

The background of the image is a panoramic view of the Nashville skyline at dusk. The sky is a mix of deep blues and oranges from the setting sun. The city lights are beginning to glow, reflecting on the water of the river in the foreground. A prominent bridge with a white truss structure spans across the river. The text 'ODTUG Kscope 24' is overlaid in the center. 'ODTUG' is in a smaller, white, sans-serif font. 'Kscope' is in a large, white, sans-serif font. '24' is in a large, multi-colored, geometric font with shades of blue, green, yellow, and orange. Below the main title, the text 'nashville, tn' and 'july 14 - 18' is written in a smaller, white, sans-serif font. At the bottom center, the word 'Welcome' is written in a large, white, sans-serif font.

ODTUG
Kscope 24

nashville, tn july 14 - 18

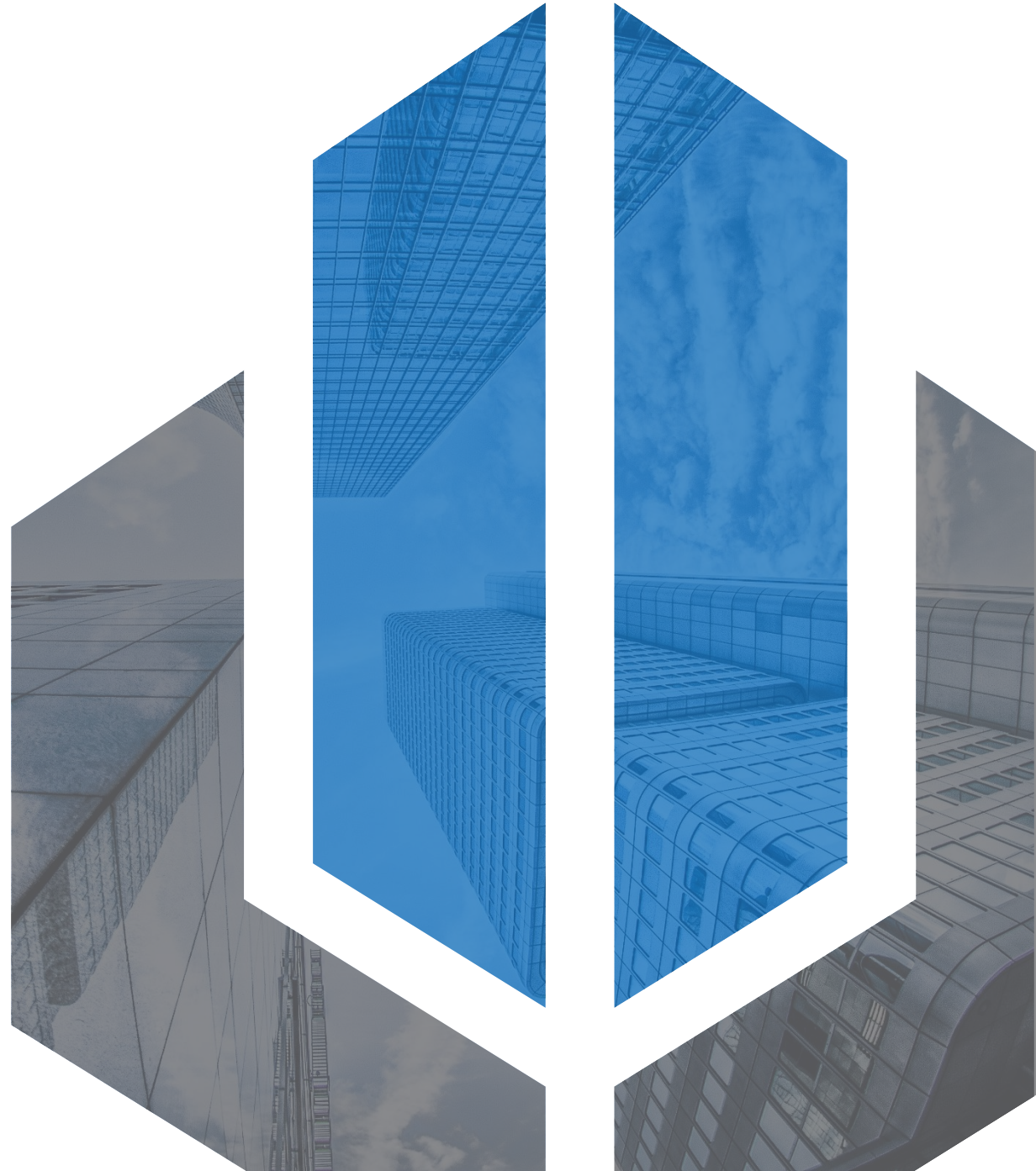
Welcome



ODTUG
Kscope24
nashville, tn july 14 - 18

FCC Calculations

Everywhere All At Once



AGENDA

Brian Marshall



Introductions



FCC Background



How Data is Stored



Calculations Types and Examples



Dynamic Calculations



Other Considerations

About the Speaker

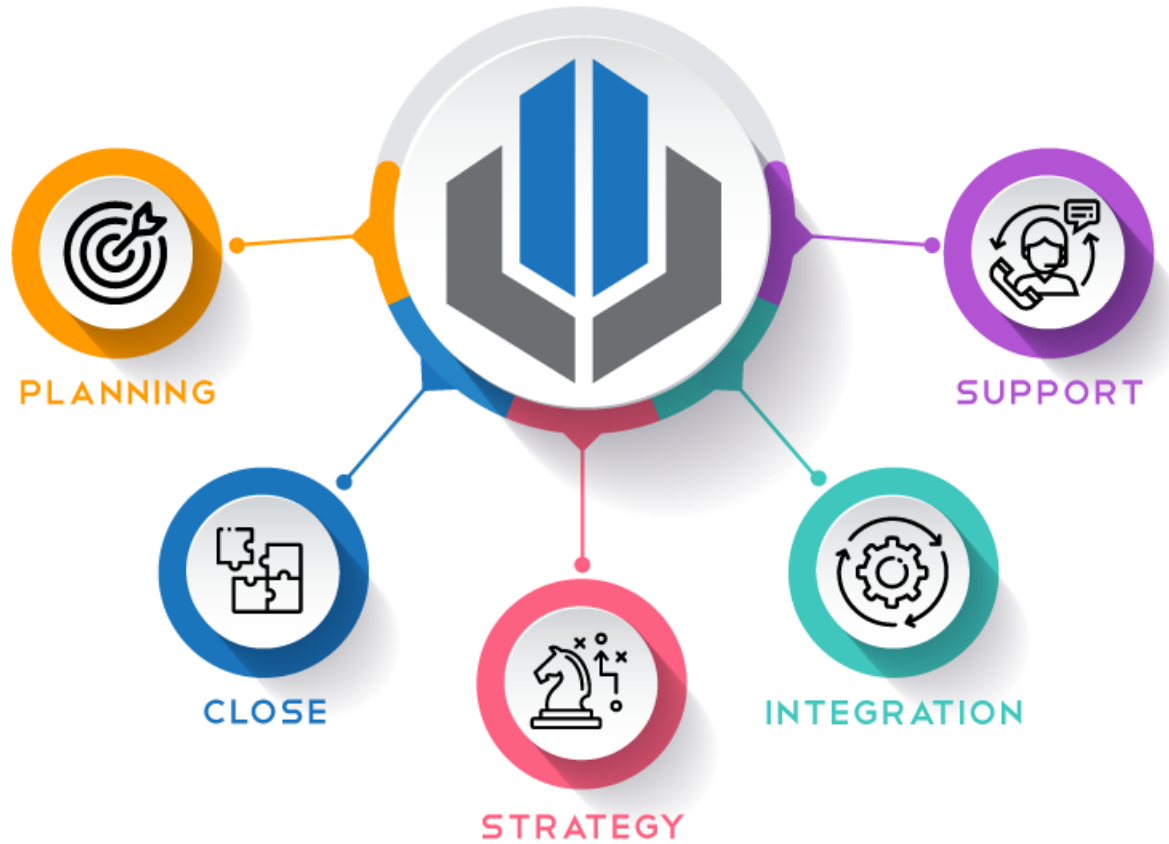


Brian Marshall

EPM Cloud
Architect

- 20+ years of Oracle EPM experience
- Oracle ACE Alumni
- Internationally recognized speaker
- Kscope Speaker 2010-2018, 2022-2024
- Kscope 2022 Best Overall Speaker, Kscope 2015 Best Speaker Award
- Over 250 projects
- Architected over 100 projects
- Experienced in FCC, Planning, ARCS, EDMCS, Cloud Data Management, Essbase (ASO, BSO and Hybrid), and HFM, Essbase Analytics, and replication of Consolidation functionality in both Essbase and Free Form Planning in PBCS)
- Extensive experience with Groovy and the REST APIs

About Olympus



Who We Are

Experienced EPM resources with innovative solutions and a diverse skillset to ensure that any project can be successfully executed.

Our Mission

Provide clients with the highest quality EPM experience using a proven methodology that ensures client interaction and satisfaction.

Our Vision

Empower clients to adopt excellent EPM solutions while having a positive impact on the EPM community at large through the knowledge we share.

Financial Consolidation and Close Cloud

An Overly Quick Introduction



- Oracle's Cloud-based replacement of Hyperion Financial Management (HFM)
- Highly customized version of Planning and Budgeting Cloud Service (PBCS)
- Both not at all like HFM And like HFM at the same time
- Provides all the functionality you would expect from a close product:
 - Currency Translation
 - Journal Entries
 - Intercompany Eliminations
 - Equity Pickup
 - Ownership Management
 - Supplemental Data Collection
 - Close Management

FCCS vs HFM

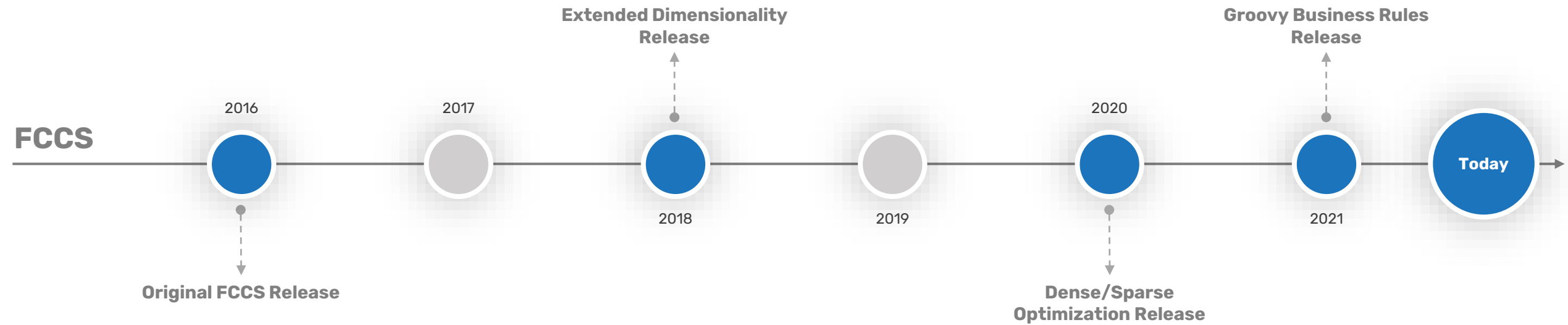
FCCS

- Cloud Only
- Based on PBCS using Essbase as the consolidation engine with a relational database to augment overall functionality
- Comes with more dimensions out of the box to support more built-in functionality
- Standard chart of accounts that provides a framework for various aspects of functionality built into the product
- Procedural logic can be built using traditional Essbase business rules, Groovy business rules, insertion rules, and configurable consolidation rules
- Dynamic member formulas defined in the meta-data and executed when data is requested

HFM

- On-Premise Only
- Based on a Relational Online Analytics Processing (ROLAP) technology with a relational database at the core of most functionality
- Less dimensions out of the box and less direct built-in functionality, but more customizable
- No standard chart of accounts, just functionality to differentiate between Income Statement and Balance Sheet accounts
- Procedural logic can be built using traditional HFM Business Rules that are based on VBA code, the only such mechanism in HFM
- Dynamic calculations are defined in a rules file and are also executed when data is requested

The History of FCCS



FCCS vs FCCS vs FCCS

Original FCCS Release

- The original release of FCCS, based on a Block Storage Option (BSO) Essbase database
- Limited to two (2) custom dimensions
- No ownership management
- Slower consolidation times
- Generally considered a not great product from Oracle, some may say closer to a beta release

Extended Dimensionality

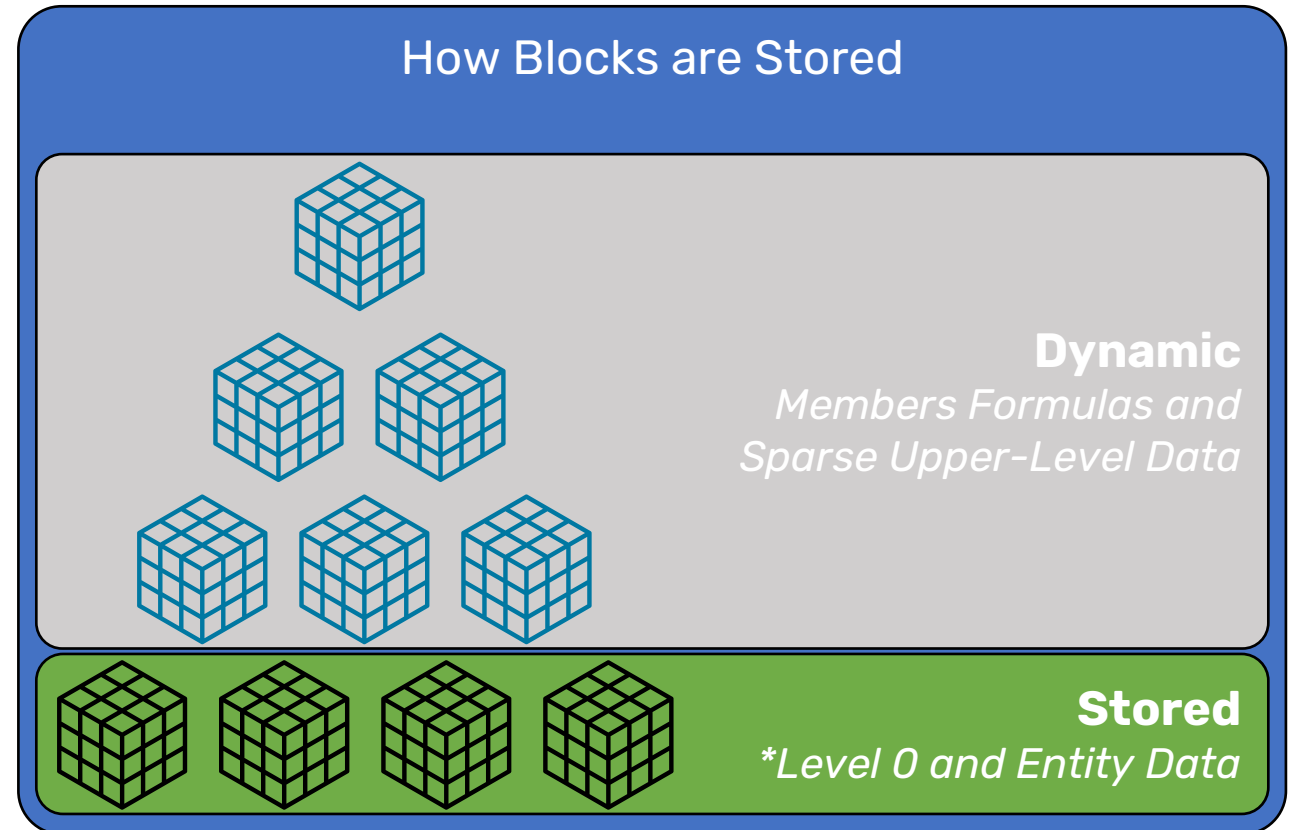
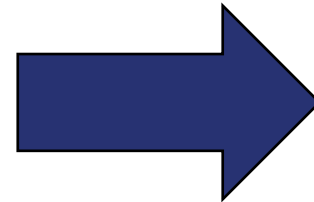
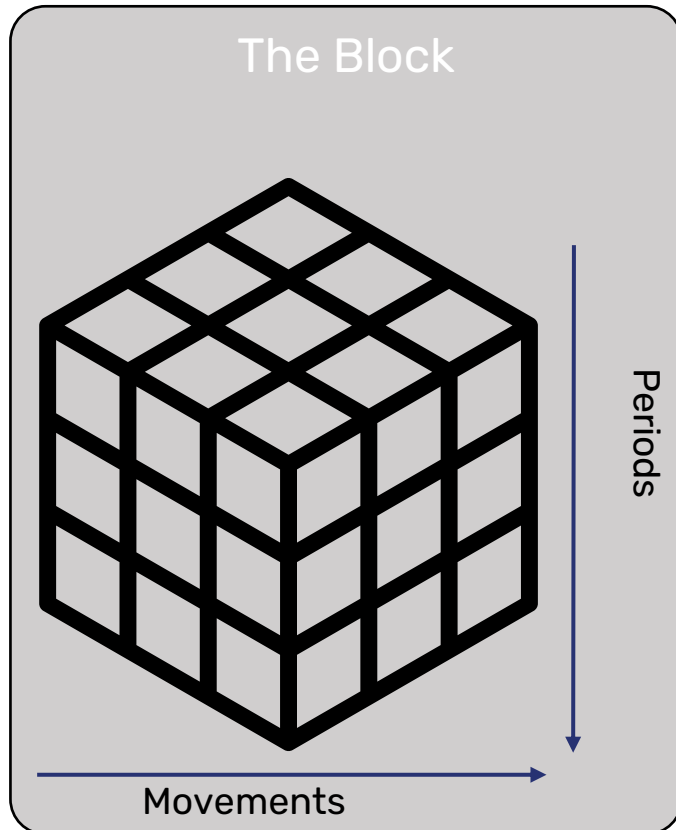
- The second release of FCCS, based on a Hybrid BSO Essbase database
- Increased dimension count to four (4) custom dimensions
- Added ownership management
- Improved consolidation times
- Generally, a far better product than the original release of FCCS

Dense-Sparse Optimization

- The current release of FCCS, still based on a Hybrid BSO Essbase database, but with significant changes
- Same custom dimension count limitations as Extended Dimensionality
- Same ownership management functionality as Extended Dimensionality
- Even more improved consolidation times, and FAR smaller databases
- The best release of FCCS to date and getting better every day

How Data is Stored in FCCS

Dense/Sparse Optimization



*When Ownership Management is enabled, the Proportion member is also stored in the consolidation dimension

So...Many...Dimensions

The Dense (what makes up the Block)

- Period
- Movement

The Sparse

- Account
- Consolidation
- *Currency (if enabled)*
- Data Source
- Entity
- *Intercompany (if enabled)*
- *Multi-GAAP (if enabled)*
- Scenario
- View
- Years

Why Is This Important? Block Creation...

- Why is the history of FCC important?
- Why do we care about dense and sparse?
- Block Creation!
- In short...Essbase does not guarantee that a block will be written to disk just because we think the calculation should
- Understanding the block structure is important to understanding the many ways to ensure that blocks are in fact created

The Many Ways to Calculate Data in FCC

- On-Demand Rules
 - Essbase Calculation Script
 - Groovy
- Configurable Rules
 - Insertion Point Rules
 - Equity Pickup
- Dynamic Calculations

On Demand Rules

Introduction



- Two main types of on-demand rules
 - Essbase Calculation Script
 - Groovy
- Executed outside the consolidation path (meaning a consolidation generally must be run after an on-demand rule is run)
- Can be attached to forms to execute at the time a user enters data via both the web-based interface and Smart View
- Can be executed independent of a form by an end-user or a pipeline (or EPM Automate/REST API)
- Can be debugged from calculation manager with full feedback and logs
- Status and results found in the **Jobs** section of FCC

On Demand Rules

Essbase Calculation Scripts



- The core mechanism for rules in FCC based on old-school Essbase calculation script language
- Includes a set of runtime prompts (restricted members) that are automatically part of a FIXPARALLEL statement:
 - Scenario
 - Period
 - Year
 - Entity
 - Consolidation
 - Currency
- Restricted members cannot be used in a FIX or the left side of an equals
- Use Cases
 - Allocations
 - Statistical Calculations
 - User-input based calculations

On Demand Rules

Essbase Calculation Scripts – Basic FIX Statement



- Let's look at a basic FIX statement:

```
1  FIX("EPS")
2
3      "Data Input" = "FCCS_Net Income" / "Shares Outstanding";
4
5  ENDFIX
```

- Our restricted members are automatically part of an invisible outer FIX and any other dimensions not listed in the FIX will be calculated for EVERY MEMBER IN THAT DIMENSION
- This example performs a very basic Earnings Per Share based on the run-time prompts for our restricted members

On Demand Rules

Essbase Calculation Scripts – Basic IF Statement

- Let's look at a basic IF statement:

```
1  FIX("EPS")
2
3      "Data Input" (
4          IF(@ISMBR("Specific Entity"))
5              "FCCS_Net Income" / "Shares Outstanding";
6          ENDIF
7      )
8
9  ENDFIX
```

- This example performs a very basic Earnings Per Share based on the run-time prompts for our restricted members, but also adds in a check for a specific Entity
- Keep in mind that our Entity cannot be used in a FIX, but it can be used in an IF to ensure that EPS, in this case, is only calculated for that specific Entity

On Demand Rules

Essbase Calculation Scripts



Pros

- Will always be the fastest way to calculate data
- Can be attached to forms, executed independently or from a pipeline
- Large set of professionals with existing experience in Essbase Calculation Script language
- Executed outside of the consolidation path

Cons

- Block creation will always be a consideration and pain point
- No visibility to changes made by the calculation
- Executed outside of the consolidation path

On Demand Rules

Groovy

- Newly introduced in 2021...adds a near limitless set of functionality
- Still subject to restricted dimensions and includes the same runtime prompts:
 - Scenario
 - Period
 - Year
 - Entity
 - Consolidation
 - Currency
- Use Cases
 - Allocations
 - Statistical Calculations
 - User-input based calculations
 - Data loads from external sources

On Demand Rules

Groovy – The World's Quickest Introduction



- Groovy is available for any Oracle Cloud EPM Enterprise customers (this is NOT available for Standard customers)
- Business Rules can be written in Groovy in addition to traditional Calculation Script language
- Groovy language for the Java platform with simplified syntax and access to many Java classes
- Oracle Cloud EPM isn't exactly like a full Groovy environment you can install on your computer and has no ability to directly import classes
- Luckily, you will seldom find that the classes you need are not already there

On Demand Rules

Groovy



- Groovy gives us a near unlimited level of flexibility in how we interact with data and even external sources with FCCS
- For math inside the cube, we can get around issues with Hybrid Essbase and Block creation using Groovy grids that retrieve data from and send data to our underlying Essbase database
- We can even use Groovy to access external data sources to load data to our system (extremely handy for things like currency rates)
- Any interaction with data performed by Groovy writes to the audit logs as if a user specifically modified the data manually...which is amazing for so many reasons
- Groovy in FCC also maintains the overall integrity of how we interact with our data, whether we like it or not

On Demand Rules

Groovy – Writing to a Grid



- With Groovy, we write to Essbase using the same Grid API that Planning forms and Smart View use. This script defines a grid to write data back to Essbase:

```
1 Cube bsoCube = operation.application.getCube("Consol")
2 DataGridBuilder writer = bsoCube.dataGridBuilder("MM/DD/YYYY")
3 writer.addPov('Parent Currency','Parent Input',sPeriod,sYears,'Minority Interest Income Out',
4             'FCCS_Periodic','Minority Interest Income',sScenario)
5 writer.addColumn('My Entity')
```

- In FCC the cube name will always be Consol
- We add a Point of View which includes all of our single member selections
- We add a column that will coincide with the data we submit

On Demand Rules

Groovy – Writing to a Grid



- We add a row of data with as many members as we would like in the first bracket set
- Note that we don't have to define the dimension associated with the members listed, FCC does not allow duplicate member names, so it will be smart enough to find the member based purely on the name
- The second bracket set includes the actual set of data

```
6  
7 writer.addRow(["Row Member"], [12345689])  
8
```

On Demand Rules

Groovy – Writing to a Grid



- Now we build a status variable to check to see if our data will successfully submit
- This step is where FCC ensures that we maintain our overall data integrity and enforces our restricted members (so don't bother trying to use this as a way around those restrictions...I tried...like a lot of times)
- Once we issue the build command we can check the status to see if our cells are accepted or rejected (or both)
- Finally we actually save the grid back to Essbase

```
9   DataGridBuilder.Status status = new DataGridBuilder.Status()
10
11   DataGrid grid = writer.build(status)
12   println("Total number of cells accepted: " + status.numAcceptedCells)
13   println("Total number of cells rejected: " + status.numRejectedCells)
14   println("First 100 rejected cells: " + status.cellsRejected)
15
16   bsoCube.saveGrid(grid)
```

On Demand Rules

Groovy – Writing to a Grid



- Putting it all together:

```
1 Cube bsoCube = operation.application.getCube("Consol")
2 DataGridBuilder writer = bsoCube.dataGridBuilder("MM/DD/YYYY")
3 writer.addPov('Parent Currency','Parent Input',sPeriod,sYears,'Minority Interest Income Out',
4             'FCCS_Periodic','Minority Interest Income',sScenario)
5 writer.addColumn('My Entity')
6
7 writer.addRow(["Row Member"],[12345689])
8
9 DataGridBuilder.Status status = new DataGridBuilder.Status()
10
11 DataGrid grid = writer.build(status)
12 println("Total number of cells accepted: " + status.numAcceptedCells)
13 println("Total number of cells rejected: " + status.numRejectedCells)
14 println("First 100 rejected cells: " + status.cellsRejected)
15
16 bsoCube.saveGrid(grid)
```

On Demand Rules

Groovy

Pros

- Can completely mitigate block creation issues
- Full audit logs available of every change made from a Groovy script
- Can be attached to forms, executed independently or from a pipeline
- Executed outside of the consolidation path

Cons

- Smaller set of professionals with experience in Groovy rules
- Steeper learning curve for non-programmers
- Executed outside of the consolidation path

Highly Recommended Session

Advanced Calculations for Planning and Essbase

Tuesday @ 10:30 am in Cheekwood C



Whether you're writing calc scripts or business rules, the process, syntax and techniques are largely the same. This session will cover advanced calculation techniques including Block Mode vs Cell Mode, Bottom-up vs Top-Down, String Functions and functions that leverage Regular Expressions. If you've ever wondered what you can and can't do on the left side of an equation, this is the session for you!

Brought to you by the **27th** most influential person at Kscope24

Configurable Rules

Introduction



- Two main types of configurable rules
 - Insertion Point Rules (Essbase Calculation Script)
 - Equity Pickup Rules
- Executed inside the consolidation path (meaning they are part of the consolidation process)
- Can only be run as part of the actual consolidation of a period
- Cannot be debugged from calculation manager

Configurable Rules

Insertion Point or Insertion Rules



- Essbase Calculation Script based rules that are executed at defined points during the consolidation process
- Modified in calculation manager through specific rules
- These specific rules vary based on the currency configuration of the application
- For Single Currency Applications the rules are split into two sections: Entity and Parent
- For Multiple Currency Applications the rules are split into three sections: Entity, Translated, and Parent
- Each of these sections is then split into two categories: before balancing the balance sheet and after

Configurable Rules

Insertion Point Rules (Single Currency)



Rule	Insertion Point	Level
FCCS_110_After Opening Balance Carry Forward	After Balance Carry Forward but before balancing of Balance Sheet	Entity Level
FCCS_120_Final_Calculations	After balancing of Balance Sheet	Entity Level
FCCS_130_After Opening Balance Carry Forward	After Balance Carry Forward but before balancing of Balance Sheet	Consolidated Level
FCCS_140_Final_Calculations	After balancing of Balance Sheet	Consolidated Level

Configurable Rules

Insertion Point Rules (Multiple Currency)



Rule	Insertion Point	Level
FCCS_10_After Opening Balance Carry Forward	After Balance Carry Forward but before balancing of Balance Sheet	Entity Level
FCCS_20_Final_Calculations	After balancing of Balance Sheet	Entity Level
FCCS_25_Before FX_Calcs	Before Foreign Exchange (FX) Calculations	Translated Level
FCCS_30_After Opening Balance Carry Forward	After Balance Carry Forward but before balancing of Balance Sheet	Translated Level
FCCS_40_Final_Calculations	After balancing of Balance Sheet	Translated Level
FCCS_50_After Opening Balance Carry Forward	After Balance Carry Forward but before balancing of Balance Sheet	Consolidated Level
FCCS_60_Final_Calculations	After balancing of Balance Sheet	Consolidated Level

Configurable Rules

Insertion Point Rules



Pros

- Executed inside of the consolidation path limited the need for any additional steps
- Large set of professionals with existing experience in Essbase Calculation Script language

Cons

- Block creation will always be a consideration and pain point
- No visibility to changes made by the calculation

Configurable Rules

Equity Pickup

- Pre-defined rules and members meant to calculate, consolidate, and store equity pickup information and associated accounts
- Part of ownership management functionality and makes use of ownership percentages

Consolidation - Enable Features

Consolidation Journal Adjustments

Consolidation Journal Workflow

Accounts Reporting

Balance Sheet Hierarchy

Traditional Balance Sheet Approach

Net Asset Approach

Basic

Ownership Management

Equity Pickup

Configurable Rules

Equity Pickup



- When enabled, additional meta-data is added:
 - Accounts for Income Statement and Balance Sheet
 - Data Source member to track data through the process
- No-code approach with seeded rules:
 - Prepare Data Source
 - Consolidate Data Source
 - Reverse Holding Company
- Best practice is to create copies of seeded rules and deploy and active those new rules to leave the seeded rules intact

Dynamic Calculations

Introduction



- All the typical dynamic flexibility we expect from an Essbase-based application
- Dimensions where dynamic calculations can be added:
 - Account
 - Data Source
 - Scenario
- This is not your grandfather's Essbase...this is Hybrid Essbase
- Solve order matters...a lot

Solve Order...moved again

Member Name	Parent Member	Default Data Storage	Consol Data Storage
▼ Account		Never Share	Never Share
▶ FCCS_System Account	Account	Label only	Label only
▶ FCCS_Balance Sheet	Account	Dynamic Calc	Dynamic Calc
▶ FCCS_Income Statement	Account	Dynamic Calc	Dynamic Calc
▶ FCCS_Drivers	Account	Label only	
Audit Groupings (ER)	Account	Dynamic Calc	
▼ Management Reporting	Account	Label only	
▶ Total Gross Profit	Management Reporting	Dynamic Calc	
▼ Reporting Metrics	Management Reporting	Label only	
Gross Profit % Sales	Reporting Metrics	Dynamic Calc	
Gross Profit % Net Income	Reporting Metrics	Dynamic Calc	
Operating Income % Sales	Reporting Metrics	Dynamic Calc	
Net Income % Sales	Reporting Metrics	Dynamic Calc	
Net Income % Gross Profit	Reporting Metrics	Dynamic Calc	

Right Click Here

Select Columns Reset to Default OK Cancel

Select All

- Description
- UDA
- ▶ Member Formula
- ▶ Member Formula Description
- ▶ Solve Order
 - Consol Solve Order
- Intercompany Account
- Is Plug Account
- Plug Account
- Default Movement
- Enable for Dynamic Children
- Number of Possible Dynamic Chil...

Selections

- Parent Member
- Default Data Storage
- Consol Data Storage
- Data Type
- Consol Consol op.
- Two Pass Calculation
- Smart Lists
- Account Type
- Variance Reporting
- Time Balance
- Exchange Rate Type
- Default Alias Table

Other Olympus Sessions

EPM Security – Zero to Sixty

Tuesday @ 10:30 in Checkwood F



EPM offers a number of security solutions and mechanisms, but there's a fine balance to create a truly efficient, maintainable security footprint. In this session we'll walk through the available security layers in order of granularity and talk through best practice scenarios and use cases for each. We'll also talk through some real-world security pitfalls to avoid and how to streamline your security maintenance.

- OCI provisioning
- Creating and Managing Access Control Groups
- Valid Intersections
- Cell-Level Security
- Security reporting/troubleshooting

Reports, Books, and Bursting Oh My!

Tuesday @ 11:30 in Checkwood F



Do I need a Book or just a Report? What are all the delivery options available with Bursting? Which option is right for me? We'll put you on the Yellow Brick Road as we walk through different reporting use cases and show you how to use the functionality within the Reports module in EPM to become a Wizard of Reporting.

- Using a book to generate multiple POVs of a given report
- Using a book to generate a report package of multiple reports for the same POV
- Sending all POVs to a user/users
- Sending custom POV to custom User (each user gets their own specific POV)
- Generating multiple POV's in separate files vs a combined single file

THANK YOU

Brian Marshall



972.998.0878



www.olympusconsulting.com



brian@olympusconsulting.com



Frisco, Texas



ODTUG Kscope24

nashville, tn

july 14 - 18



Don't Forget To Fill Out Your Evals