back for every tab in the Excel workbook.

#### Submit Sheet

After editing data in Excel, click this icon to send it back to OneStream for Cube Views, Quick Views, XFSetsCells, and Table Views on the active sheet.

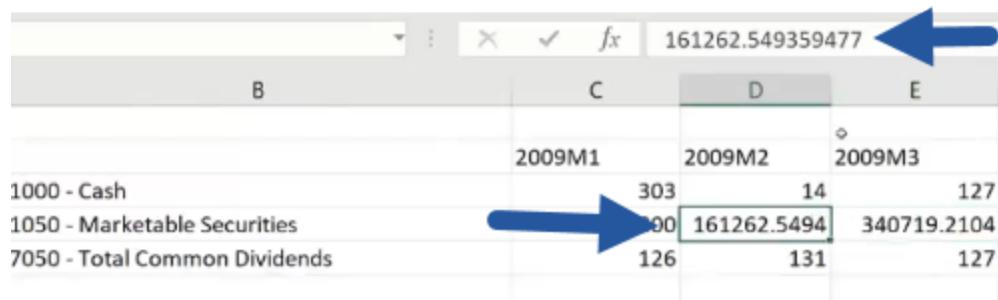
## Navigating the Excel Add-In

---

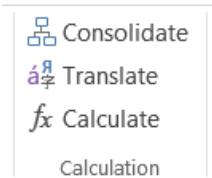
	Refresh Sheet	Refresh Workbook	Submit Sheet	Submit Workbook
Function Behavior	Refreshes the selected tab only	Refreshes all tabs in the file	Identifies data changes on the selected tab and stores these changes to the database	Identifies data changes on every tab and stores these changes to the database
Data Impacts	Clears all dirty cells on selected tab only	Clears all dirty cells on all tabs regardless of selected tab	Submits all data for the selected tab only	Submits all data for all tabs
Parameter Impacts (CV Only)	Prompts the user with any Parameters used on selected tab	Prompts the user with all the Parameters used in the workbook	No prompts	No prompts

### Numeric Precision

Decimals are automatically truncated after the ninth character in a cell or a function.



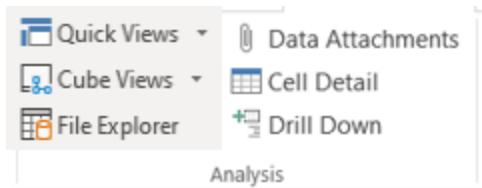
## Calculation



### Consolidate/Translate/Calculate

If permission is granted, these calculations can be performed on the selected cell.

# Analysis



## Quick Views

### Create a Quick View

This will create a new Quick View in the worksheet's selected cell.

### Create Copy of Selected Quick View

This will copy the selected Quick View in order to paste a version of it in another spreadsheet.

### Create Quick View Using POV from Selected Cell

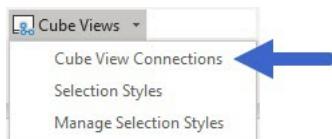
This will create a new Quick View based on the current POV from the selected cell. This can be done using a Quick View cell's POV or a Cube View cell's POV.

For more details on this feature, see *Quick View* in "About Excel Add-In" on page 1001.

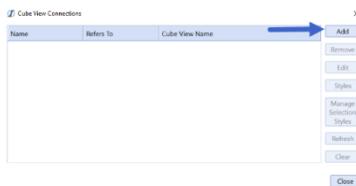
## Cube Views

Add a Cube View to an Excel sheet.

1. Click Cube Views > Cube View Connections.



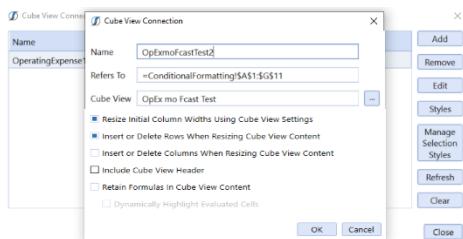
2. From this window, the cube views added to an Excel workbook can be managed. You can add, remove, edit, or go to styles. Click Add, to add a new Cube View.



## Navigating the Excel Add-In

---

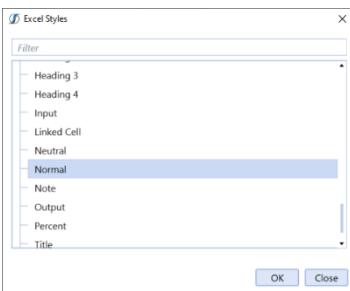
3. Name the connection and then choose the Cube View.



4. **Resize Initial Column Widths Using Cube View Settings** is the default setting. If you uncheck it, you can change the columns and save the Cube View. However, if you go back into the same Cube View Connection, the check box will be enabled and you'll need to uncheck it again to keep your new cube view settings.
5. Select whether there needs to be inserted or deleted rows and/or columns when resizing. This setting will move around other content in the sheets if the size of the Cube View changed since the last refresh.
6. You can select **Include Cube View Header** to add header rows to the spreadsheet.
7. **Retain Formulas in Cube View Content** allows you to enter formulas in the Cube View (Excel or Spreadsheet) and retain those formulas pre and post submission of the sheet or workbook. When the sheet or workbook is refreshed the formulas will remain. If the value resulting from the value is different than the value of the OneStream database, the cell will initially become a dirty cell and will turn the cell format to yellow.

**NOTE:** When using external Excel workbooks, or after any updates to referenced sheets within the same workbook, you must Refresh Sheet to visualize the dirty cells and then Submit Sheet, unless Dynamically Highlighted Evaluated Cells is turned on.

8. After the Cube View is added, it will appear on the sheet. If formatting was applied to the Cube View (see Cube Views in Presentation), the formatting will come forward into the Excel sheet. Otherwise, apply Excel Styles. These styles are stored in the Excel sheet and can be copied from workbook to workbook. For more information on Excel Styles, see "Styles" on page 1071.



**NOTE:** In order to copy Excel spreadsheet cells into a Data Explorer Grid on the web, click CTRL, select the cells desired, and then click CTRL-C. Navigate to the Data Explorer Grid, select a cell, and click CTRL-V, this will paste the cells into the grid. This can also be done from a Data Explorer Grid into an Excel Spreadsheet.

## Retain Formulas in Cube View Content in Excel and Spreadsheet

Retain Formulas in Cube View Content allows you to form Cube View grids of data in Excel, using the Cube Views menu function, that can be linked to other Excel models for easy submission into OneStream. This feature allows formulas (to writeable cells) in Excel (or Spreadsheet) for an attached Cube View to be retained on submission and retrieval instead of being replaced with the value of the represented formula.

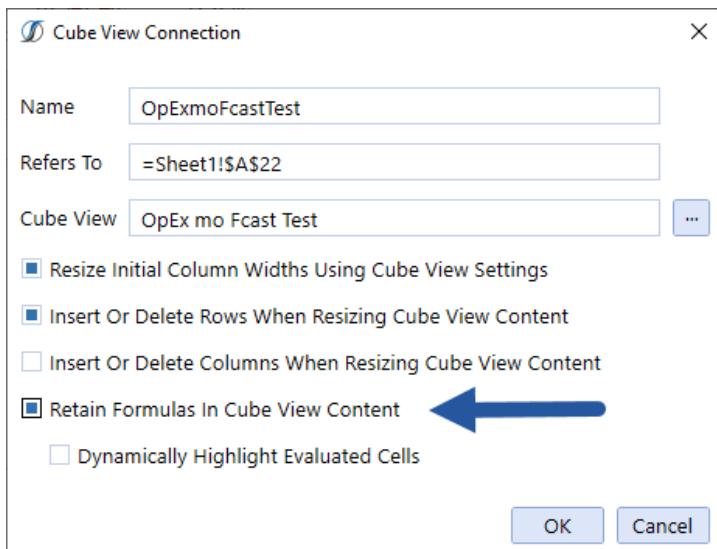
The Retain Formulas in Cube View Content feature, allows users to plan, budget or forecast and use the familiar functionality of Excel while still submitting data back to the OneStream database.

Use the Retain Formulas in Cube View Content feature to enter formulas in the Cube View (Excel or Spreadsheet) and retain those formulas pre and post submission of the sheet or workbook. When the sheet or workbook is refreshed, the formulas will remain. If the value resulting from the formula differs from the existing value in the OneStream database, the cell will initially become a dirty cell and will turn the cell format to yellow.

**NOTE:** When using external Excel workbooks, or after any updates to referenced sheets within the same workbook, you must Refresh Sheet to visualize the dirty cells and then Submit Sheet, unless you've turned on Dynamically Highlight Evaluated Cells in the cube view.

Retain Formulas in Cube View Content links to other Excel worksheets or worksheets in other Excel workbooks.

1. From the **OneStream** menu, select Cube Views > Cube View Connections.
2. Click **Add** in the Cube View Connection window or click **Edit** if you already have a cube view.
3. Click **Retain Formulas in Cube View Content** box and click **OK**.



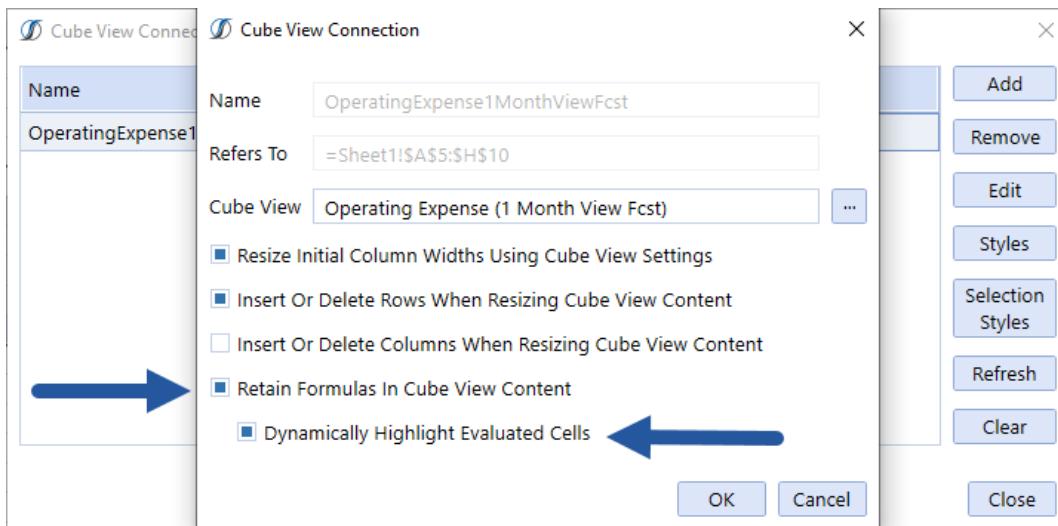
4. Add the Cube View, if one is not already selected, and click Close.

## Dynamically Highlight Evaluated Cells in Excel or Spreadsheet

When **Retain Formulas in Cube View Content** is enabled, the option to **Dynamically Highlight Evaluated Cells** becomes available to enable. When it's enabled, every time you make a change to a cell in Excel or Spreadsheet that is referenced in a Cube View, the cell will immediately update and show the update with a change in color. This cell update is called a dirty cell, which indicates that the cell value is different from the information in the OneStream database.

## Navigating the Excel Add-In

---



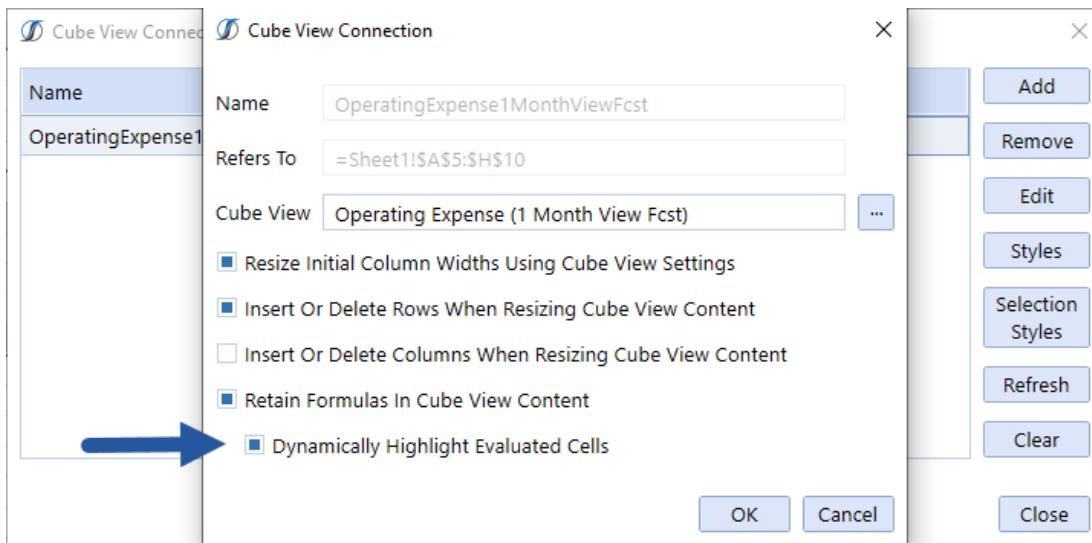
Dynamically Highlight Evaluated Cells saves you a step because the cell changes without requiring a refresh. This feature identifies the values in the cube view that have changed relative to its original value in the database. Evaluating all the cells in the spreadsheet.

Excel users who want to continue working in Excel to access can log in through the OneStream menu, update the cube view content and submit it to the database without leaving Excel. You can also perform these tasks in Spreadsheet within the application.

You can **Retain Formulas in a Cube View Content** that are related to values within a function, within an existing workbook, within a sheet, within other sheets, in external workbooks, and in external renamed worksheets in Excel. Spreadsheet also offers this functionality, but it doesn't allow you to point the cell references to external workbooks.

Click **Refresh Sheet** to see all changes within the cube view content and then click **Submit Sheet** or activate **Dynamically Highlight Evaluated Cells** and the cell updates automatically.

## Navigating the Excel Add-In



When a value for a formula in the cube view is changed by a cell reference, or a function related to a different cell is modified, if the value is different than what is in the database, a dirty cell is created. This means the value of the cell is different than the value of what is in the database and the cell will change colors.

The screenshot shows the OneStream XF ribbon with tabs like File, Home, Insert, Page Layout, Formulas, Data, Review, View, and OneStream XF. The OneStream XF tab is active. Below the ribbon is a spreadsheet with data in columns A-K. A blue arrow points to a cell in column E (row 11) containing '17,781.35'. Another blue arrow points to a cell in column E (row 12) containing '2,100'. The background of these cells is highlighted in yellow, indicating they contain formulas.

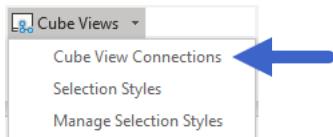
	A	B	C	D	E	F	G	H	I	J	K
1				Transportation	800						
2				Lodging	700						
3											
4				Baseline	Revisions	Full Forecast	PY Actual	Variance	Var %		
5	52010 - Exhibitions			117,813.20	150	117,963.20	88,359.90	-29,603.30	-25.1 %		
6	52000 - Promotions			166,647.05	200	166,847.05	124,985.29	-41,861.76	-25.1 %		
7	52020 - Consulting					75,000	75,000.00		-75,000.00	-100.0 %	
8	52030 - Advertising			180,359.92	13,500	193,859.92	135,269.94	-58,589.98	-30.2 %		
9	52000 - Promotions			166,647.05	200	166,847.05	124,985.29	-41,861.76	-25.1 %		
10	52199 - Travel & Entertainment			15,681.35	2,100	17,781.35	11,761.01	-6,020.34	-33.9 %		
11	52100 - Transportation			8,501.78	800	9,001.78	6,376.34	-2,625.45	-29.2 %		
12	52110 - Lodging			1,812.78	700	2,112.77	1,359.58	-753.19	-35.6 %		
13	52120 - Meals			3,503.74	500	4,003.74	2,627.81	-1,375.94	-34.4 %		
14	52130 - Entertainment			1,863.06	800	2,663.06	1,397.30	-1,265.77	-47.5 %		
15	52200 - Rent					2,800	2,800.00		-2,800.00	-100.0 %	
16											

The number of cells with formulas in the cube view determines the amount of time it takes to update the cells. You can turn the feature on or off and only use **Refresh Sheet** to update the values in the cells. Changes will show very quickly, no matter the size of the worksheet, when using Spreadsheet.

# Using Retain Formulas and Dynamically Highlight Evaluated Cells

You can use retain formulas and dynamically highlighted evaluated cells within a cube view to automatically display updated values in an existing workbook, a sheet or sheets, external workbooks, and external renamed worksheets in Excel. You can also do this in Spreadsheet within the OneStream application, however, you can't point the cell references to external workbooks.

1. In Excel, go to the OneStream menu and Log on.
2. Click **Cube Views > Cube View Connections**.



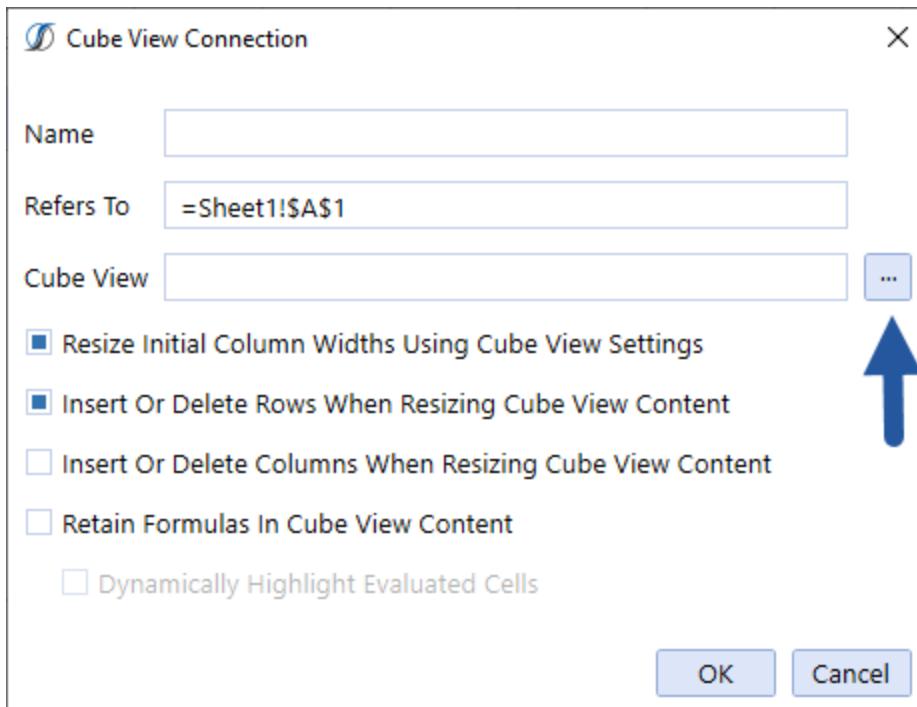
3. Click **Add**.



## Navigating the Excel Add-In

---

4. In the **Cube View Connection** window, click **Cube View**.

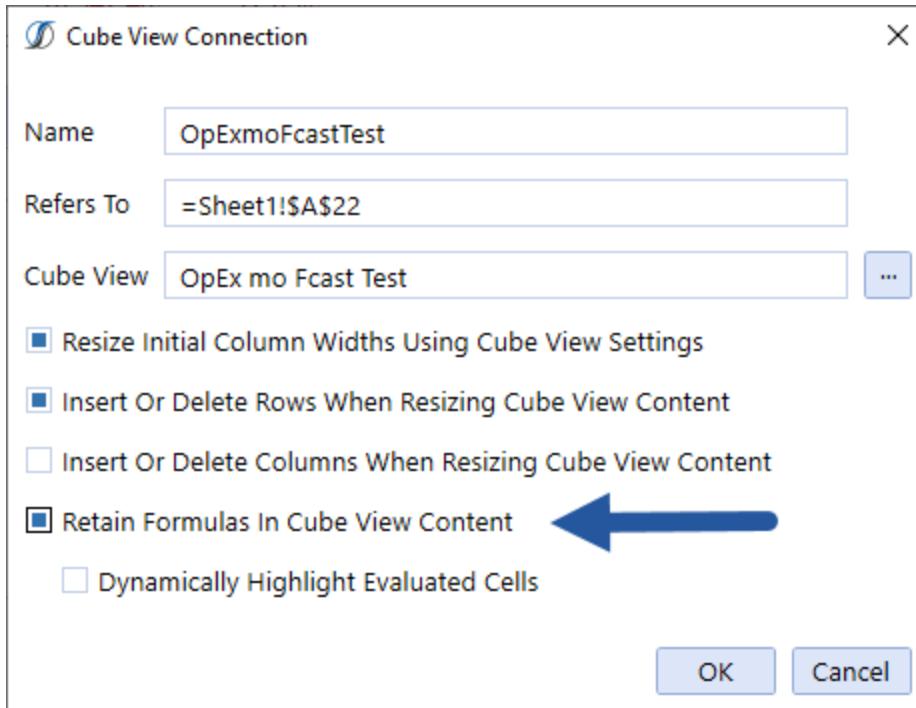


5. Select your choice and click **OK**.

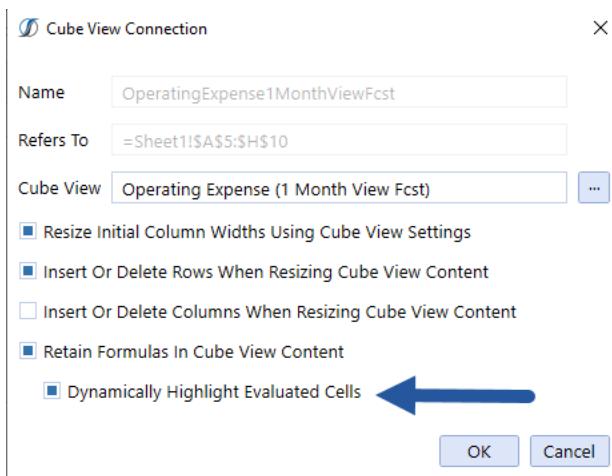
## Navigating the Excel Add-In

---

6. Click **Retain Formulas in Cube View Content** to activate **Dynamically Highlight Evaluated Cells**.



7. Then click **Dynamically Highlight Evaluated Cells** so you can see the changes as they are made.



## Navigating the Excel Add-In

8. Even if you don't activate the dynamically highlight evaluated cells feature, you can click **Refresh Sheet** after you make changes to see them.
9. If you're prompted, click **OK** once you've selected the parameters for the cube view.
10. Once the cube view has been added, you can click **Edit** to review, if needed.
11. Make changes to the sheet and press <Enter> to see the updated cell, which will change from white to yellow.

The screenshot shows the OneStream XF Excel Add-In interface. On the left is a standard Excel ribbon with tabs like File, Home, Insert, Page Layout, Formulas, Data, Review, View, and OneStream XF. The main area is a spreadsheet with columns A through G. Row 1 contains 'Holiday', 'Sick', and 'Overtime' with values 5000, 3500, and 800 respectively. Row 2 contains 'Baseline', 'Revisions', 'Full Forecast', 'PY Actual', 'Var \$', and 'Var %'. Rows 3 through 23 list various compensation categories with their respective values. To the right of the spreadsheet is a 'OneStream XF' pane titled 'POV' which displays a tree structure of perspectives: Global POV, Workflow POV, and Cube POV, with specific items like '2018M3' and 'Houston' selected. The bottom of the screen shows the standard Windows taskbar.

12. Click **Submit Sheet** to automatically save changes to the database.

This screenshot is identical to the one above, showing the OneStream XF Excel Add-In interface. The spreadsheet contains the same data as before, and the 'OneStream XF' pane on the right shows the same perspective hierarchy. The bottom of the screen shows the standard Windows taskbar.

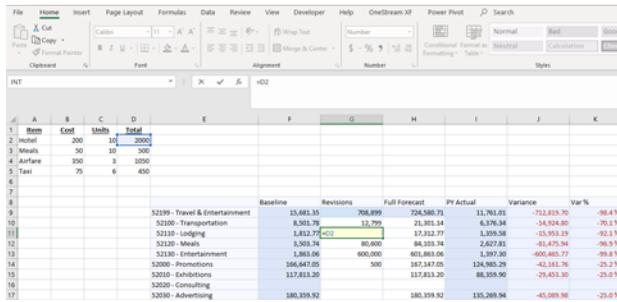
## Use Cases

These use cases are for both Excel and Spreadsheet unless otherwise noted.

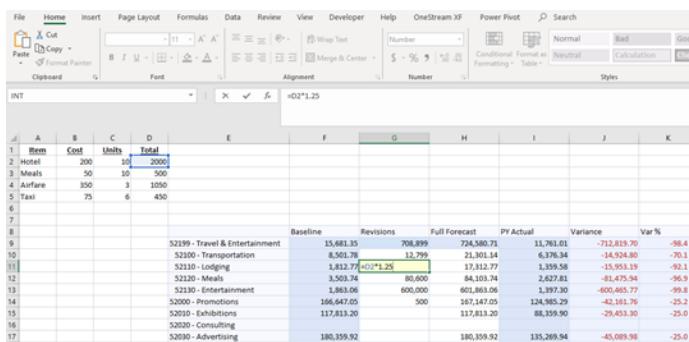
The placing of formulas or cell references. Retain Formulas can reference the following types of formulas. In all instances the formula will stay after refresh and/or submission.

Cell References of individual cells of data on the same sheet.

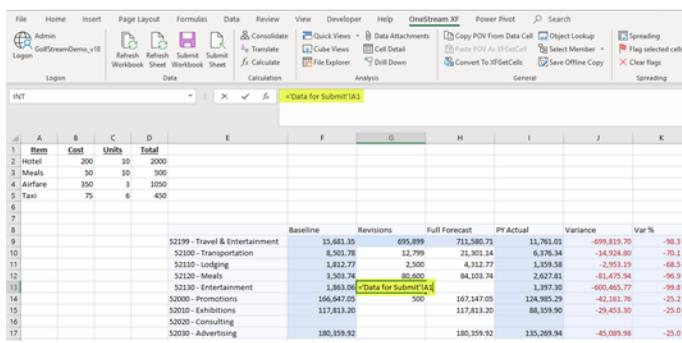
## **Navigating the Excel Add-In**



Cell References to a cell on the same sheet, factored by another value.



Cell References to cells on other sheets. These can also be factored by another value as well.



Referenced cell(s) on another saved workbook can also be factored by another value. (This applies to Excel only.)

## Navigating the Excel Add-In

---

Item	Cost	Units	Total
Hotel	200	10	2000
Meals	300	30	9000
Airfare	350	3	1050
Taxi	75	6	450

	Baseline	Revisions	Full Forecast	PY Actual	Variance	Var %
52199 - Travel & Entertainment	15,681.35	153,550	169,231.35	11,761.01	-157,470.34	-93.1 %
52100 - Transportation	8,501.78	(C:\ExternalData\Excel\Workbook.xlsx\Sheet2!\$D\$5)			-2,375.45	-28.8 %
52101 - Lodging	8,137.77	4,830.77	1,309.50	3,250.77	-2,949.23	-68.5 %
52102 - Meals	5,503.74	86,600	84,100.74	3,637.81	-81,473.94	-96.4 %
52130 - Entertainment	1,863.06	70,000	71,863.06	1,397.30	-70,465.77	-98.1 %
52000 - Promotions	166,647.05	500	167,147.05	124,985.29	-42,161.76	-25.2 %
52010 - Exhibitions	117,813.20		117,813.20	88,359.90	-29,453.30	-25.0 %
52020 - Consulting	180,359.92		180,359.92	135,269.94	-45,089.98	-25.0 %

## Best Practices

### Well-Formed Grid

It is suggested to create a “Well-Formed Grid” (Root.List or Comma Separated List) in Cube Views. When using this “Well-Formed Grid” (Root.List or Comma Separated List) in Cube Views, the Excel/SpreadSheet relative (=C2) and absolute formulas (=C\$2) will be retained.

However, when using these relative and absolute formulas within an Excel/Spreadsheet formula, users can use either the cell reference or text within the formula depending upon how members will be added or removed:

- =VLOOKUP(D30,Sheet1!A:B,2,FALSE) will work in a List or Comma-Separated list (Well-formed grid) when Accounts are added to the end.
- =VLOOKUP("52000 - Promotions",Sheet1!A:B,2,FALSE) will work in a case when a Member of a Row is moved up or down.

### Member Expansion Functions

When using Member Expansion Functions in Cube Views for Excel and SpreadSheet, the cell being referenced within the function (Vlookup, etc), will need to be adjusted and/or referenced as text.

- =VLOOKUP("52000 - Promotions",Sheet1!A:B,2,FALSE) will work in a Dynamic or when a Member of a Row is moved.
- =VLOOKUP(D30,Sheet1!A:B,2,FALSE) will NOT work in a Dynamic or when a Member of a Row is moved as this is using the cell ref of D30.

## **Other Notable Considerations**

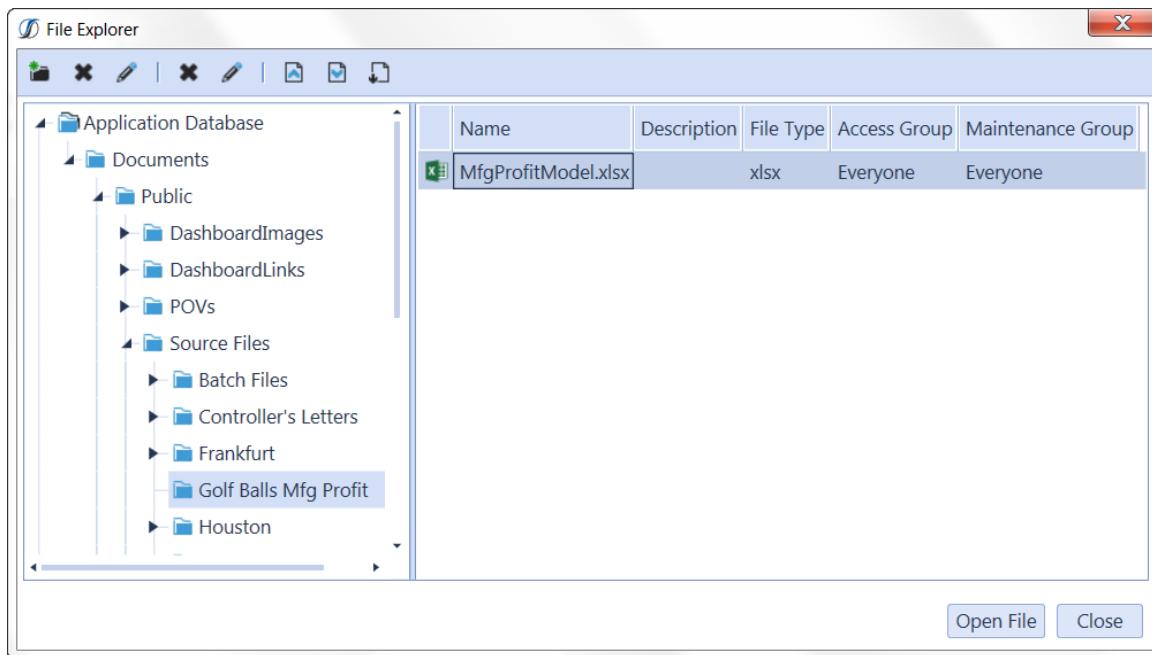
- Deselecting the Retain Formulas for Cube View Content will eliminate all formulas that were established /existed on the Cube View grid.
- Pivoting the existing Dimensions of the Cube View will break formulas.
- Changing the “structure” of the Cube View grid in the rows or columns will also break the formulas. For example; If you have Account, Entity, UD3 as the dimensions used in the row and switch it to UD3, Entity, Account, it will break the formulas.
- Users can change the POV to select a new dimension. This will change the Cube View results but retain the existing formulas that were established. The user at this point can choose to utilize the existing formulas, modify or delete. If the original formulas are modified or deleted, the last action will be saved.
- Linking to a white cell (writeable cell) to another cell in a different workbook will work ONLY in Excel and NOT in Spreadsheet.
- Prior to establishing links to an external workbook, the user should save the external workbook being referenced.
- When the user renames or saves as the (referenced) file, the user will need to update the links to the newly created file. Updating the links on the spreadsheet should be done BEFORE doing a refresh or submit.
- Formulas with cell references (VLOOKUP, INDEX(MATCH(, etc) that return errors (#N/A, #ERROR, etc) or non-numeric data will not retain the formula and return to its original value from the Cube View ; this error text cannot be converted into a number so the formulas will not retain.
- If a Dimension Member Name is renamed; i.e.; “52200 – Rent” is now “52200 – Rent Commercial”, the formula will break.

## **File Explorer**

Use this option to upload and download files.

## Navigating the Excel Add-In

---



### Create Folder

This creates a new folder under the selected folder on the left-hand side of the File Explorer pane.

### Delete Selected Folder/File

This deletes the selected folder on the left-hand side of the File Explorer pane or the selected file.

### Edit Selected Folder/Edit Selected File Information

This edits the Description, Maintenance Group, and Access Group for the selected folder or file.

### Upload File

This uploads the selected file and allows the user to save.

### Download Selected File

This downloads the selected file and allows the user to save.

### Download Selected File's Content File

This downloads the selected file's content file and allows the user to save.

### Data Attachments

This pulls up the Data Attachments dialog to show existing comments or attachments on a selected cell, or to allow data attachment edits.

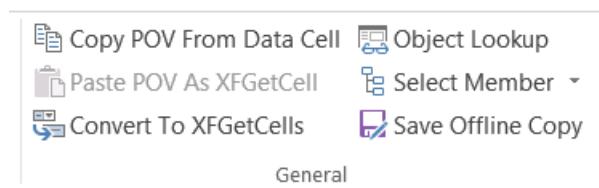
### Cell Detail

Enter Cell Detail for a Cube View or Quick View data cell. See Cell Detail in "Using OnePlace Cube Views" on page 962 for more details on this function.

### Drill Down

Drill down on a specific cell in order to see more details or gather more information. See Drill Down in "Using OnePlace Cube Views" on page 962 for more details on this function.

## General



### Copy POV from Data Cell

This captures the Point Of View of the currently selected cell. After clicking this option, the Paste POV As XFGGetCell becomes available and the Copy POV From Data Cell goes to grey. The ability to paste this into another cell is now available and OneStream will automatically convert this into an XFGGetCell formula with all of the appropriate Parameters.

### Paste POV As XFGGetCell

This option is only available after clicking Copy POV From Data Cell. After clicking this option, OneStream will convert the copied cell into an XFGGetCell formula. Click Refresh Data to retrieve the data.

### Convert to XFGGetCells

This will convert an existing Quick View into an XFGGetCells. After clicking this option, OneStream will prompt with the following: Are you sure you want to convert all of the data in Quick View 'Name of the Quick View' to XFGGetCells? By clicking OK, the Quick View definition will be deleted and converted to XFGGetCells.

### Object Lookup

Use the Object Lookup to insert objects from OneStream into Excel such as Foreign Exchange Rate Types when building formulas. If creating an Extensible Document in Excel, users can also use the Object Lookup to insert Parameters, Substitution Variables, or Image Content. See Object Lookup in "Presenting Data With Books, Cube Views and Dashboards" on page 547 for more details on this feature.

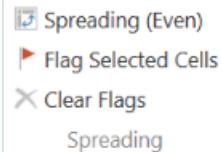
### Select Member

Select a Dimension Type from the drop-down list in order to view the Members of that Dimension. Select a Member of the hierarchy, and the Member name will display in the selected cell.

### Save Offline Copy

Use this to save an offline copy of the current worksheet without the functions. Users without the Excel Add-In can open this copy and see the saved values.

## Spreading



Allows users to see what type of spreading was used to spread data values over several columns or rows without having to type in each cell's values.

## Spreading Types

### Spreading (Even)

This distributed the active cell amount evenly across all selected cells.

### Spreading (445)

This distributed the active cell amount using a weighted 445 pattern across all selected cells.

### Spreading (454)

This distributed the active cell amount using a weighted 454 pattern across all selected cells.

### Spreading (544)

This distributed the active cell amount using a weighted 544 pattern across all selected cells.

### Spreading (Factor)

Multiplied all cells in the data range by the specified rate.

### Spreading (Fill)

Filled all cells in the data range with a specified value.

### Spreading (Proportional)

Distributed a value in all cells in the data range by the proportional based on the number of cells in the range.

### Spreading (Accumulate)

This starts with the active cell amount and cumulatively multiplies it by the specified rate.

### Spreading (Clear)

Cleared all data that was previously entered in the data range.

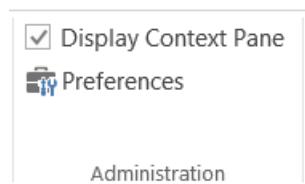
### Flag Selected Cells

Flags selected cells so the original amount in the cell is retained during the spreading process.

### Clear Flags

Select this to clear any flagged cells.

# Administration



## Display Context Pane

In order to display the OneStream task pane on the right-hand side of the screen, check this box. To hide the task pane, uncheck the box.

## Preferences

### General

#### Enable Microsoft Sign In

Set this to True if Azure is used for authentication to sign into the Excel Add-In. Setting this property to true will enable the Microsoft Sign In button on the login dialog allowing users to enter their Azure credentials. Set this to False, to disable the Microsoft Sign In button and users will be prompted to enter their username and password.

#### Enable Macros for Event Processing

If set to True, this enables Excel macros for OneStream API calls. The default is False.

#### Invalidate Old Data When Workbook is Opened

If set to True, this will force a data refresh on the opened workbook. The default is False.

#### Use Minimal Calculation for Refresh Sheet

This is for Excel Add-In only, not the Spreadsheet feature in OneStream Windows App. The default is True this will only calculate formulas and Excel functions in the active sheet. Set to False to revert to a full calculation of all workbooks and all sheets.

**NOTE:** Performance is best when Excel is set to use Manual Calculation Mode.

### Disable Interactive User During Refresh

Accounts for a known Excel situation when running on certain touchscreen hardware. If the **Refresh Sheet** or **Refresh Workbook** is pressed but the cells containing functions do not complete their calculations when processed, change the **Disable Interactive User During Refresh** setting under **Preferences** to True.

**NOTE:** Setting this to True may result in incompatibility issues with other Excel Add-ins.

### Retain All Formatting when Saving Offline Copy

This is for Excel Add-In only, not Spreadsheet. The default is False to derive basic formatting and better performance. Set this to True to obtain all character by character formatting, this will force a data refresh on the opened workbook.

### Use Add-In Compatibility Filter

When True, only cell selection change events, such as keystrokes or mouse clicks, are allowed. Third-party add-ins and macros cannot change cells. When False, users, third-party add-ins, and macros can make cell changes.

### Quick View Double-Click Behavior

Default Expansion for Rows/Columns

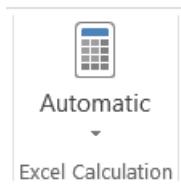
This determines what level of expansion displays when a user double-clicks a Quick View Row or Column Header. NextLevel is the default setting and allows multiple expansion paths when a user double-clicks a row or column header. There is also the ability to double-click an expanded item to collapse it again. This feature only works with the NextLevel setting.

For the following properties, See Quick View in "About Excel Add-In" on page 1001 In

Default Display Settings for New Quick Views

Default Suppression Settings for New Quick Views

## Excel Calculation



The Excel Calculation icon has the option of Automatic, Automatic Except for Data Tables, and Manual. It is recommended that the Calculation be set to Manual when using OneStream spreadsheets because the Automatic setting results in an Excel re-calculation every time a OneStream's interactive workbook changes data (e.g., when navigating a Quick View). However, this is not forced because a user might prefer Excel's Automatic calculation, especially when there is not a significant amount of OneStream data in the workbook.

# Right-Click Options

When working with a Cube View in Excel, the following right-click options are available:

### **Quick View**

See Quick View in "Navigating the Excel Add-In" on page 1034.

### **Expand**

Select the cell of a Member and choose how to view its data.

### **AllTops**

This returns the Top of the given Dimension.

### **AllBase**

This returns all Base Members of the given Dimension regardless of what Member is selected.

### **All**

This returns all Members in a given Dimension.

### **NextLevel**

This returns the next level of Members under the selected Member.

### **KeepOnly**

This will only keep the selected Members.

### **Parents**

This returns the direct Parents of the selected Member regardless of how many hierarchies to which the Member belongs.

### **Ancestors**

This returns all Members up the chain from the selected Member.

### **Children**

This returns the first level of Children under the selected Member.

### **ChildrenInclusive**

This returns the selected Member and its first level of Children.

### **Descendants**

This returns every Member under the selected Member in a list, not a hierarchy.

### **DescendantsInclusive**

This returns the selected Member and every Member under it in a list, not a hierarchy.

### **TreeDescendants**

This returns every Member under the selected Member in a hierarchy.

### **TreeDescendantsInclusive**

This returns the selected Member and every Member under it in a hierarchy.

### **Base**

This returns the Base level for the selected Member.

### **Paste POV**

This allows a user to Paste a POV into a selected cell in order to change the data within that Quick View.

### **Apply POV from Selected Cell**

This allows a POV to be passed from a selected cell within a Quick View.

### **Apply User POV**

This allows a Point of View to be passed between Quick Views.

### **Clear POV**

This will clear the POV for the selected Quick View.

For the following properties, refer to Quick View in "Navigating the Excel Add-In" on page 1034

### **Undo**

### **Redo**

### **Options**

### **Refresh**

### **Calculate / Translate / Consolidate**

Similar to the icons in the Ribbon, but here there is also the choice to do Force operations and additional Logging.

### **Select Member**

Select a Dimension Type from the drop down list in order to view the Members of that Dimension. Select a Member of the hierarchy, and the Member name will display in the selected cell.

### **Copy POV from Data Cell**

See General

### Paste POV As XFGetCell

See General

### Convert to XFGetCells

See General

### Cell Detail

See Analysis

### Data Attachments

See Analysis

### Cell POV Information

This gives a detailed summary of the selected Members related to this intersection as well as the full Member Script used to get this value. All the major properties of these Members can be seen from this dialog.

### Cell Status

This returns a long list of properties about a given cell.

### Drill Down

See Analysis

### Spreading

This allows users to enter data into an aggregate Member, like an annual time period, and spread values over several columns or rows without having to type in each cell's values.

## Spreading Type

### Fill

This fills each selected data cell with the value in the Amount to Spread property.

### Clear Data

This clears all data within the selected cells.

### Even Distribution

This takes the Amount to Spread and distributes it evenly across the selected cells.

### 445 Distribution

This takes the Amount to Spread and distributes it with a weight of 4 to the first two selected cells and a weight of 5 to the third cell.

### 454 Distribution

This takes the Amount to Spread and distributes it with a weight of 4 to the first selected cell, a weight of 5 to the second cell and a weight of 4 to the third.

### 544 Distribution

This takes the Amount to Spread and distributes it with a weight of 5 to the first selected cell and a weight of 4 to the second and third cells.

#### Factor

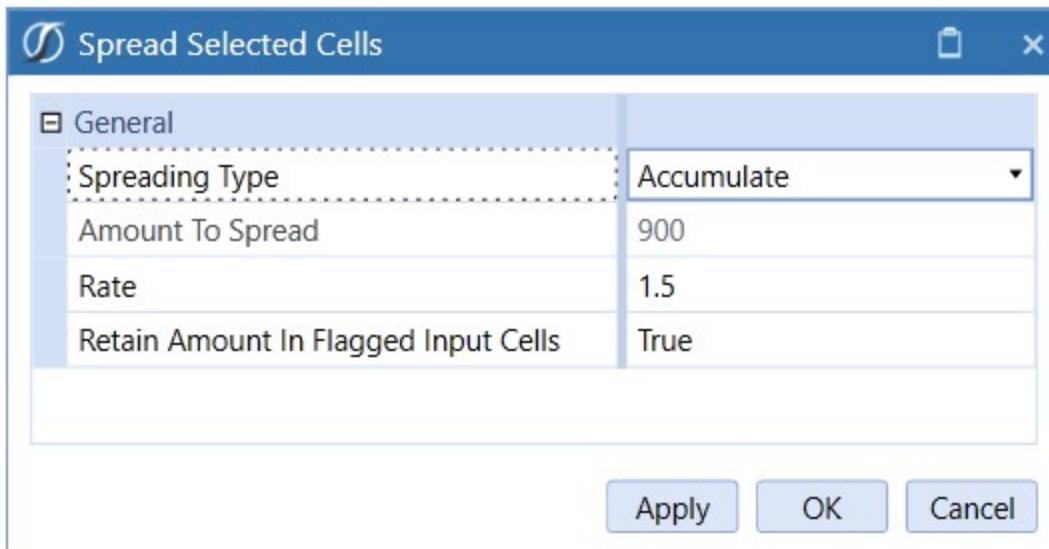
Multiply all cells by the specified rate.

#### Accumulate

This takes the first selected cell's value and multiplies it by the rate specified. It then takes that value, multiplies it by the specified rate and places it in the second cell selected, and does this for all selected cells. For example, four cells are selected and the first cell has a value of 900.

Apr 2011	May 2011	Jun 2011	Jul 2011
909.00	14.00	622.50	33.75
900.00	0.00	0.00	0.00

The Accumulate Spreading is setup as follows with a rate of 1.5:



When the spreading is applied the outcome is as follows:

## Navigating the Excel Add-In

---

Apr 2011	May 2011	Jun 2011	Jul 2011
1,359.00	2,039.00	3,660.00	4,590.00
1,350.00	2,025.00	3,037.50	4,556.25

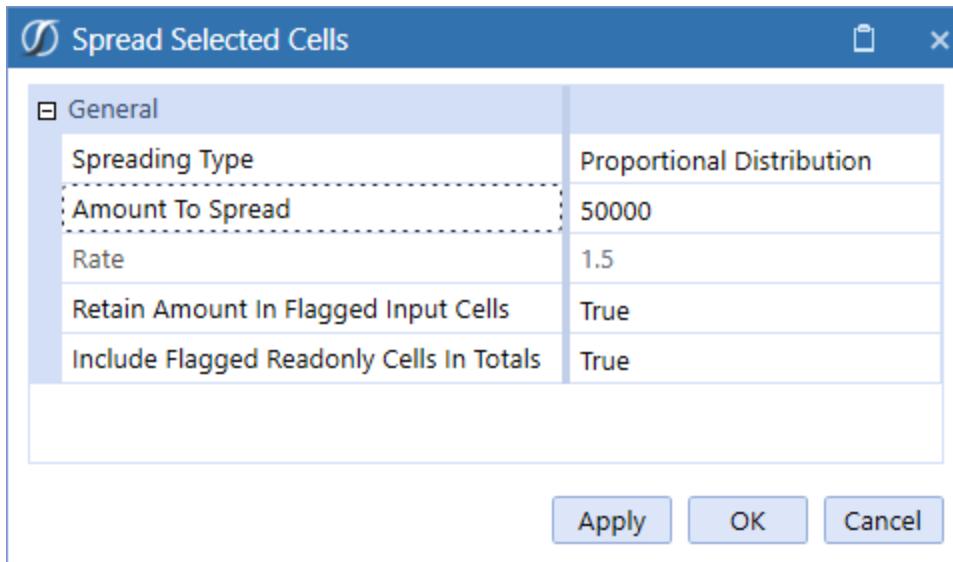
Each cell's value is a factor of the previous cell amount.

### Proportional Distribution

This takes the selected cell's value, multiplies it by the specified Amount to Spread, and then divides it by the total sum of all selected cells. If all the cells have a zero value, the Amount to Spread will behave like an Even Distribution.

Apr 2011	May 2011	Jun 2011	Jul 2011
1,359.00	2,039.00	3,660.00	4,590.00
1,350.00	2,025.00	3,037.50	4,556.25

A proportional amount of 50,000 is applied to the cells:



Result:

Apr 2011	May 2011	Jun 2011	Jul 2011
6,162.85	9,244.77	14,468.65	20,802.98
6,153.85	9,230.77	13,846.15	20,769.23

## Spreading Properties

### Amount to Spread

Specify the value to spread over the selected cells. The value defaults to the last cell selected. The way the amount in this field spreads varies by Spreading Type.

Rate (Factor and Accumulate Spreading Types Only) Enter a rate to multiply by a cell value.

### Retain Amount in Flagged Input Cells

Users can flag specific cells in order to retain the data within the cell. If this property is set to True, spreading will not apply to the selected flagged cells.

### Include Flagged Readonly Cells in Totals

Set this to True to include locked base-level cell values when calculating spreading totals. True is the default.

# Retain Formatting in Cube Views with Selection Styles and Conditional Formatting

## Using Selection Styles

You can bring an existing cube view into Excel or Spreadsheet and format using functionality that is included in Excel, to create highly formatted reporting. You can also modify it locally as well as in spreadsheet. You can use existing cube view formatting, add new styles to apply changes to rows, columns, or cells, or a combination of existing styles with new styles, and add conditional formatting.

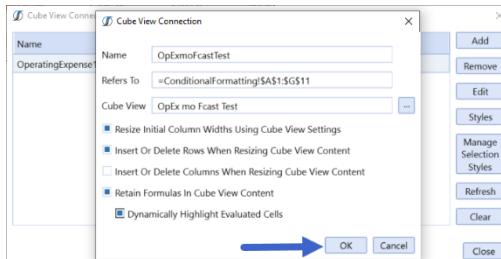
## Creating a Selection Style

1. Log in using the OneStream menu.
2. Click **Cube Views > Cube View Connections**.

## Navigating the Excel Add-In

---

3. Click **Add**.
4. Select the cube view, click **OK**, then **Close**.



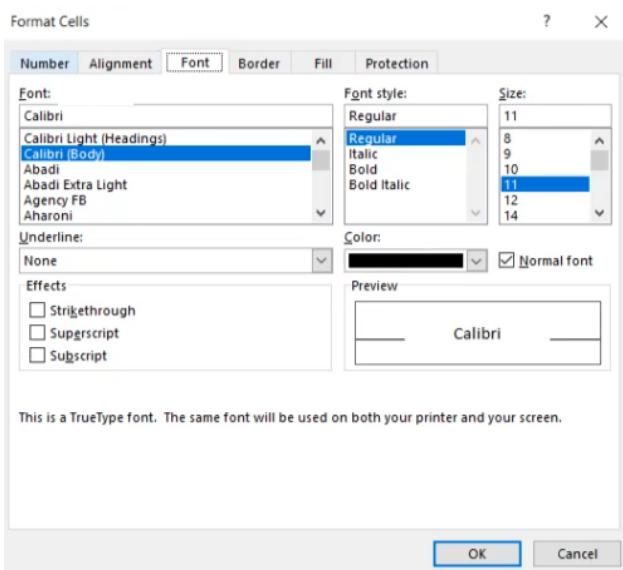
5. Select the cell, a group of cells, a column, or row to format.

	Revisions	Full Forecast	PY Actual
50200 - Base		367,608.79	275,706.59
50210 - Overtime		51,231.96	38,423.97
50220 - Bonus			
50229 - Commission Above		52,881.57	39,661.18
50230 - Commission		15,894.55	11,920.91
50231 - Commission Below		487,616.88	365,712.66
50240 - Holiday		93,385.46	70,039.10
50250 - Vacation		390.44	292.83
50260 - Sick		5,789.90	4,342.43
50299 - Total Employee Salary		10,422.28	7,816.71
50100 - Health Ins			
50110 - Disability Ins		109,988.09	82,491.07
50120 - Workers Compensation Ins		5,732.26	4,299.19
50130 - Retirement Plan		45,336.94	34,002.71
50140 - Other Fringe Benefits		39,865.24	29,898.93
50199 - Total Employee Ins & Benefits		90,934.44	68,200.83
50000 - FUTA		688,539.40	516,404.55
50010 - SUTA			
50020 - FICA			
50099 - Total Payroll Taxes		8	8.00
50300 - Total Employee Compensation			
51000 - Gas			

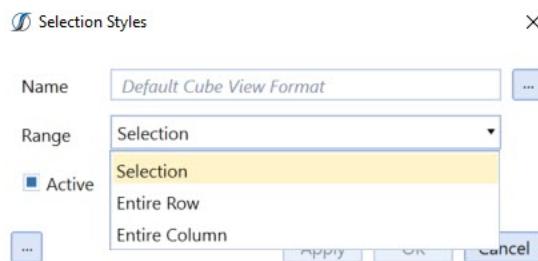
## Navigating the Excel Add-In

---

6. Either right-click to open Format Cells or use the Home menu to choose formatting.



7. After formatting, click **OneStream > Cube Views > Selection Styles**.
8. Enter a Name and Range to apply to the current Selection, Entire Row, or Entire Column. Then click **OK**.



## Navigating the Excel Add-In

---

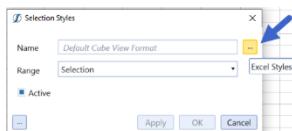
9. The formatting is applied to cube view.

	Revisions	Full Forecast
50200 - Base		\$ 367,608.79
50210 - Overtime		\$ 51,231.96
50220 - Bonus		
50229 - Commission Above		
50230 - Commission		
50231 - Commission Below		
50240 - Holiday		
50250 - Vacation		\$ 52,881.57
50260 - Sick		\$ 15,894.55
50299 - Total Employee Salary		\$ 487,616.88
50100 - Health Ins		93,385.46
50110 - Disability Ins		390.44
50120 - Workers Compensation Ins		5,789.90
50130 - Retirement Plan		10,422.28

10. Save the file to save the formatting to the cube view.

## Using an Existing Style

1. Select the cell, a group of cells, a column, or row to format, then click **Cube Views > Selection Styles**.
2. Click Excel Styles to select a style. If a style is not selected the Default Cube View Format is used.



## Navigating the Excel Add-In

3. On the style palette, hover to view the styles.

The screenshot shows a Microsoft Excel spreadsheet with data in columns A through J and rows 1 through 11. A black arrow points from the text in step 3 to the 'Selection Styles' palette window, which is overlaid on the spreadsheet. The palette has a title bar 'Selection Styles' and a close button 'X'. It includes fields for 'Name' (set to 'Default Cube View Format') and 'Range' (set to 'Selection'). A checkbox 'Active' is checked. At the bottom are 'Apply', 'OK', and 'Cancel' buttons. The main area of the palette displays various color-coded styles under sections like 'Custom', 'Data and Model', and 'Titles and Headings'. A second black arrow points from the text in step 4 to the 'GreenCurrency' style in the 'Custom' section of the palette.

4. Click the style to use and then Apply or OK.
5. Save the file to save the cube view formatting.

## Adding a Selection Styles Shortcut

In Excel, right-click Selection Styles then click Add to Quick Access Toolbar. This adds a shortcut to your toolbar that launches the Selection Styles window.

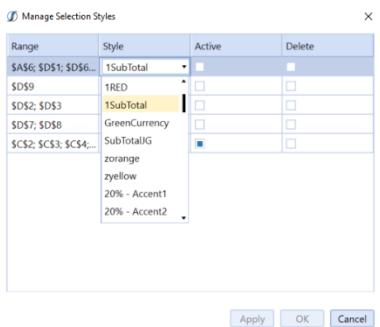


## Reviewing Styles and Ranges

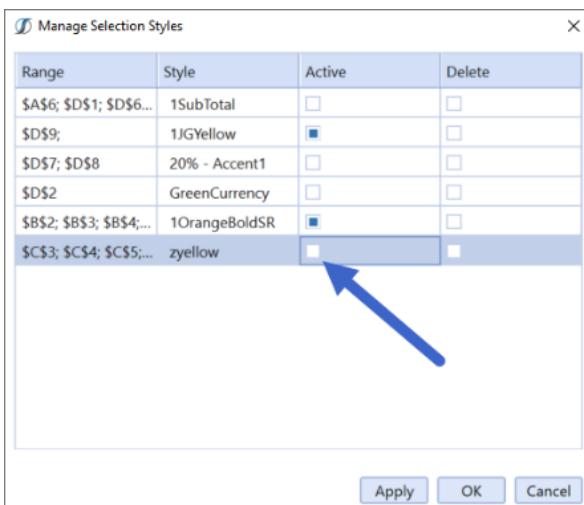
1. Click Cube Views > Manage Selection Styles.
2. Review your styles that are applied to the cube view.

## Navigating the Excel Add-In

### 3. Modify selected styles.

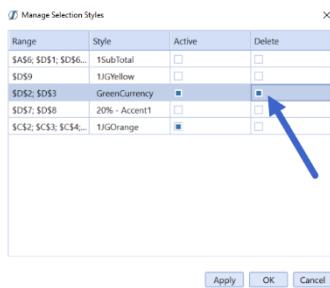


### 4. Enable or disable Active styles as needed and click OK.



### 5. Click Apply to preview the changes, or OK to apply the changes.

6. In Selection Styles, you will see style that are no longer Active. Click **Activate** to enable them.
7. You can delete a style from the cube view but it will be available in the current workbook.



8. Save the file to save the cube view formatting.

## Using Right-click Menu Options

You can manage selection styles by right-clicking in a cube view and selecting **OneStream > Cube Views > Cube View Connections > Selection Styles > Manage Selection Styles**.

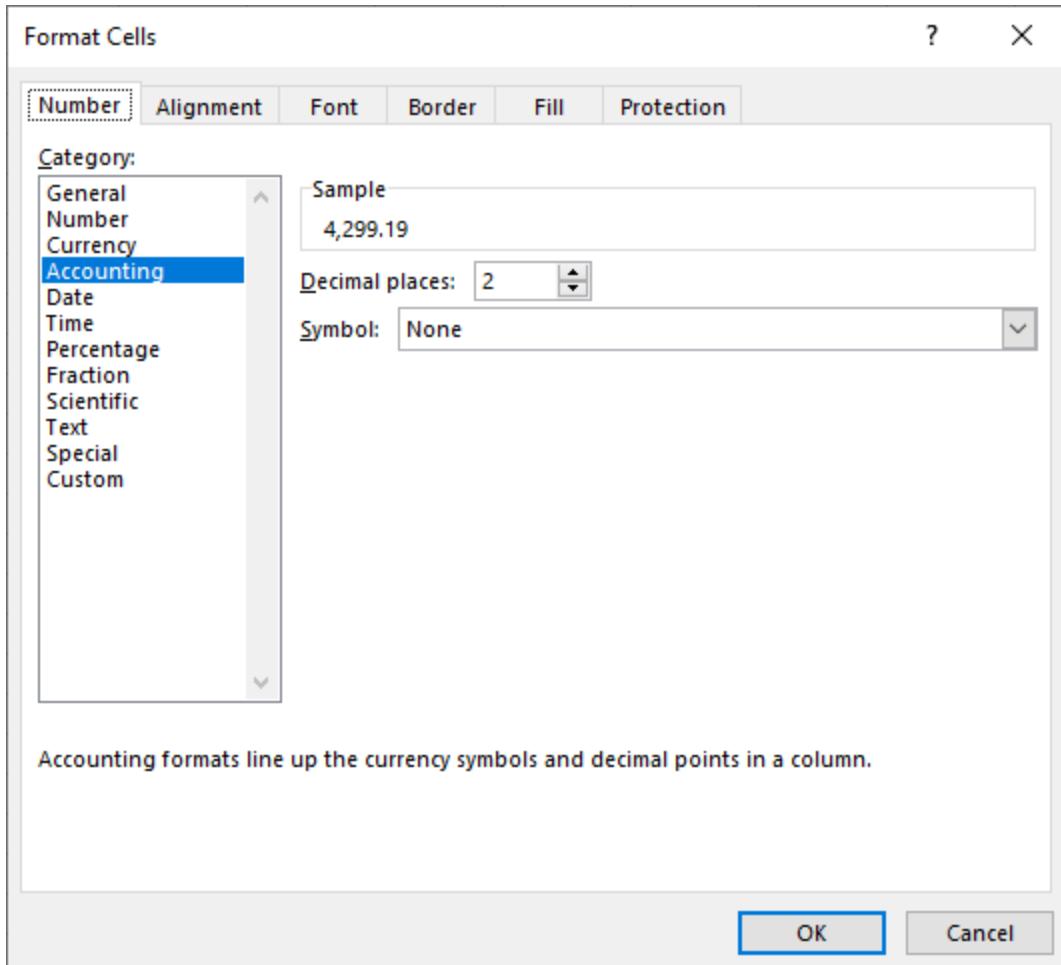
## Modifying and Duplicate Styles

1. In the Home menu, click **Styles** and right-click a style.
2. Click **Modify** or **Duplicate** and then **Format**.

## Navigating the Excel Add-In

---

3. Make any necessary changes and click **OK**.



4. Save the file.

## Merging Styles

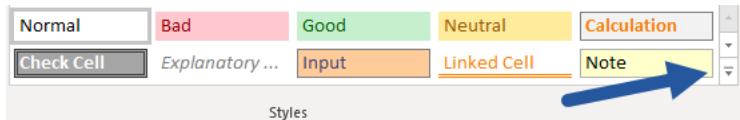
You can use styles created in other workbooks in Excel only.

1. Open a new workbook.
2. Add a cube view.

## Navigating the Excel Add-In

---

3. Click **Home > Styles** then the **More** arrow.



4. Click **Merge Styles**, select the style, and click **OK**.

The style is now available in the workbook.

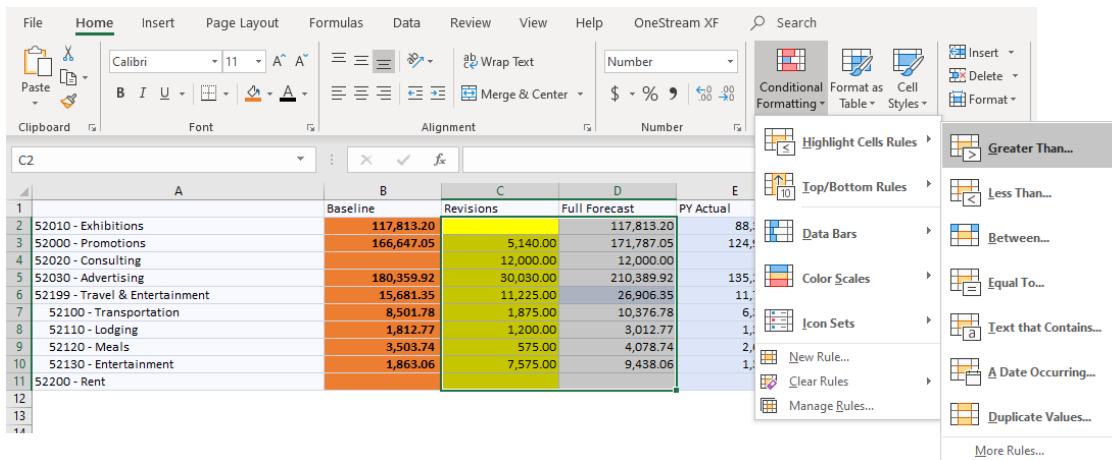


## Conditional Formatting

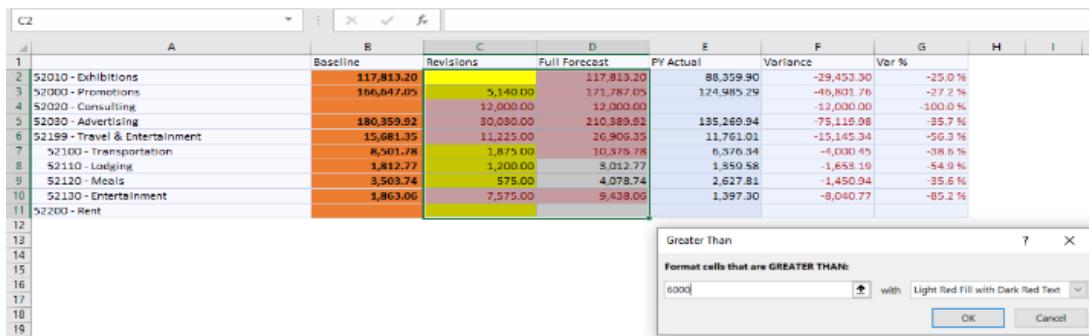
Use Conditional Formatting in Cube Views to visually explore and analyze data. You can highlight cells or ranges of cells, identify key values, and represent data using data bars, color scales, and icon sets that correspond to specific variations in the data. If there are any existing formats prior to applying Conditional Formatting, they will be retained if the range of cells containing the conditional formats do not meet the conditions of the rule. All styles from the cube view and the selection styles that had been previously applied to that range are overridden by conditional formatting.

## Create Conditional Formatting in an Existing or New Cube View

1. Click Home > Conditional Formatting > Highlight Cells Rules > Greater Than.



2. Enter 6000 and click OK.



3. Go to OneStream and click Refresh Sheet to see that your changes have been applied.

## Navigating the Excel Add-In

---

4. If you make a change that is different than the value in the database, the cell will change to pale yellow, until you refresh or submit.

A	B	C	D	E	F	G
1	Baseline	Revisions	Full Forecast	PY Actual	Variance	Var %
2 52010 - Exhibitions	117,813.20		5,000.00	88,359.90	-29,453.30	-25.0 %
3 52000 - Promotions	166,647.05	5,140.00	171,787.05	124,985.29	-46,801.76	-27.2 %
4 52020 - Consulting		12,000.00	12,000.00		-12,000.00	-100.0 %
5 52030 - Advertising	180,359.92	30,030.00	210,389.92	135,269.94	-75,119.98	-35.7 %
6 52199 - Travel & Entertainment	15,681.35	11,225.00	26,906.35	11,761.01	-15,145.34	-56.3 %
7 52100 - Transportation	8,501.78	1,875.00	10,376.78	6,376.34	-4,000.45	-38.6 %
8 52110 - Lodging	1,812.77	1,200.00	3,012.77	1,359.58	-1,653.19	-54.9 %
9 52120 - Meals	3,503.74	575.00	4,078.74	2,627.81	-1,450.94	-35.6 %
10 52130 - Entertainment	1,863.06	7,575.00	9,438.06	1,397.30	-8,040.77	-85.2 %
11 52200 - Rent						
12						

5. If you submit, it will revert to the formatting that was in the cube view since it is no longer greater than 6000.

A	B	C	D	E	F	G
1	Baseline	Revisions	Full Forecast	PY Actual	Variance	Var %
2 52010 - Exhibitions	117,813.20	-112,813.20	5,000.00	88,359.90	83,359.90	1667.2 %
3 52000 - Promotions	166,647.05	5,140.00	171,787.05	124,985.29	-46,801.76	-27.2 %
4 52020 - Consulting		12,000.00	12,000.00		-12,000.00	-100.0 %
5 52030 - Advertising	180,359.92	30,030.00	210,389.92	135,269.94	-75,119.98	-35.7 %
6 52199 - Travel & Entertainment	15,681.35	11,225.00	26,906.35	11,761.01	-15,145.34	-56.3 %
7 52100 - Transportation	8,501.78	1,875.00	10,376.78	6,376.34	-4,000.45	-38.6 %
8 52110 - Lodging	1,812.77	1,200.00	3,012.77	1,359.58	-1,653.19	-54.9 %
9 52120 - Meals	3,503.74	575.00	4,078.74	2,627.81	-1,450.94	-35.6 %
10 52130 - Entertainment	1,863.06	7,575.00	9,438.06	1,397.30	-8,040.77	-85.2 %
11 52200 - Rent						
12						

6. If you make a change to a cell that has conditional formatting and a selection style,

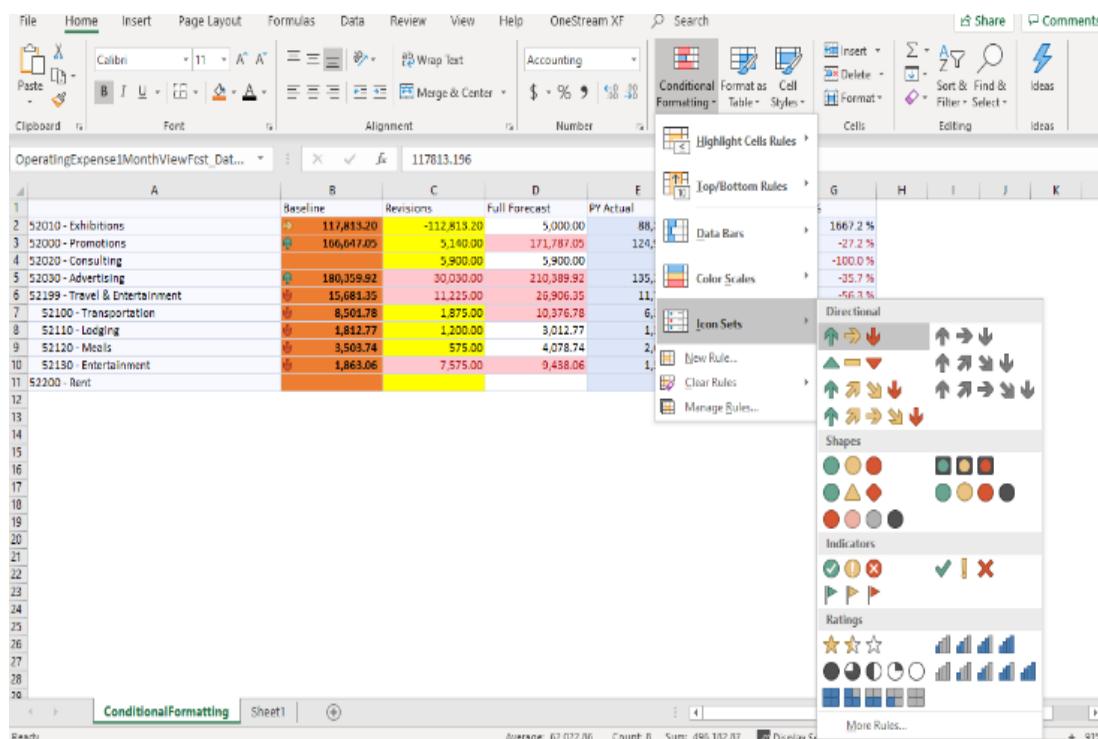
A	B	C	D	E	F	G
1	Baseline	Revisions	Full Forecast	PY Actual	Variance	Var %
2 52010 - Exhibitions	117,813.20	-112,813.20	5,000.00	88,359.90	83,359.90	1667.2 %
3 52000 - Promotions	166,647.05	5,140.00	171,787.05	124,985.29	-46,801.76	-27.2 %
4 52020 - Consulting		5,900.00	12,000.00		-12,000.00	-100.0 %
5 52030 - Advertising	180,359.92	30,030.00	210,389.92	135,269.94	-75,119.98	-35.7 %
6 52199 - Travel & Entertainment	15,681.35	11,225.00	26,906.35	11,761.01	-15,145.34	-56.3 %
7 52100 - Transportation	8,501.78	1,875.00	10,376.78	6,376.34	-4,000.45	-38.6 %
8 52110 - Lodging	1,812.77	1,200.00	3,012.77	1,359.58	-1,653.19	-54.9 %
9 52120 - Meals	3,503.74	575.00	4,078.74	2,627.81	-1,450.94	-35.6 %
10 52130 - Entertainment	1,863.06	7,575.00	9,438.06	1,397.30	-8,040.77	-85.2 %
11 52200 - Rent						
12						

## Navigating the Excel Add-In

when you submit, it will convert back to the selection style since it is no longer greater than 6000.

	A	B	C	D	E	F	G
1		Baseline	Revisions	Full Forecast	PY Actual	Variance	Var %
2	52010 - Exhibitions	117,813.20	-112,813.20	5,000.00	88,359.90	83,359.90	1667.2 %
3	52000 - Promotions	166,647.05	5,140.00	171,787.05	124,985.29	-46,801.76	-27.2 %
4	52020 - Consulting		5,900.00	5,900.00		-5,900.00	-100.0 %
5	52030 - Advertising	180,359.92	30,030.00	210,389.92	135,269.94	-75,119.98	-35.7 %
6	52199 - Travel & Entertainment	15,681.35	11,225.00	26,906.35	11,761.01	-15,145.34	-56.3 %
7	52100 - Transportation	8,501.78	1,875.00	10,376.78	6,376.34	-4,000.45	-38.6 %
8	52110 - Lodging	1,812.77	1,200.00	3,012.77	1,359.58	-1,653.19	-54.9 %
9	52120 - Meals	3,503.74	575.00	4,078.74	2,627.81	-1,450.94	-35.6 %
10	52130 - Entertainment	1,863.06	7,575.00	9,438.06	1,397.30	-8,040.77	-85.2 %
11	52200 - Rent						

7. To add icons, go Home > Conditional Formatting > Icon Sets and select the icons to use, in this example, select the arrows.



The screenshot shows the Microsoft Excel ribbon with the 'Home' tab selected. In the 'Conditional Formatting' section of the ribbon, the 'Icon Sets' option is highlighted. A dropdown menu is open, displaying the 'Icon Sets' category. Within this category, the 'Arrows' icon set is selected, showing a preview of various directional arrow icons (up, down, left, right, etc.) in different colors. The main Excel interface shows a table of operating expenses with conditional formatting applied to the 'Revisions' column.

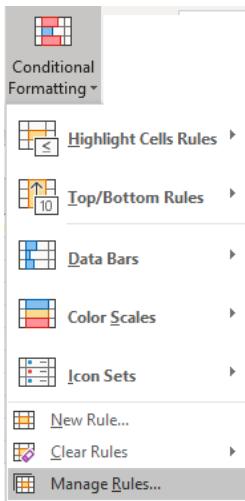
## Navigating the Excel Add-In

---

8. The icons are part of the cube view.

	A	B	C	D	E	F	G
1		Baseline	Revisions	Full Forecast	PY Actual	Variance	Var %
2	52010 - Exhibitions	→ 117,813.20	-112,813.20	5,000.00	88,359.90	83,359.90	1667.2 %
3	52000 - Promotions	↑ 166,647.05	5,140.00	171,787.05	124,985.29	-46,801.76	-27.2 %
4	52020 - Consulting	↑ 180,359.92	30,030.00	210,389.92	135,269.94	-75,119.98	-35.7 %
5	52199 - Travel & Entertainment	↑ 15,681.35	11,225.00	26,906.35	11,761.01	-15,145.34	-56.3 %
6	52100 - Transportation	↑ 8,501.78	1,875.00	10,376.78	6,376.34	-4,000.45	-38.6 %
7	52110 - Lodging	↑ 3,503.74	1,200.00	3,012.77	1,359.58	-1,653.19	-54.9 %
8	52120 - Meals	↑ 1,863.06	575.00	4,078.74	2,627.81	-1,450.94	-35.6 %
9	52130 - Entertainment		7,575.00	9,438.06	1,397.30	-8,040.77	-85.2 %
10	52200 - Rent						
11							
12							

9. You can also create, edit, delete, and view all conditional formatting rules.



10. If you save the workbook, the conditional formatting is saved.

## Styles

The same standard Styles are used in Excel, however, if you want to create a new style in order to change the format of how the numbers are displayed, see the example that follows.

### Creating a Custom Style

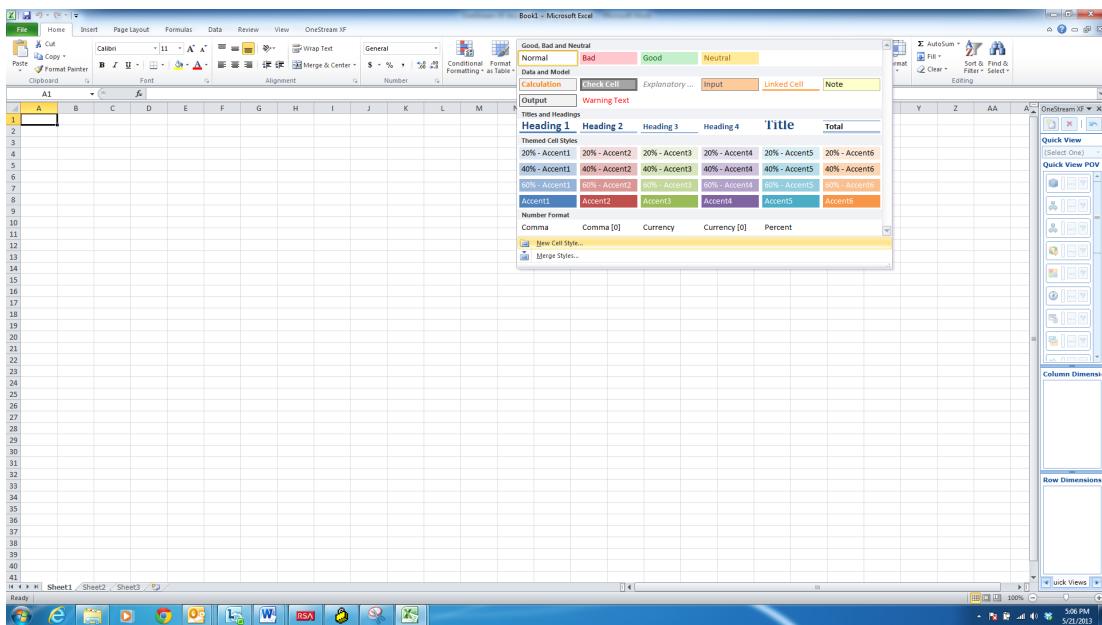
The following example was created in the Excel 2010 Version.  
On the Home Tab, click on the new style sheet.

## Navigating the Excel Add-In

---



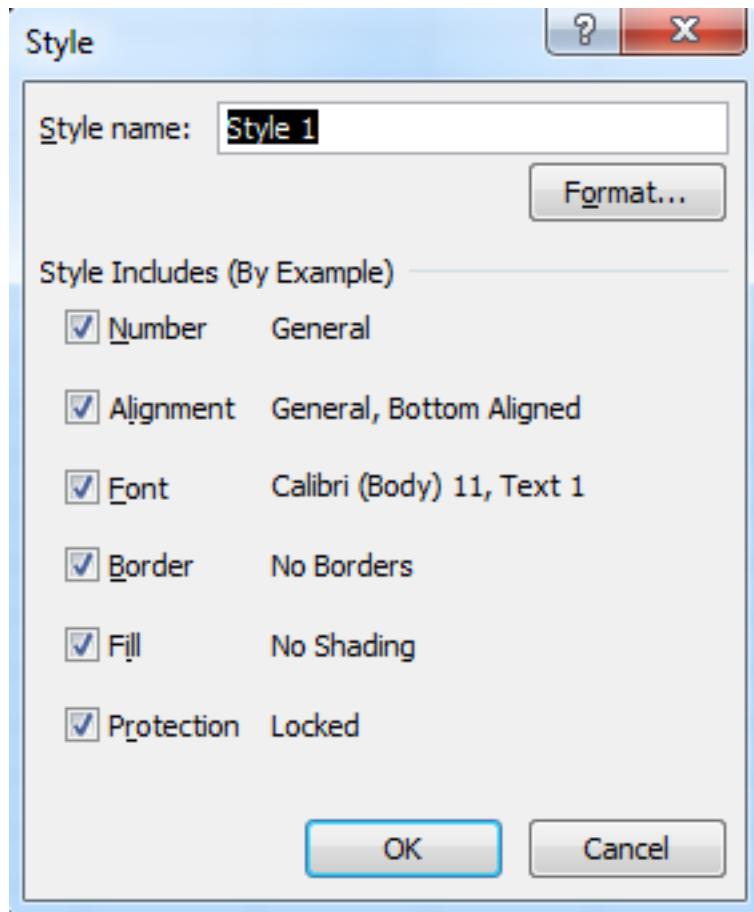
### Select New Cell Style



A new style window will appear, click Format.

## Navigating the Excel Add-In

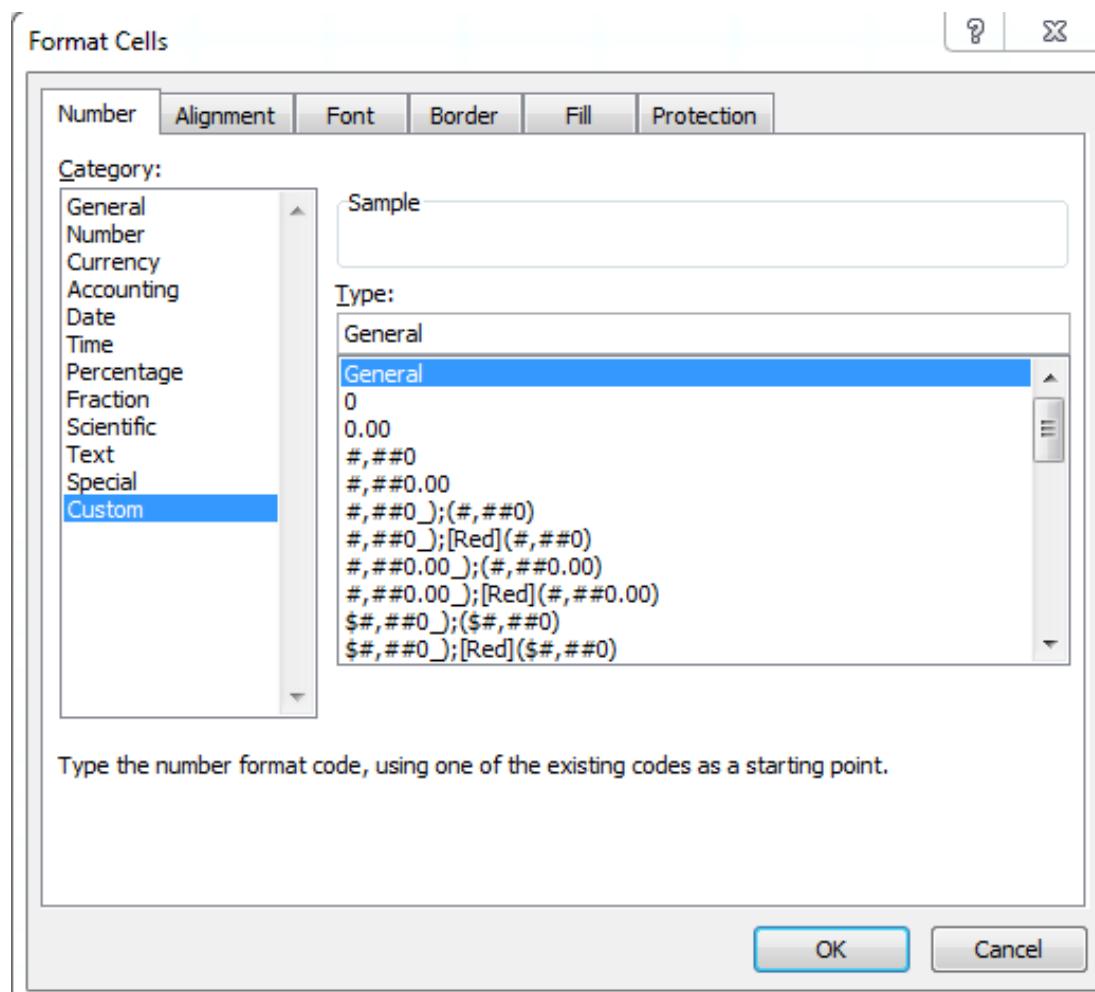
---



The format page will then display, click Custom.

## Navigating the Excel Add-In

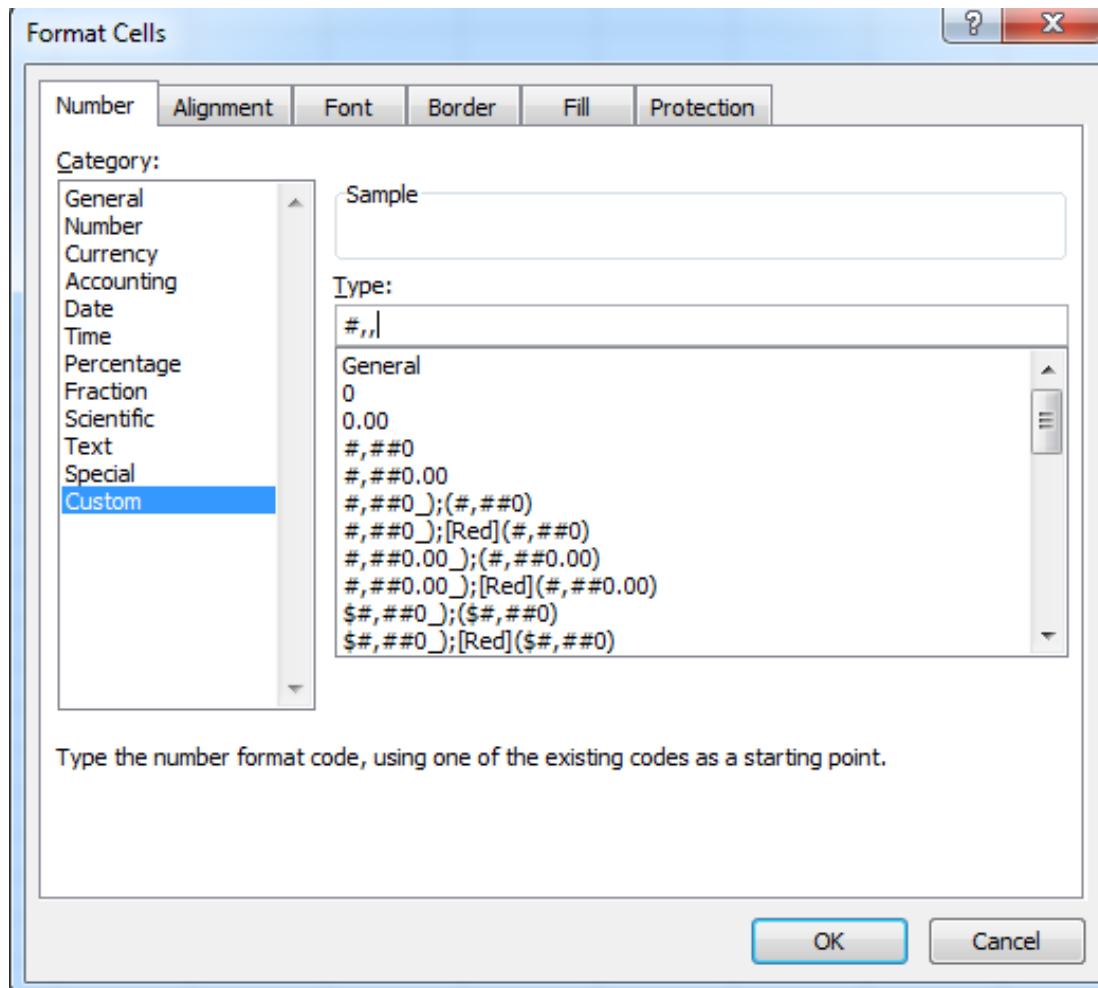
---



Finally, under Type, enter the custom formatting. This example will be formatting for Millions. (#,,)

## Navigating the Excel Add-In

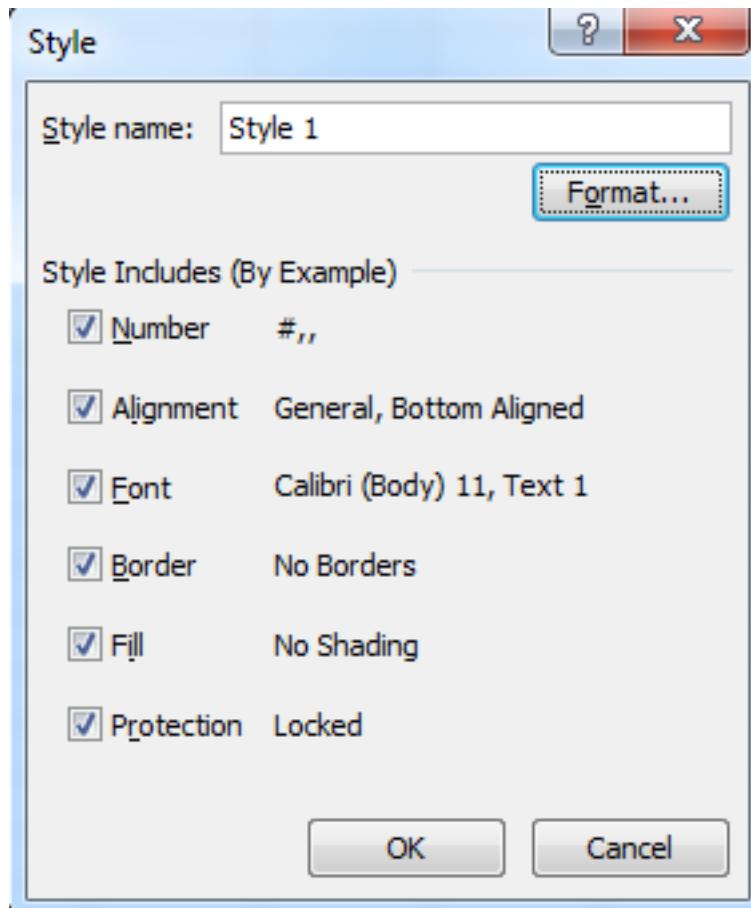
---



Click OK.

## Navigating the Excel Add-In

---



Type in a Style Name and click OK.

Now that a new style sheet has been created in Excel, it can be assigned to a Quick view.

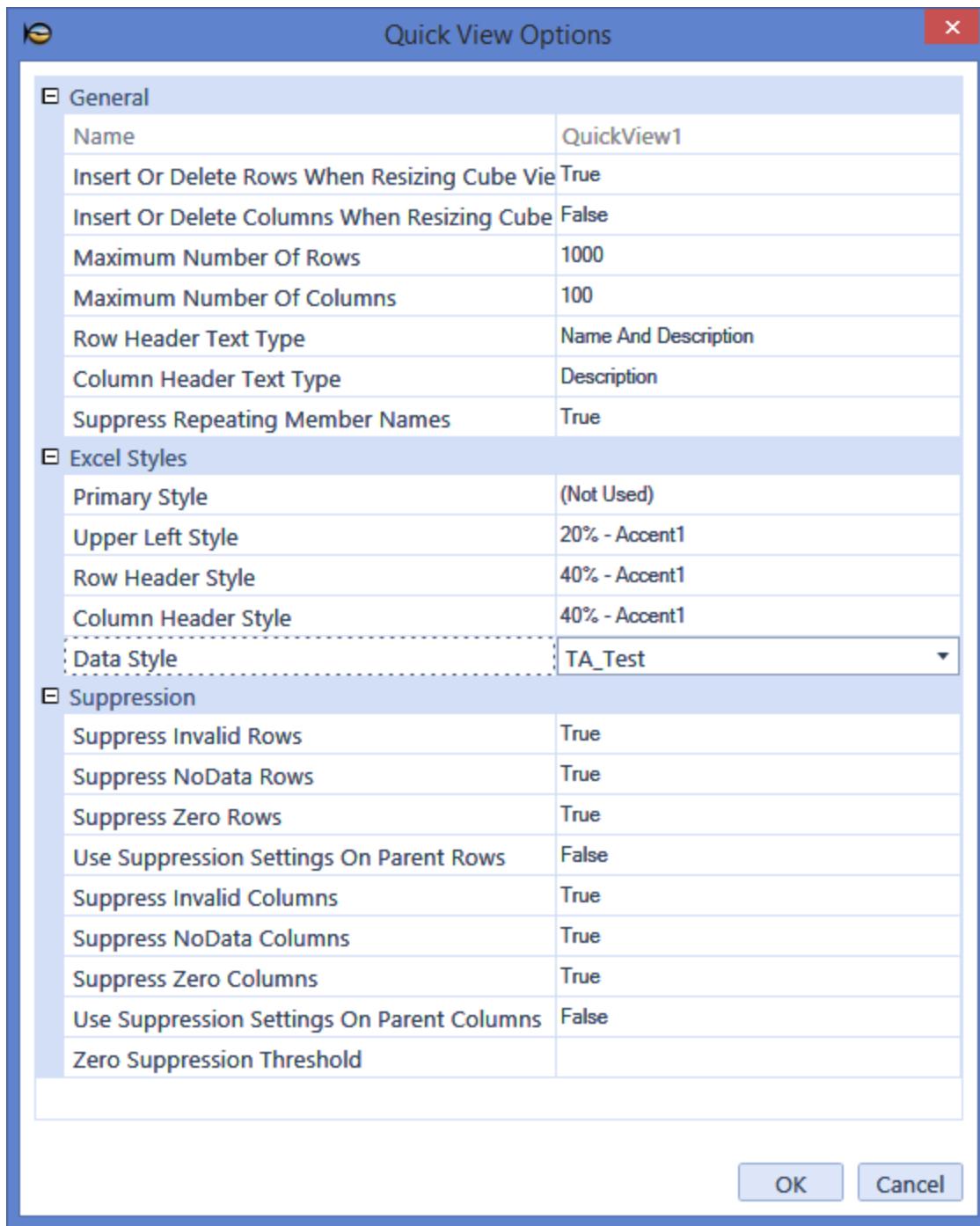
This is the Quick View before the formatting:

2013M2 - Feb 2013	
MXPISTN - Federal-Mogul Puebla Pistons	
MXN - Mexico, Pesos	USD - United States of America, Dollars
6099 - Total Revenue	389672805.8
1699 - Net Inventories	192558938
	30671437.83
	15068034.87

To add the formatting, click on the Edit Quick View Options in the Quick View Tab on the right side of the screen.

## Navigating the Excel Add-In

---



This is the Quick View after the update:

	2013M2 - Feb 2013	
	MXPISTN - Federal-Mogul Puebla Pistons	
	MXN - Mexico, Pesos	USD - United States of America, Dollars
6099 - Total Revenue	390	31
1699 - Net Inventories	193	15

## Creating Excel Forms and Reports

Well-designed Cube Views help you create flexible and rich Excel-based forms and reports. You can retrieve capabilities through formula, but inserting Cube Views in a spreadsheet offers a richer experience with less maintenance.

### Named Regions

Bringing a Cube View into Excel creates several Named Regions that you can select, refer to, and use with Styles. Named Regions are created for the Cube View, column headers, row headers, and data sections.

If there are multiple named columns or rows in the Cube View, go to the intersection-based Named Regions and use different formatting to differentiate sections. For example, a Total row is separated from detailed data. This combination of Named Regions and Styles generates a nicely formatted report:

	Net Income	Net Sales	Cost of Sales	Total Cur Assets
<b>Clubs</b>	5,719,400	204,339,789	114,594,599	282,477,228
<b>NA Clubs</b>	20,559,431	196,445,083	103,266,897	278,972,392
<b>Canada Clubs</b>	(15,679,089)	23,664,927	16,938,624	6,961,760
Montreal	(6,671,953)	10,070,182	7,207,925	2,962,451
Quebec City	(9,007,136)	13,594,745	9,730,699	3,999,309
<b>US Clubs</b>	35,947,682	173,219,127	86,642,475	272,149,769
Augusta	11,058,078	23,378,642	11,434,701	47,394,427
Carlsbad	11,920,655	17,421,183	5,151,048	40,585,020
Houston	12,983,949	132,419,302	70,056,727	184,172,322
<b>Frankfurt</b>	(11,216,955)	5,967,276	8,562,133	2,649,158

# Retrieve Functions

Retrieving and changing data can be done by using functions. To see the functions and their Parameters, open Excel and select the Formulas tab. Select Insert Function and select OneStreamExcelAddIn.XFFunctions where it says to Select a category.

The output of the function will look something like this: =XFGGetCell(A1,A2,A3,A4,A5)

The equivalent functions like XFGGetCell provide a separate Parameter to specify each Dimension Member without using the Member Script syntax. (e.g., E#CT:A#Sales would not be used, CT and Sales would be used in the correct Parameter for that Dimension)

Here are the main functions to use:

**NOTE:** If a field within the function is unneeded, enter a double quote to ignore it.

## XFGGetCell

This function retrieves data based on the Parameters supplied. Each Parameter needs to be defined.

XFGGetCell(NoDataAs Zero, Cube, Entity, Parent, Cons, Scenario, Time, View, Account, Flow, Origin, IC, UD1, UD2, UD3, UD4, UD5, UD6, UD7, UD8)

## XFGGetCell5

This has the same functionality as XFGGetCell except it limits the User Defined Dimensions to five instead of eight.

## XFGGetFXRate

This function retrieves rates from the system. Each Parameter needs to be defined.

XFGGetFXRate(DisplayNoDataAsZero, FXRateType, Time, SourceCurrency, DestCurrency)

## XFGGetCalculatedFxRate

This function directly retrieves an exchange rate even if only the inverse rate exists in the system.

## XFGGetMemberProperty

This function retrieves any Dimension Member property from the Member Properties tab in the Dimension Library. Note there are no spaces used when defining property name.

XFGGetMemberProperty("DimTypeName", "MemberName or Script", "PropertyName", "VaryByCubeTypeName", "VaryByScenarioTypeName", "VaryByTimeName")

**NOTE:** If the function does not need to vary by Cube Type, Scenario, or Time, enter a double quote in order to ignore it.

**Example:** Retrieving Currency for the Houston Entity

XFGGetMemberProperty("Entity", "Houston", "Currency", "", "", "")

**Example:** Retrieving an Account Formula that only occurs in the Budget Scenario  
XFGetMemberProperty("Account", "51000", "Formula", "", "Budget", "")

**Example:** Retrieving the Short Description Property for Time Dimension  
XFGetMemberProperty("Time", "2015M8", "ShortDescription", "", "", "")

### **XFGetRelationshipProperty**

This function retrieves any Dimension relationship property from the Relationship Properties tab in the Dimension Library.

XFGetRelationshipProperty("DimTypeName", "ParentMemberName or Script", "ChildMemberName or Script", "PropertyName", "VaryByScenarioTypeName", "VaryByTimeName")

**NOTE:** If the function does not need to vary by Cube, Scenario, or Time, enter a double quote in order to ignore it.

**Example:** Retrieving a Flow Members Aggregation Weight

XFGetRelationshipProperty("Flow", "TotalBalance", "Total Movement", "AggregationWeight", "", "")

**Example:** Retrieving an Entity's Percent Consolidation for July, 2015

XFGetRelationshipProperty("Entity", "Houston", "South Houston", "PercentConsolidation", "", "2015M7")

### **XFGetHierarchyProperty**

This function determines whether or not a Dimension has children and returns True or False

XFGetHierarchyProperty("DimTypeName", "DimName", "MemberName or Script", "PropertyName", "PrimaryCubeName", "ScenarioTypeNameForMembers", "MergeMembersfromReferencedCubes")

**Example:** Retrieving Child Hierarchy

XFGetHierarchyProperty("entity", "HoustonEntities", "Houston Heights", "HasChildren", "Houston", "Actual", FALSE)

### **XFGetDashboardParameterValue**

This function is available to Excel Add-in and Spreadsheet. If that function is used within an XLSX file that is using a function like XFGetCell or XFSetCell (or similar) where these are referencing a custom parameter value (e.g. ParamEntity) that is on the Dashboard that references this Spreadsheet from within it as a Component. The practice to get this Custom Parameter value is to use XFGetDashboardParameterValue to fetch the text from that Parameter or its default value and place it in a cell on the Spreadsheet (e.g. B1). Then the cell that is using a retrieve function such as XFGetCell would reference this other cell (i.e. B1).

XFGetDashboardParameterValue("myParamName", "Text For Default Value")

### **XFGetMemberInfo**

This function retrieves the description in the system. Each Parameter needs to be defined.

`XFGetMemberInfo(MemberInfoType, DimTypeName, MemberName, NameorDesc, NameandDesc)`

### **XFIInternalGetDataFromServer**

This function returns True or False. It does not take any arguments.

### **XFSetCell**

This function saves data to the amount field based on the Parameters supplied. Each Parameter needs to be defined.

`XFSetCell(CellValue, StoreZeroAsNoData, Cube, Entity, Parent, Cons, Scenario, Time, View, Account, Flow, Origin, IC, UD1, UD2, UD3, UD4, UD5, UD6, UD7, UD8)`

### **XFSetFXRate**

This function saves rates to the system. Each Parameter needs to be defined.

`XFSetFXRate(Value, StoreZeroAsNoData, FXRateType, Time, SourceCurrency, DestinationCurrency)`

### **XFGetCellUsingScript**

### **XFGetMemberInfoUsingScript**

### **XFSetCellUsingScript**

All of the functions that have ...UsingScript are based on a Member Script (e.g., A#Sales:E#Texas). The multiple Parameters provide the ability to specify multiple portions of the full Member Script using different Excel cells. All of the Member Scripts in the function Parameters combine to create one Member Script. It will then use the combined Member Script to retrieve the data cell.

### **XFGetCellUsingScriptEx**

### **XFGetMemberInfoUsingScriptEx**

### **XFSetCellUsingScriptEx**

All of the functions that have ...Ex have many more Parameters to use for combining Member Scripts (e.g., commonly used when creating another version of a function that has extra Parameters). Ex would also be used to combine many Member Script Parameters.

**XFSetCellLocalForms**  
**XFGetCellLocalAdjInput5**  
**XFGetCellLocalForms5**  
**XFGetCellLocallImport5**  
**XFGetCellLocalOTop5**  
**XFGetCellTransAdjInput5**  
**XFGetCellTranForms5**  
**XFGetCellTransImport5**  
**XFGetCellTransOTop5**  
**XFSetCellLocalForms5**

These functions use the Consolidation and Origin Dimensions. For example, XFSetCellLocalForms is using Local Consolidation and the Forms Origin Member. The number five at the end of the functions limit the User Defined Dimensions to five instead of eight.

**XFGetCellUSingScriptVolatile**  
**XFGetCellVolatile**  
**XFGetFXRateVolatile**  
**XFGetMemberInfoUsingScriptExVolatile**  
**XFGetMemberInfoUsingScriptVolatile**  
**XFGetMemberInfoVolatile**  
**XFSetCellUsingScriptExVolatile**  
**XFSetCellUsingScriptVolatile**  
**XFSetCellVolatile**  
**XFSetFXRateVolatile**

In some cases, Excel requires a volatile function for proper refreshing, for example, some Excel Charts that reference calculated cells.

**XFInternalPrepareCalculationStep**  
**XFInternalSendDatatoServer**  
**XFInternalSetConnectionInfo**

All of the functions that begin with XFInternal only work for internal processes.

## Visual Basic for Applications (VBA) Procedures

A Sub procedure is a series of Visual Basic statements that perform a specific task. It can take arguments, such as constants, variables, or expressions that are passed by a calling the procedure. If a Sub procedure has no arguments, the Sub statement must include an empty set of parentheses.

## Navigating the Excel Add-In

---

A Sub procedure begins with a Sub statement, followed by the tasks to be performed, and ends with an End Sub statement. The following snippet of VBA code represents the structure of a Sub procedure:

```
Sub [ProcedureName] (Arguments)  
[Statements]  
End Sub
```

OneStream Functions are used with Sub procedures to automate data submission from XFSetCells formulas.

**NOTE:** To use these functions in Excel, you must have the latest version of OneStream and its corresponding Excel add-in installed.

Procedures currently supported are:

- LogonAndOpenApplication - Login varies depending on the authentication type setup for your application:

**NOTE:** Resource Owner Password Credentials (ROPC) is no longer supported. As a result, the following ProcessSSOAuthenticationAndCreateToken has been deprecated. See Legacy Single Sign On (SSO) for Bearer Token procedure.

```
Sub SSOLogon()  
Set xfAddIn = Application.COMAddIns("OneStreamExcelAddIn")  
If Not xfAddIn Is Nothing Then  
If Not xfAddIn.Object Is Nothing Then  
ssoToken = xfAddIn.Object.ProcessSSOAuthenticationAndCreateToken  
("https://golfstream.onestreamtest.com/OneStreamWeb",  
"user1@mycompany.com", "P@$$w0&D")  
xfUserName = xfAddIn.Object.GetXFUserNameFromSSOToken(ssoToken)
```

- OneStream IdentityServer (OIS) - (url, PAT, application)  
Supports the use of Personal Access Tokens. Refer to the Identity and Access Management Guide for creating and managing PATS.

```
Sub OIS_PAT_Logon()  
Set xfAddIn = Application.COMAddIns("OneStreamExcelAddIn")  
If Not xfAddIn Is Nothing Then
```

## Navigating the Excel Add-In

---

```
If Not xfAddIn.Object Is Nothing Then
    ' OIS using PAT
    loggedIn = xfAddIn.Object.LogonAndOpenApplicationWithToken
    ("https://yoursite.onestreamcloud.com/OneStreamWeb", InsertPersonalAccessToken,
    "GolfStreamDemo_v36")
        MsgBox ("Is Logged In : " & loggedIn)
    End If
End If
End Sub
```

- Legacy Single Sign On (SSO) - (url, ssoToken, application)  
Supports the use of Bearer token from your identity provider

```
Sub SSOLogon()
    Set xfAddIn = Application.COMAddIns("OneStreamExcelAddIn")
    If Not xfAddIn Is Nothing Then
        If Not xfAddIn.Object Is Nothing Then
            ' Legacy SSO via Bearer Token
            Dim ssoToken As String
            ssoToken = <obtain Bearer token from your Identity Provider>
            loggedIn = xfAddIn.Object.LogonAndOpenApplicationWithToken
            ("https://yoursite.onestreamcloud.com/OneStreamWeb", ssoToken, "GolfStreamDemo_v36")
                MsgBox ("Is Logged In : " & loggedIn)
            End If
        End If
    End Sub
```

- Native Authentication (Self-hosted) - (url, user, password, application)  
Support login via userid and password

```
Sub Native_Logon()
    Set xfAddIn = Application.COMAddIns("OneStreamExcelAddIn")
    If Not xfAddIn Is Nothing Then
        If Not xfAddIn.Object Is Nothing Then
            ' Native - UserName & Password
            loggedIn = xfAddIn.Object.LogonAndOpenApplication
            ("https://yoursite.onestreamcloud.com/OneStreamWeb", UserName, Password,
            "GolfStreamDemo_v36")
                MsgBox ("Is Logged In : " & loggedIn)
            End If
        End If
    End Sub
```

- Logoff()

## Navigating the Excel Add-In

---

- RefreshXFFunctions() -- refer to the following example:

```
Sub RefreshXFFunctions()
Set xfAddIn = Application.COMAddIns("OneStreamExcelAddIn")
If Not xfAddIn Is Nothing Then
If Not xfAddIn.Object Is Nothing Then
Call xfAddIn.Object.RefreshXFFunctions
End If
End If
End Sub
```

- RefreshXFFunctionsForActiveWorksheet()
- RefreshQuickViews()
- RefreshQuickViewsForActiveWorksheet()
- RefreshCubeViews()
- RefreshCubeViewsForActiveWorkSheet()
- ShowParametersDlg()
- ShowParametersDlgForActiveWorksheet()
- SubmitXFFunctions () -- Automates the data loading process and eliminates the need to open the Excel files individually and submit data manually. Using a VBA routine, files with XFSET functions that are linked to other cells, sheets, and files can be programmatically submitted to OneStream. This procedure calls only XFSetCells. Refer to the following example:

```
Sub SubmitXFFunctionsTest()
Set xfAddin = Application.COMAddIns("OneStreamExcelAddin")
If Not xfAddin Is Nothing Then
If Not xfAddin.Object Is Nothing Then
Call xfAddin.Object.SubmitXFFunctions
Call xfAddin.Object.RefreshXFFunctions
End If
End If
End Sub
```

## Error Logs

The Excel Add-In stores error log details on your local drive in the AppData folder. This allows customers and customer support to retrieve log details when experiencing Excel Add-In related issues, resulting in decreased time to resolution.

Error logs in the folder will automatically be deleted after 60 days.