Cube View Advanced Embedded Content Block Guide

Block Version: 2.0.0
Minimum Genesis Version: 2.0.0

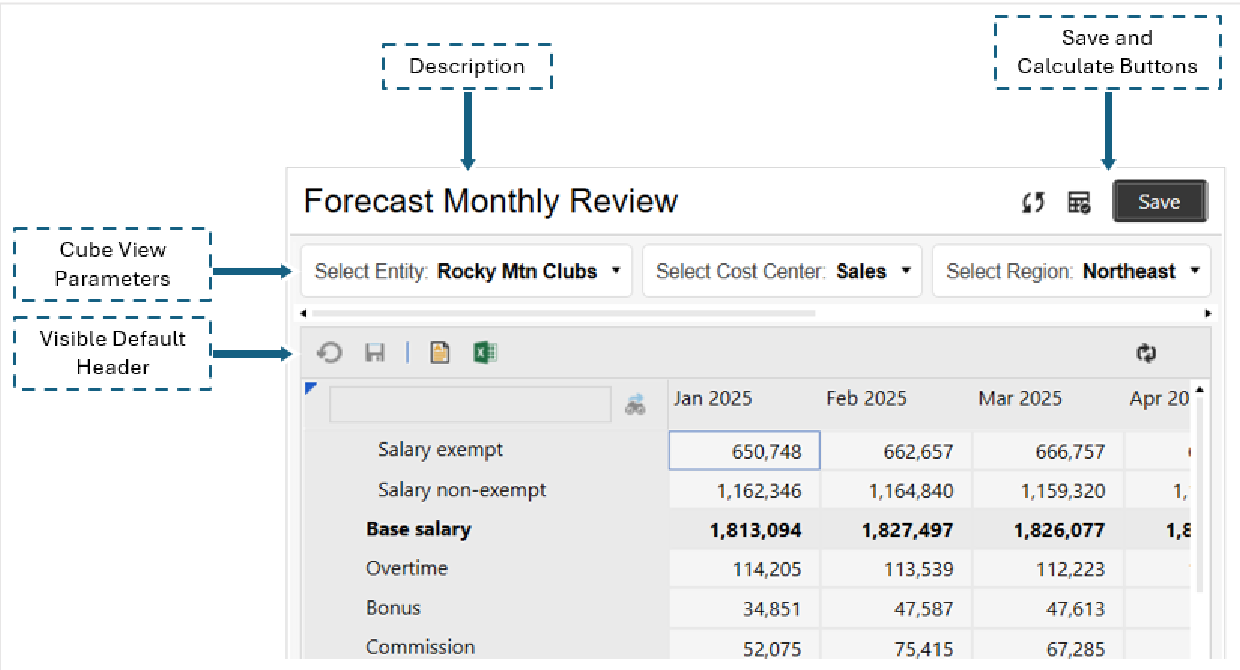
Minimum Platform Version: 9.0.0

Jump to Release Notes

Table of Contents

Overview

The Cube View Advanced Embedded content block is used to display a Cube View with optional filters in a layout block. **This block is very similar to Cube View Advanced but it is designed to be used in a layout block** Filters are dynamically generated for each Parameter in the Cube View which generate Combo Boxes and/or Member selectors for dynamic Member selection. Additionally, buttons can be configured to Save upon data entry and/or run calculations and Data Management Sequences.



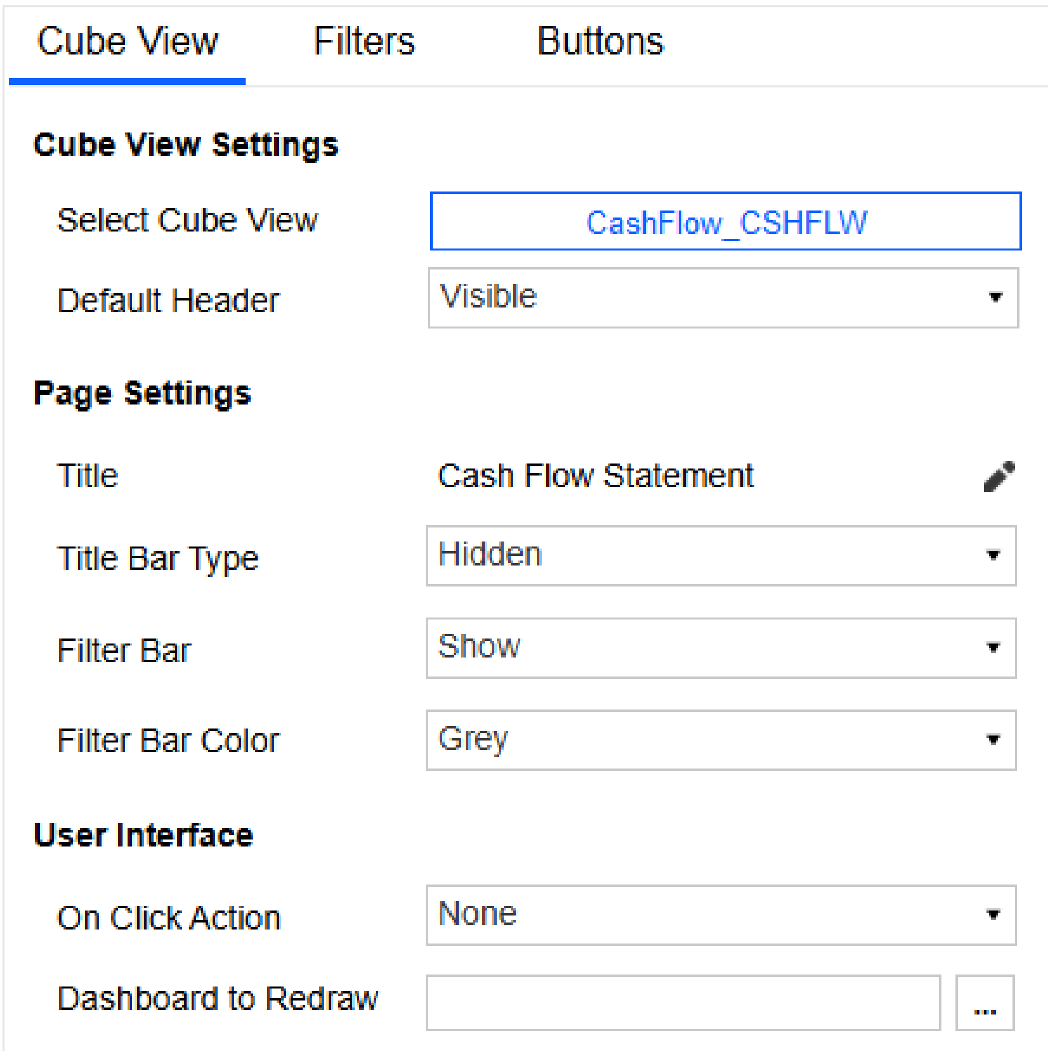
Use Cases

- Used with other blocks within a layout block
- Data Entry Forms
- Data Consumption
- Workflow Activities
  - This includes calculations, consolidations, and Complete/Revert Workflow.
- **Standard Data Analysis**:
  - This involves dynamic Member Selections via Cube View's Parameters, including:
    - Drill down
    - Linked Cube Views and Dashboards
    - Export to Excel/Spreadsheet.
    - Open as a Report

Designer Page (Configuration)

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

Cube View Settings



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.

> **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.

Default Header

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

Title

If your Cube View includes a description the Title will automatically populate based on that description. You can also enter your own title.

Title Bar Type

The Title Bar has 3 modes:

- Hidden: Hides the Title. Useful for when you want the tool bar just to have filters.
- Large: Includes a larger Title and Buttons
- Slim: Is a smaller sleeker title for use when embedding this block in a layout block.

Filter Bar

You can hide the Filter Bar entirely, which is helpful when parameters and filters are managed externally in an embedded layout.

Filter Bar Color

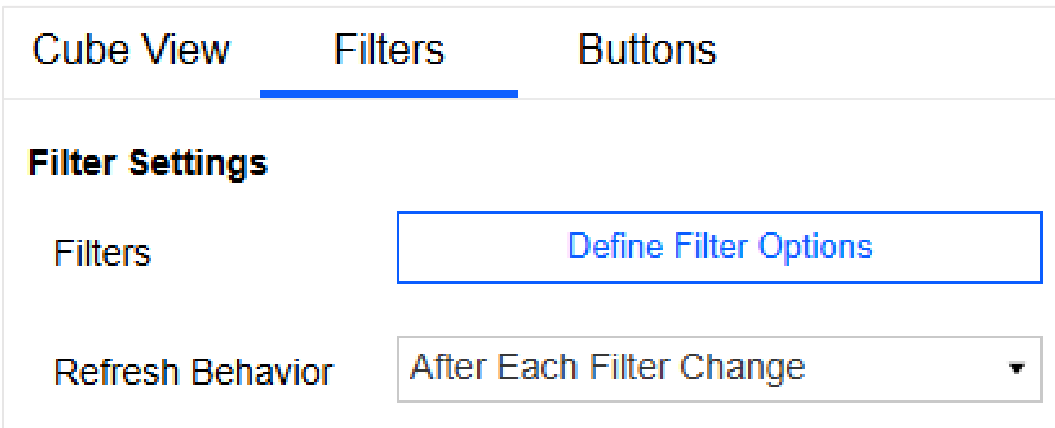
Choose one of the two options. Grey is the default color because it is the background color of all layout blocks that this Embedded block is designed to work with.

On Click Action / Dashboard to Redraw

Drives actions when the end user clicks a cell. The dashboard to redraw allows you to select what other dashboards you want to refresh on click.

> **NOTE:** When using a layouts block, refresh the card or row dashboard rather than an embedded block directly. For multiple toolbars, list all toolbars for refresh to keep parameters and filters synchronized.

Filter Settings



Filters

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

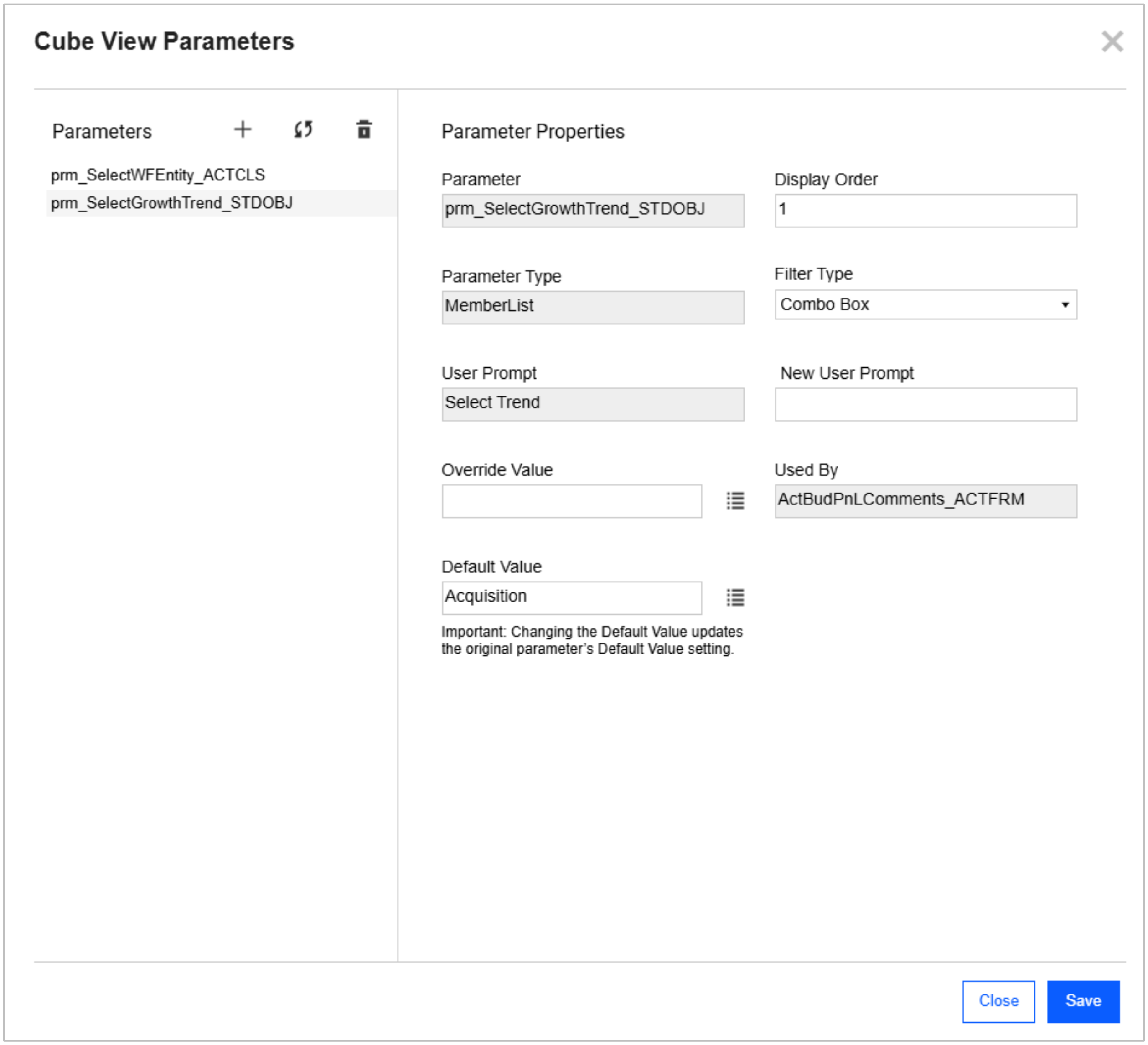
Refresh

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 Options

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



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.

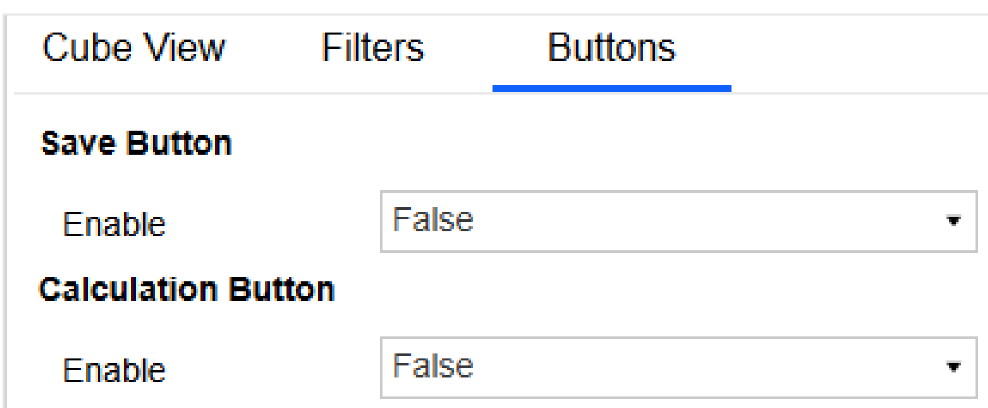
- **Default Value**
  Sets the initial value when the parameter runs for the first time. Enter the value directly or click the **Select Member** button to choose from a lookup dialog.

> **NOTE: Important Behavior for saving 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.

> **NOTE: Important Behavior for Default Value and use of Parameters**
> If you're using Parameters for row/column sharing, ensure the Parameter has a Default Value to avoid error upon rendering.
>
> See *Appendix 1* for more details on Workspace requirements and considerations.

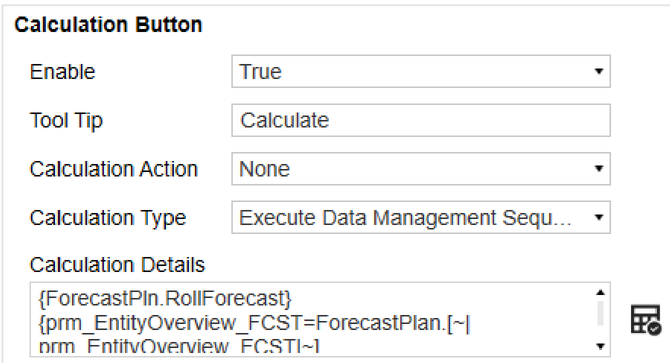
Buttons Settings

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



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.



> **NOTE:** The syntax for Parameters and Substitution Variables is different when used in Genesis:
>
> **Parameter Syntax:** ~|ParameterName|~
> **Substitution Variable Syntax:** ~SubVariableName~

> **NOTE:** Specify the Workspace Name in the calculation arguments where applicable:
> - WorkspaceName.DataManagementSequenceName
> - WorkspaceName.ParameterName

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 • All objects stored together Default Workspace • All objects stored in the Default Workspace Default/Shareable Workspace • Universal objects stored in Default Workspace (Parameters, Data Management Sequences) • Cube View stored in separate shareable Workspace
Row/Column Sharing Cube View Storage:	• 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:	• 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.

Fixed Issues

- Literal Parameters no longer display in Filter Settings.

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 1.1.0

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.

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