



Cube View Comments Guide

Block Version: 2.0.0

Minimum Genesis Version: 2.0.0

Minimum Platform Version: 9.0.0

Copyright © 2026 OneStream Software LLC. All rights reserved.

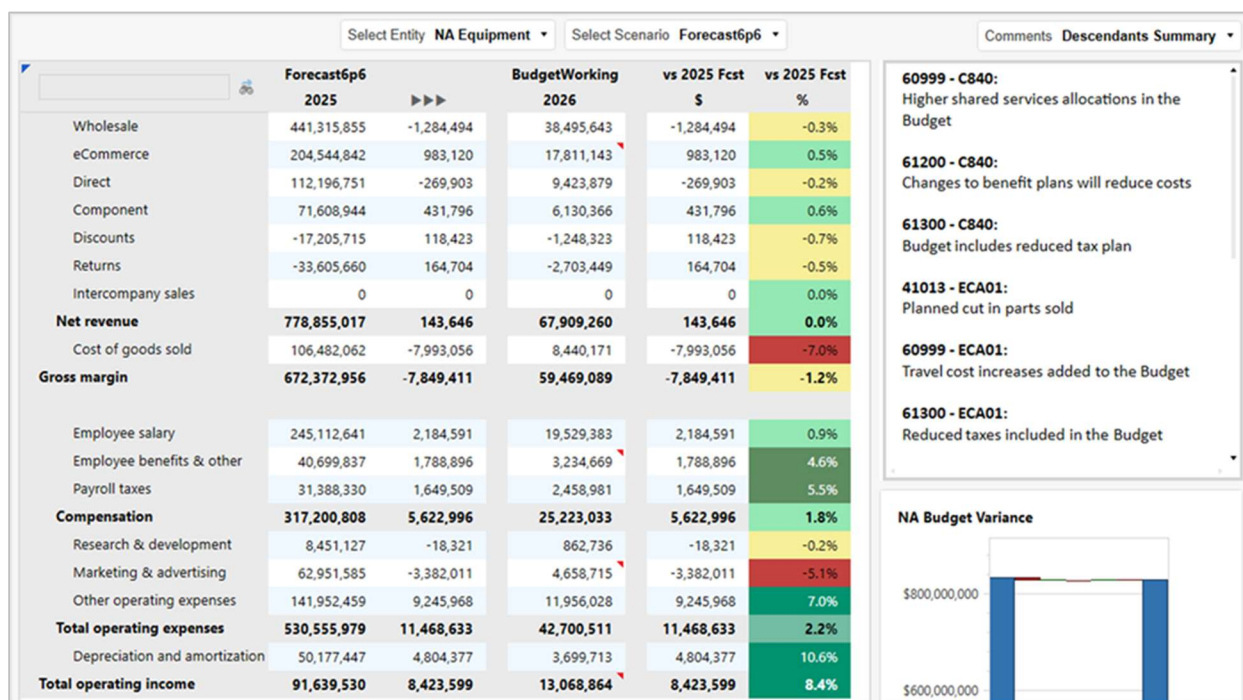
All trademarks, logos, and brand names used on this website are the property of their respective owners. This document and its contents are the exclusive property of OneStream Software LLC and are protected under international intellectual property laws. Any reproduction, modification, distribution or public display of this documentation, in whole or part, without written prior consent from OneStream Software LLC is strictly prohibited.

Table of Contents

- Overview 3
 - Use Cases 3
- Designer 4
 - Cube View Settings 4
 - Filters Settings 5
 - Define Filter Settings 6
 - Buttons Settings 10
 - Comments Settings 11
 - Layouts Settings 15
- Appendix 1 16
 - Workspace Requirements and Considerations 16
- Release Notes 17
 - Version 2.0.0 17
 - Version 1.2.0 19
 - Version 1.1.0 20
 - Version 1.0.0 20

Overview

The Cube View Comments content block contains three content cards. Content cards 1 and 2 display a side-by-side view of a Cube View and an editable text box. The text box is used specifically for Cube View commentary such as Annotations or Variance Explanations. The commentary configuration provides options to enter comments and analyze them in a variety of ways such as a summary or aggregated view. Filters are dynamically generated for each Parameter in the Cube View which generate Combo Boxes and/or Member selectors for dynamic Member selection. The optional third content card allows you to add supplemental content such as charts or drill down analysis while performing all core interactions with the Cube View.



Use Cases

- **Cube View Commentary Data Entry**
 - Entering standard OneStream comments like variance explanations, annotations, footnotes, etc. in a rich text format.
- **Cube View Commentary Dynamic Analysis**
 - Analyzing a summary view of all comments within the Cube View or a filtered summary view by Entity, Account hierarchy, time period, etc.
 - Analyzing an aggregated view of the comments based on a Parent Entity.
- **Cube View Commentary Entry/Analysis with additional analysis or visualizations**
 - Displaying supplemental content to provide additional details while entering and analyzing commentary.

Designer

After injecting the block, use the Designer page to configure it. The page is divided into five core sections: Cube View, Filters, and Buttons, Comments, Layout. The settings in each section control Cube View behaviors and capabilities and are configured based on user interaction and consumption.

Cube View Settings

Cube View	Filters	Buttons	Comments	Layouts
Cube View Settings				
Select Cube View	<input type="text" value="BudgetReview_BUDRVW"/>			
Description	<input type="text" value="Budget Review with Comments"/>			
Default Header	<input type="text" value="Hidden"/>			

Select Cube View

Assign a Cube View to the block. Select the Workspace where the Cube View is stored and use the Contains filters to refine your search results.

Cube View Selector

Select Workspace:

Filter (Contains):

- Default
 - Maintenance Units**
 - Default
 - Cube View Groups**
 - Budget Review (BUDRVW)
 - BudgetReview_BUDRVW**

NOTE: Workspaces must be **shareable**, or they will not appear in the drop-down menu. Set the Shareable property to True on all applicable Workspaces; this excludes the Default Workspace.

See *Appendix 1* for more details on Workspace requirements and considerations.

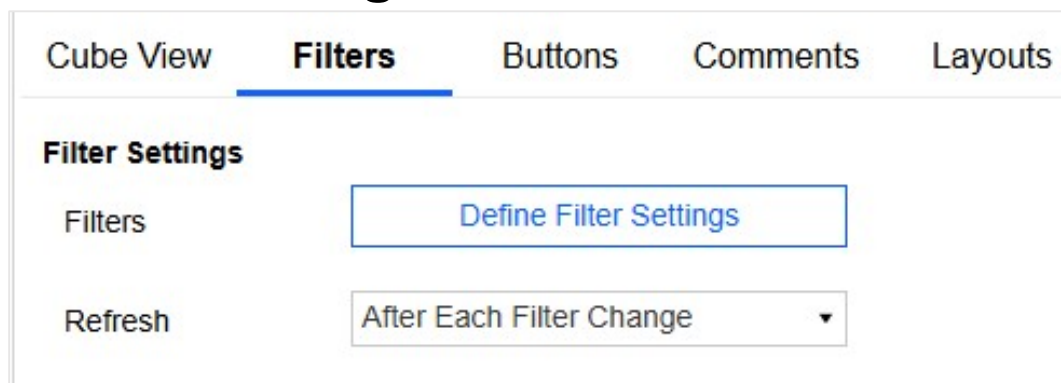
Description

The description is used as the Cube View's main header. If the Cube View has an existing Description, it will automatically display in this field. You can modify the description or add one if desired.

Default Header

The Default Header hides or displays the standard Cube View Toolbar. This is hidden by default; to display the toolbar, select Visible.

Filters Settings



The screenshot shows a settings panel with five tabs: 'Cube View', 'Filters', 'Buttons', 'Comments', and 'Layouts'. The 'Filters' tab is selected and highlighted with a blue underline. Below the tabs, the section is titled 'Filter Settings'. There are two rows of settings: the first row has a label 'Filters' and a button labeled 'Define Filter Settings'; the second row has a label 'Refresh' and a dropdown menu currently showing 'After Each Filter Change' with a downward arrow.

Filters

Click Define Filter Settings to view and manage the Cube View's parameters.

Refresh

Choose to automatically refresh the Cube View after each filter selection or manually refresh as needed.

NOTE: If you're using Parameters for row/column sharing, ensure the Parameter has a Default Value to avoid error upon rendering.

Define Filter Settings

Any Member Parameters in the Cube View will automatically be included as Filters. Literal Value and Input Parameters are excluded.

Cube View Parameters

Parameters + ↺ 🗑️

- prm_SelectWFEntity_ACTCLS
- prm_SelectGrowthTrend_STDOBJ

Parameter Properties

Parameter	prm_SelectGrowthTrend_STDOBJ	Display Order	1
Parameter Type	MemberList	Filter Type	Combo Box
User Prompt	Select Trend	New User Prompt	
Override Value	<input type="text"/>	Used By	ActBudPnLComments_ACTFRM
Default Value	Acquisition		

Important: Changing the Default Value updates the original parameter's Default Value setting.

Close Save

Manual Add

Allows users to add parameters that were not automatically detected by the Cube View block.

- Click Add Parameter.
- In the Select Parameter dialog:
 - Choose a workspace from the dropdown.
 - Browse or filter parameters in the tree view.
 - Select the parameter, then click Save.

Requirements:

- Parameter must already exist in a shareable workspace.
- Parameter must be assigned to the Cube View before adding it to the block.

Common Use Cases:

1. New Parameter Added to Cube View
 - If a Cube View is already assigned to the block and a new parameter is added later, the block will not auto-detect it.
 - Users can manually add the parameter so it appears in the filter toolbar.
2. Row/Column Sharing Beyond Level 1
 - The block auto-detects parameters in:
 - a. **Level 0:** Assigned Cube View
 - b. **Level 1:** Cube Views used for row/column sharing
 - Parameters beyond Level 1 are not detected automatically. Users can add these manually to support additional prompts.
3. Missed Parameters
 - If the block fails to detect valid parameters, users can add them manually as a safeguard.

Manual Remove

Remove a parameter from Filter Settings and the filter toolbar display. Does not delete the parameter from the workspace.

- Select the parameter and click Remove Parameter.

Display Impact:

- If removed from both the block and Cube View → parameter no longer appears or prompts.
- If removed from the block but still in the Cube View → user will be prompted for a value before the page renders.

Redetect Parameters

Performs a complete rescan of the assigned Cube View and replaces the current parameter list with the parameters detected in that Cube View. This ensures that the block reflects the Cube View's most current parameter configuration.

- Click Redetect Parameter.

Behavior:

When you select Redetect Parameters:

1. Removes all existing parameters currently listed in the block's Filter Settings.
2. Performs a complete rescan of the assigned Cube View.

3. Detects and adds all supported parameter types.
4. Repopulates the list with detected parameters using default settings.

Impact:

All manual edits are lost. Overrides and customizations must be re-applied after redetection

Parameter Properties

To configure a parameter's filter behavior, select it and modify its properties:

- **Display Order**
Enter a numeric value to set the sequence in which parameters are displayed.
- **Filter Type**
Determines how the user selection displays:
 - **Combo Box:** Displays a drop-down list of available values. (default option)
 - **Member Selector:** Displays a hierarchical member tree for selection.
 - **Hidden:** Parameter does not appear in the filter toolbar. The Override Value is used as the parameter value instead. If no Override Value, the user is prompted on page launch to select a value.
- **New User Prompt**
Defines the text displayed as the parameter label in the filter toolbar. Enter a descriptive message that guides the user. For example, *Select a Product member* or *Enter your name*. If left blank, the parameter Name will display as the label.
- **Override Value**
Assign a static value to the parameter. Users will not be able to modify or interact with the filter when viewed on the page. Enter the value directly or click the Select Member button to choose from a lookup dialog.
 - **Enter a Member Name**
 - Used when overriding a Parameter that prompts for Members such as Entity or Time.
 - Any member name can be entered; it does not have to be one of the Parameter values.
 - Does not require a Dimension Token.
 - **Enter a Cube View Name**
 - Used when overriding a Parameter that prompts for Cube View column or row sets.

- Any Cube View Name can be entered; it does not have to be one of the Parameter values.

NOTE: Invalid Override Value Behavior

If the Override Value entered is invalid, the Cube View will not render, consistent with current POV error handling in Cube Views. Invalid scenarios include:

- Typo or non-existent value
- Value from a different Dimension Type
- The user does not have permission to access the specified member or Cube View.

- **Default Value**

Sets the initial value when the parameter runs for the first time. Enter the value directly or click the **Select Member** button to choose from a lookup dialog.

NOTE: Important Behavior for Default Value

Changing the Default Value in Filter Settings does not apply only to the block, it updates the original parameter's Default Value property at the source level.

Implications:

- The new value becomes the parameter's default across all contexts where it is used.
- This change is persistent and affects any other Cube Views or blocks referencing the same parameter.

Best Practice:

Confirm that the updated default is appropriate for all dependent views before saving.

Buttons Settings

The buttons section gives you the option to include Save and/or Calculate buttons.

Cube View	Filters	Buttons	Comments	Layouts
Save Button				
Enable	True			
Tool Tip	Save			
Calculation Button				
Enable	False			

NOTE: Save is required if the user is entering comments. The standard Cube View save button will display if you set the Default Header to Visible, otherwise make sure to enable the Save button here.

Calculation Button

The Calculation Action on the Calculate button can perform a Save & Calculate or a stand-alone Calculate. Calculation Type includes all standard OneStream calculation options such as Force Calculate or Consolidate, execute Dashboard Extenders, Custom Calculates or Data Management Sequences.

Calculation Button	
Enable	True
Tool Tip	Calculate
Calculation Action	None
Calculation Type	Execute Data Management Sequ...
Calculation Details	
<pre>{ForecastPlan.RollForecast} {prm_EntityOverview_FCST=ForecastPlan.[~] prm_EntitvOverview_FCSTI~1</pre>	

NOTE: The syntax for Parameters and Substitution Variables is different when used in Genesis:

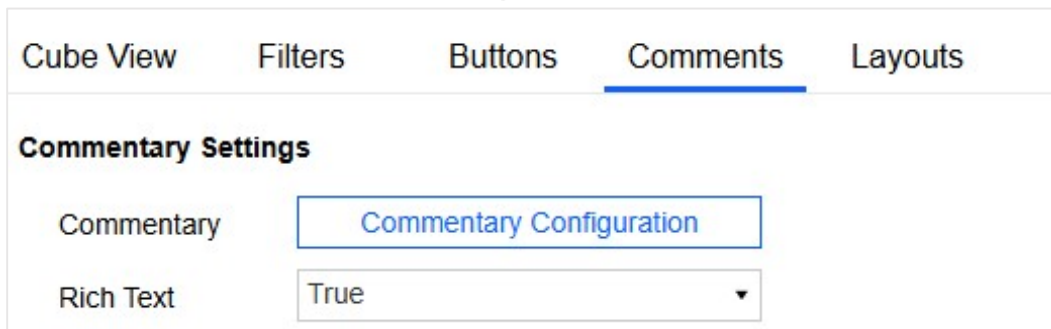
Parameter Syntax: ~|ParameterName|~

Substitution Variable Syntax: ~SubVariableName~

Specify the Workspace Name in the calculation arguments where applicable:

- WorkspaceName.DataManagementSequenceName
- WorkspaceName.ParameterName

Comments Settings



Cube View	Filters	Buttons	Comments	Layouts
Commentary Settings				
Commentary	Commentary Configuration			
Rich Text	True ▼			

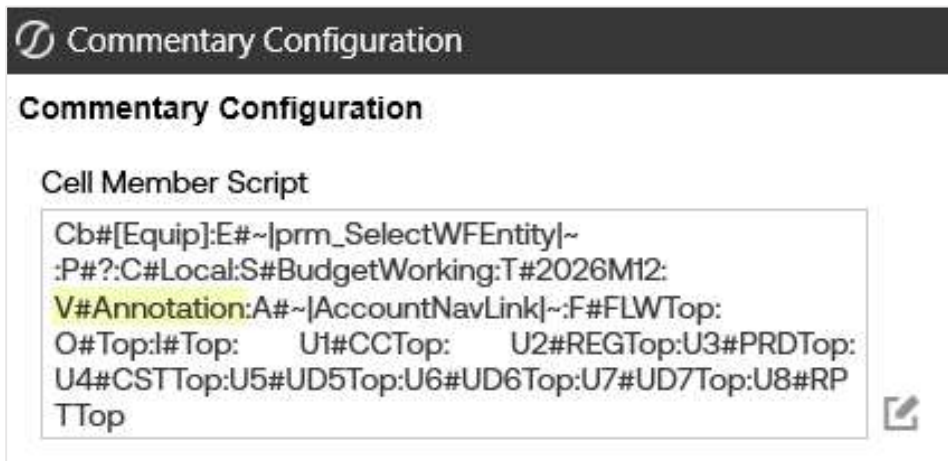
Comments are both entered and reviewed via the built-in text box. Users have four ways they can interact with these comments based on the commentary configuration.

- 1) **Edit:** used to enter and/or edit comments for a selected cell.
- 2) **View:** displays the stored comment for a selected cell.
- 3) **Cube View Summary:** Provides a broader view of all comments within a Cube View based on the specified Member Filters. Displays comments for all related cells within the Member hierarchy, not just the selected cell.
- 4) **Descendants Summary:** Provides an aggregated view of comments based on the specified Member Descendants. Comments are aggregated for all related cells within the hierarchy, not just a selected cell. This is useful for providing a more detailed and inclusive view of the commentary.


This page is where you configure and manage the commentary options as they relate to the Cube View. This is designed to enhance the way we enter, view and aggregate commentary, providing a richer and more interactive experience for users.

Commentary Configuration is where you define:

- **Cell Member Script:** This script defines the intersection where the comments are entered, stored and viewed.
 - The View Member must be one of the standard OneStream commentary Members such as Annotation or Variance Explanation.
 - The cells do not have to be in the Cube View.
 - This essentially replaces the need for a comments column in the Cube View and improves the overall user experience.



- **Summary Filters:** The filters expand upon the Cell Member Script to provide the Summary View.
 - Enter a comma delimited list of Member Filters where you want to create a broader view. This will display all comments related to the Member hierarchy.
 - Typically used for showing all Accounts in a Cube View.

 **Commentary Configuration**

Commentary Configuration


Cell Member Script

Cb#[Equip]:E#~|prm_SelectWFEntity|~
 :P#?:C#Local:S#BudgetWorking:T#2026M12:
 V#Annotation:A#~|AccountNavLink|~:F#FLWTop:
 O#Top:|#Top: U1#CCTop: U2#REGTop:U3#PRDTop:
 U4#CSTTop:U5#UD5Top:U6#UD6Top:U7#UD7Top:U8#RP
 TTop

Summary Filters

A#96999.DescendantsInclusive

- **Descendants Summary Filters:** The filters expand upon the Cell Member script to provide comment aggregation.
 - Enter a comma delimited list of Member Descendant filters where you want to see aggregated comments. This will display all comments related to Member hierarchy.

 **Commentary Configuration**

Commentary Configuration

Cell Member Script

Cb#[Equip]:E#~|prm_SelectWFEntity|~
 :P#?:C#Local:S#BudgetWorking:T#2026M12:
 V#Annotation:A#~|AccountNavLink|~:F#FLWTop:
 O#Top:|#Top: U1#CCTop: U2#REGTop:U3#PRDTop:
 U4#CSTTop:U5#UD5Top:U6#UD6Top:U7#UD7Top:U8#RP
 TTop

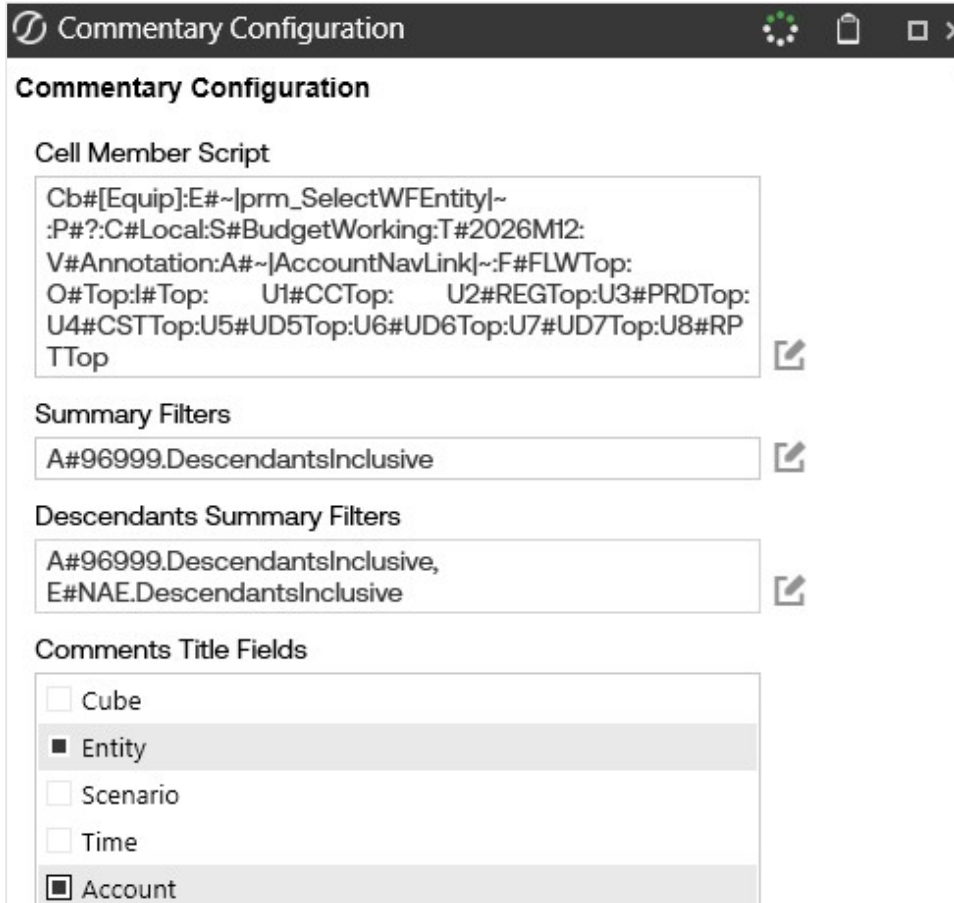
Summary Filters

A#96999.DescendantsInclusive

Descendants Summary Filters

A#96999.DescendantsInclusive,
 E#NAE.DescendantsInclusive

- **Comments Title Field:** provides a header for each comment using Dimension Member Names. This presents the comments in a more structured way making it easier for users to understand the context and relevance of each comment.
 - Enable the Dimensions you want included in the header.



NOTE: You can use Parameters in all comment configuration fields to make the comment POV dynamic.

Dynamic based on Member Selection:

- Parameters used in combo box/member selector.
- Use the same Member Parameter in the configuration scripts.

Dynamic based on Selected Cell:

- Use Default Navigation Links to specify which Dimensions this applies to.
- Default Navigation Links are enabled on the Cube View by setting the property to True.
- Bound Parameters are not supported.

Parameter Syntax: ~|ParameterName|~

Default Navigation Links:

~|CubeNavLink|~, ~|EntityNavLink|~, ~|ParentNavLink|~,~|ConsolidationNavLink|~,
 ~|ScenarioNavLink|~, ~|TimeNavLink|~, ~|ViewNavLink|~,
 ~|AccountNavLink|~,~|FlowNavLink|~, ~|OriginNavlink|~, ~|ICNavLink|~,
 ~|UD1NavLink|~, ~|UD8NavLink|~

Layouts Settings

This block contains two content cards by default (Cube View and text box) and an optional third content card. The Layouts section is where you determine how that content will display on the page by setting the width and height of each card.

The screenshot shows the 'Layouts' configuration page with the following settings:

- Card 1 (Cube View):** Width: 1145, Height: *
- Card 2 (Text Box):** Text Box Height: *
- Card 3 (Content):** Visibility: Hide, Height: 300. Manage Content buttons: Link, Add.

The visual diagram on the right shows three black boxes: a tall box labeled '1', a smaller box labeled '2' to its right, and another smaller box labeled '3' below box '2'.

For **Height** and **Width** settings, enter a size in pixels. Enter * for the element to dynamically adjust its size based on the size of its container.

Card 3 (Content): If you're adding more content to this page:

- Set the Visibility to Show
- Manage Content: This is where you configure the card
 - **Link:** use this to display an existing Dashboard
 - **Add:** use this to insert and configure another Content Block. Once you have added the desired block, click Edit to navigate to its configuration settings. Click Clear to remove it.
- Determine the height in relation to the card content and the rest of the page. You may need to update the sizes of all cards for ideal display.

Appendix 1

Workspace Requirements and Considerations

Workspace objects:

- Cube Views
- Parameters
- Data Management Sequences/Steps
- Business Rule Assemblies:
 - XFBRStrings
 - Dashboard Extenders

Requirements for all Workspaces and objects you intend to use in Genesis:

Requirements	Options
Shareable Workspaces:	Ensure the Is Shareable Workspace setting is set to True on each Workspace; this excludes the Default Workspace.
Share Workspaces with Genesis Workspace:	Locate the Genesis Workspace and ensure each Workspace is assigned to the Shared Workspace Names property.
Object Storage:	Same Shareable Workspace <ul style="list-style-type: none"> • All objects stored together Default Workspace <ul style="list-style-type: none"> • All objects stored in the Default Workspace Default/Shareable Workspace <ul style="list-style-type: none"> • Universal objects stored in Default Workspace (Parameters, Data Management Sequences) • Cube View stored in separate shareable Workspace
Row/Column Sharing Cube View Storage:	<ul style="list-style-type: none"> • Default Workspace • Shareable Workspace Must be assigned to Genesis via Shared Workspace setting Stored in the same shareable Workspace
Include the Workspace Name when calling objects:	<ul style="list-style-type: none"> • Calculation Details: Parameters, Assemblies or Data Management Sequences Ex: {WSName.MySequenceName}{NameValuePair=WSName.ParamName}

Release Notes

Version 2.0.0

This major release introduces block upgradeability further strengthening content development and management. This capability is foundational to how Genesis content will evolve going forward.

Important Notes

Upgradeability is only available for blocks running version 2.0.0 or later. Existing blocks on version 1.x must be migrated to 2.0.0 first. This is a one-time process. See *Content Block Upgrade Path* below for migration details.

Enhancements

Block Upgradeability

This block has been updated to fully support the new upgradeability framework introduced in Genesis 2.0.0. This ensures that the block can evolve over time without requiring pages to be rebuilt or configurations to be recreated.

The following capabilities are now available for all blocks at version 2.0.0 and above:

1. **Version Awareness**

Blocks now report their version and indicate when a newer version is available. Administrators can see at a glance which pages are running older versions.

2. **Compatibility Validation**

Each block now includes version-to-version compatibility rules. During an upgrade, Genesis automatically checks whether content and configuration can be safely migrated.

3. **Selective Upgrades**

Identify and select specific blocks to upgrade on specific pages.

4. **Configuration and Content Preservation**

Upgrades carry forward existing filter settings, parameters, and layout configuration. Pages do not need to be rebuilt.

Important Note: This feature is not retroactive. Blocks on version 1.x do not include the upgradeability framework. They must be migrated to 2.0.0 first. See *Content Block Upgrade Path* below for migration details.

Platform and Genesis Compatibility

Genesis Version Requirement

Blocks at version 2.0.0 are only compatible with Genesis 2.0 or later.

Do not inject a 2.0.0 block into a Genesis 1.x Instance. The block will not function, and the upgradeability features will not be available.

Minimum Platform Version	9.0.0
Minimum Genesis Version	2.0.0

Content Block Upgrade Path

One-time Migration: Moving an Existing Block to Version 2.0

This process applies only to blocks currently on version 1.x. Follow these steps once to bring your existing block up to version 2.0.

After completing this migration, future upgrades can be performed directly within Genesis and will not require content re-building.

1. Create a new page and set its visibility to Hidden.
2. Inject and configure the 2.0.0 version of the block on this page.
3. Verify functionality to ensure the block behaves as expected.
4. Hide the original page and make the new page visible.
5. After end-user confirmation, delete the original block and page.

Why is this still required for 2.0?

Block upgradeability is functionality built into the 2.0.0 block itself. Blocks on version 1.x do not include this functionality and therefore cannot be upgraded in place. The steps above are the final time you will need to follow this process for this block.

Upgrading from 2.0.0 to Future Versions

Once a block is running version 2.0.0 or later, upgrades are handled directly within Genesis. No page rebuilding or reconfiguration is required.

Refer to the [Genesis Guide](#) for more details on the block upgrade process.

Version 1.2.0

This release focuses on automatic parameter detection enhancements and greater control over filter behavior.

Enhancements

- **Support for Member Dialog Parameters**
A Cube View's Member Dialog parameters are now automatically detected and added to the block's Filter Settings.
- **Support for Nested Parameters in Row/Column Definitions**
Nested Parameters from a cube view's row/column definitions are now automatically detected and added to the block's Filter Settings.
 - Nested parameters: parameter within a parameter.
- **Support for One Level of Parameters in Row/Column Sharing**
One level of Parameters from a Cube View 's row/column sharing sets are now automatically detected and added to the block's Filter Settings.
 - Level 0: A parameter exists in the Cube View's row/column sharing property.
 - Level 1: One of the selected row/column sets also contains a parameter in its own row/column definition. Traditionally, after the initial parameter selection, the user is prompted again to provide values for these additional parameters before the Cube View fully renders.
- **New Filter Type 'Hidden'**
Parameter will not appear in the filter toolbar. The Override Value is used as the parameter value instead. If no Override Value, the user is prompted on page launch to select a value.
- **New Filter Property 'Override Value'**
Assign a fixed value to the parameter, replacing any parameter options. When entered, this value is used for the POV, and users cannot change it. Enter the value directly or click the Select Member button to choose from a lookup dialog.
- **New Filter Property 'Default Value'**
Set the initial value when the parameter runs for the first time. Enter the value directly or click the Select Member button to choose from a lookup dialog.
- **Manually Add/Remove Parameters**
Manage filter parameters that are not automatically synchronized with the Cube View. Add parameters that were introduced to the Cube View after the block was created. Remove parameters that no longer exist in the Cube View or are no

longer needed in the block.

- **Redetect Parameters**

Performs a complete rescan of the assigned Cube View and replaces the current parameter list in Filter Settings with the parameters detected in that Cube View.

Version 1.1.0

The theme of this minor release is big fixes.

Fixed Issues

- **Cube View Parameter Handling**

When using NavLink parameters in the comments member filter (e.g., ~|ScenarioNavLink|~), Cube View parameters remain unset until the user interacts with the grid. This causes errors when attempting to view Cube View comments or request AI summaries. A validation check now ensures that each parameter has a value before processing. If a value is missing, the system automatically applies the None member to the member filter.

- **Edit Button Hidden After Injecting a Block into a Card**

When another block was injected into the Cube View AI Comments block, the Edit button did not appear, preventing users from configuring the block. The issue has been resolved, and the Edit button now displays correctly after block injection.

Platform and Genesis Compatibility

- Minimum Platform Version: 9.0.0
- Minimum Genesis Version: 1.0.0

Version 1.0.0

This is the initial release of the block.

Platform and Genesis Compatibility

- Minimum Platform Version: 9.0.0
- Minimum Genesis Version: 1.0.0