onestream

Modern Browser Experience Guide

8.5.0 Release

Table of Contents

About This Guide

This guide introduces the modern browser experience and highlights the end user benefits and functional differences of using OneStream's web-accessible interface. This guide is intended to:

- Familiarize OneStream end users, power users, administrators, partners, and consultants with browser navigational features and UI (user interface) elements.

- Provide prerequisites and directions for implementation of a browser experience in a customer environment during various stages of development.

- Provide high-level information about supported features and capabilities, general usage patterns, end-user flows, and best practices.

- Provide high-level information about differences between the browser and Windows Application.

Modern Browser Experience Overview

Welcome to the modern browser experience, OneStream's latest, web-based offering. The modern browser experience works alongside the existing OneStream Windows application, giving users more flexibility by providing them with a way to access their data from anywhere using a wide range of devices. OneStream in the browser delivers a streamlined, modernized experience that strives to provide comprehensive coverage of end user features. With a strong foundation developed utilizing contemporary concepts and technologies, the modern browser experience introduces a suite of innovative designs and improved solutions.

All MarketPlace Solutions should render in the Modern Browser Experience. To find out which solutions are officially supported in the Modern Browser Experience, refer to solution documentation. Additionally, functionality provided through the System and Application Tabs is not yet available through the browser.

The following sections highlight benefits, explore differences, and provide current limitations to help customers and partners evaluate the browser experience for their business needs.

For requirements, see Hardware and Software requirements.

For more information on the installation process, see the Installation Guide.

For information on prerequisites for self-hosted environments, refer to the Legacy Authentication section in the Upgrade Guide.

Modern Browser Experience Updates 8.5

8.5.0 introduces a variety of new features across the platform, which compliment earlier delivered functionality to enable a wider range of end users needs.

Functional Capabilities

Feature	Description
Smart Links	Smart Links combine navigation to a dashboard and configurable parameters that define specific views of components and data. Once generated, a Smart Link URL can be leveraged to navigate directly to a dashboard in browser while bypassing the steps needed to configure specified pages or components.
Environment Name - Color Settings	The Environment Name displayed in the Browser Experience header has been updated to reflect the selected color in System Configuration.

Workflow

Feature	Description

Select All Journals	Journal select functions have been updated to ensure all items are selected in large lists.
Bulk Journal Processing	When you select multiple journals with the same status, you can perform processing actions on all of them simultaneously.
Saving Large Journals	Journal save functions have been optimized to prevent errors when saving journals with 1000+ line items.
Journal Line Item Tabbing	You can now tab across cells in journal line items.

Dashboard

Feature	Description
Waterfall Charts	The Advanced Charts component has been updated this release to support the Waterfall chart type.
Spreadsheet	The ability to submit changes to data in a sheet or workbook through Cube View Connections and Table View Definitions has been implemented in Spreadsheet. The XFSetFXRate Set Cell function has also been implemented this release.

Embedded Web Content	Updates to the Web Content component allow users to render content in a dashboard instead of opening link in new tab.
Filter Editor	The Filter Editor component gives users a visual interface for modifying and applying advanced logic to filters.

Access the Browser Experience

The browser URL structure uses a company subdomain (for example, https://cloudsitename.onestreamcloud.com). OneStream recommends using Chrome, Edge, or Safari to access the modern browser experience.

Authentication

For OneStream hosted environments, OneStream IdentityServer (OIS) technology enables browser users to authenticate through a browser and enables administrators to set up online environments to use native IDs or an identity provider (IdP). For more information on authentication using OIS, see the [Identity and Access Management Guide.](#)

The setup and configuration of the browser experience may vary based on how your environments are hosted. For self-hosted environments, users can authenticate into the browser using their existing authentication method. Additional instructions for self-hosted authentication can be found in the [Installation Guide.](#)

General Navigation

Learn how to navigate browser experience. This section explores foundational capabilities of the browser experience, general features and highlights notable differences between the browser and Windows Application.

Intuitive User Experience

Log in

The modern browser experience is accessed through a web browser instead of using a Windows shortcut or ClickOnce to open a local instance of OneStream. You can use bookmarked URLs to navigate to the browser. Once successfully authenticated, you can select an application from the drop-down menu and click the **LOG IN** button to access OneStream.

The browser environment is accessed through a custom URL. For example, https://cloudsitename.onestreamcloud.com/. To start the authentication process, input a Username and select Next. Once authenticated, the browser will display a OneStream Home Page or, if multiple Applications are available, an Application Selector.
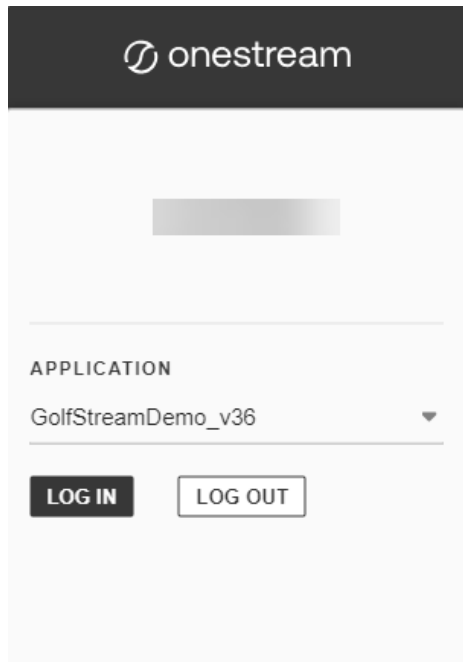
Log out

1. In the **User Menu**, select **Log Out,** or
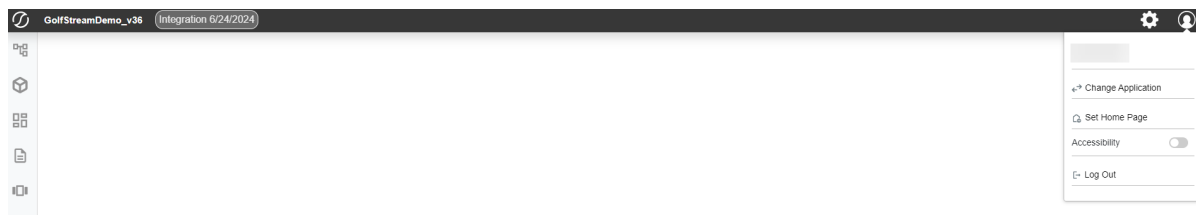
2. On the **Application Selection** page, click the **LOG OUT** button, or
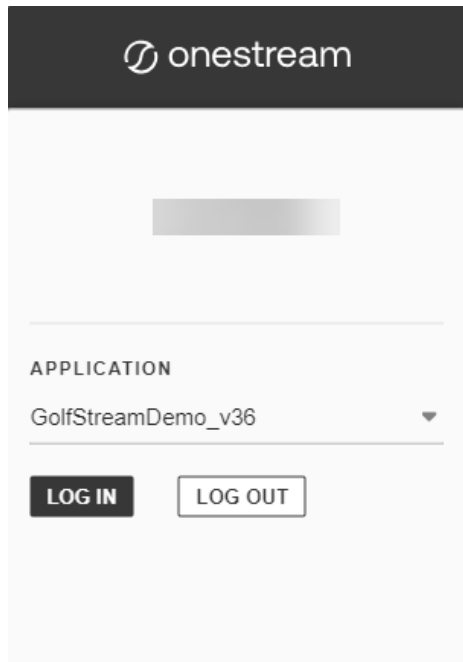
Logging off will not log off other open OneStream instances, such as the Windows Client or Excel Add-In. If using single sign-on, the user will not be logged out of your single-sign on session.

Change Applications

To change applications, select **Change Applications** from the **User Menu**. The Application Selector will be displayed with available applications.

Header

The browser experience has a header and a OnePlace panel that persist throughout the user experience. Similar to the Windows Application, the header has a Home button embedded in the OneStream logo and displays the Application Name. Other header functionality has been moved to two menus in the top right corner of the screen.

- **Gear Icon** – Application Name, Global POV, Version Number, Help
Documentation, Task Activity, File Explorer

- **User Icon** – User's Name, Change Application, Set/Clear Home Page, Log
Out

The OnePlace Panel can be fully collapsed, opened to display Navigation controls, or expanded as an overlay when working with any of the menu areas.

> **NOTE:** The OnePlace panel may be fully hidden for certain users using security settings.
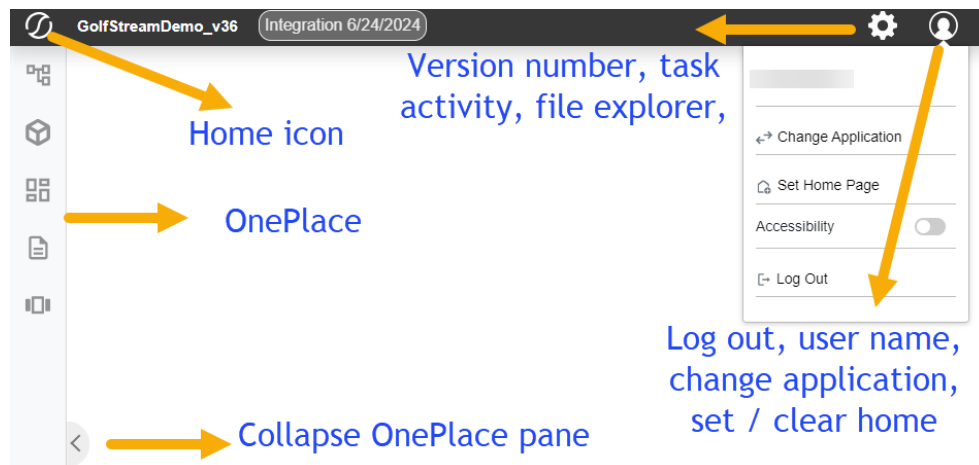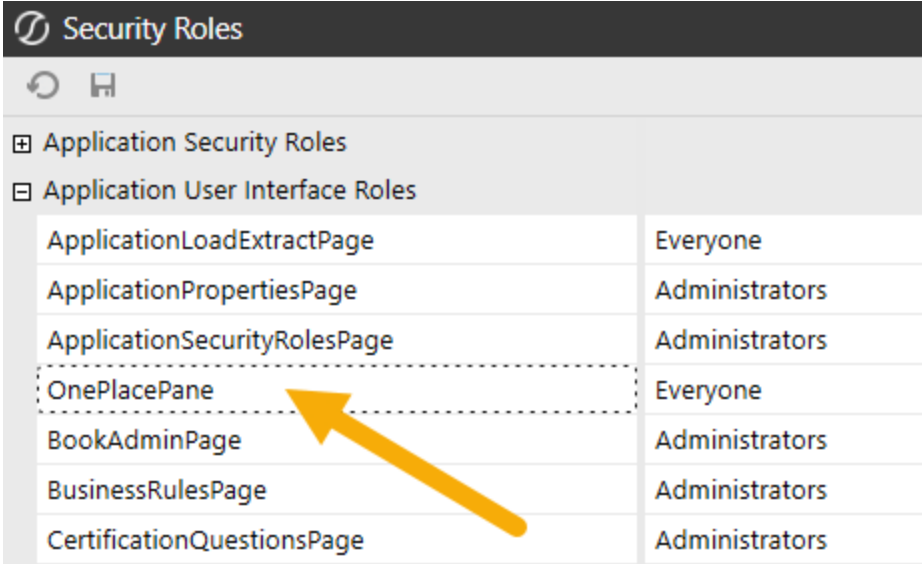
OnePlace Panel

Hide OnePlace Panel

The OnePlacePane security role can be used to hide the Navigation panel. This configuration has been added under Security Roles to enable or disable end user access

to the OnePlace navigation panel.

Navigate OnePlace Panel Sections

In the expanded OnePlace panel, there is a drop-down menu at the top of each area to navigate between them.
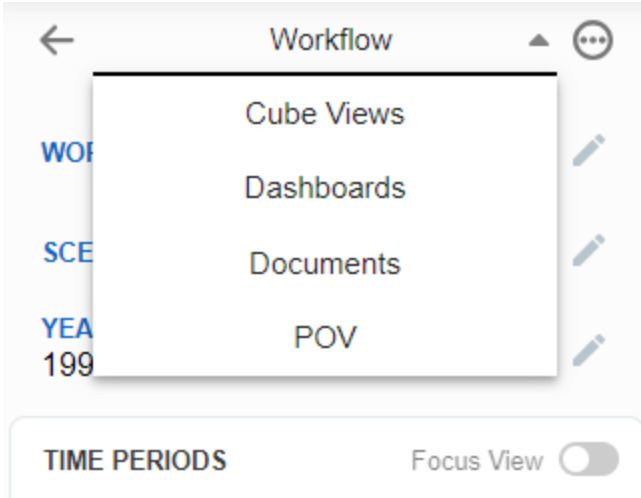
File Explorer

File Explorer is used to manage all OneStream files and saved POVs. All file actions are available in the toolbar, including create, edit, delete, upload, download, and apply POV.
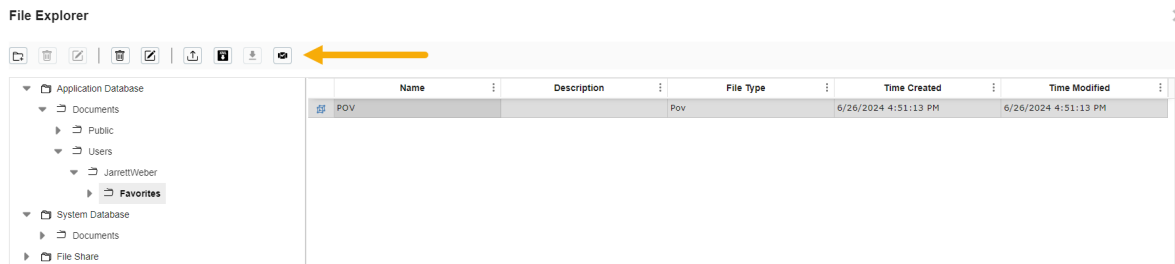
Grids

While the browser grids offer the same functionality as the Windows Client, some differences were added to improve the user experience. Many of the grids have virtual scrolling instead of paging to handle large datasets. In addition to the Sorting, Filtering, and Column controls found in the Windows Client, the browser experience has added a column Lock feature, which allows users to lock in place any column and the columns to its left, while being able to scroll the remaining columns horizontally. In the header, the browser has added an AutoFit button, which will automatically fit all the column widths.

Many of the grids will also have a context menu button ⊙ in the left corner of the grid header with additional functionality. See Appendix D for a list of supported Number Formats.

Amount	⋮	HoustonEntities	⋮
86,400,000.0	↑	Sort Ascending	
28,704,154.5			
3,879,484.1	↓	Sort Descending	
3,561,084.4			
11,839,604.0	▌▌▌	Columns	⌄
1,600.0	▼	Filter	⌄
3,200.0	▤	Set column position	⌃
1,050,481.8			
3,740,384.4	🔒	Lock	
2,121,928.9			
-221,962.4	🔓	Unlock	
2,546,879.1	◄	Move Previous	
495.6	▶	Move Next	

Dashboards / **GridViews_Dashboard**

⊙⊙⊙

⤢	AutoFit		
Dr		it here to group	
↺	Reset State	⋮	**Registe**
☐			
☐	⤶ Reset All States	T...	
☐	⬈ Export	▶	Excel
☐			CSV
☐			

Anchored Dialogs

Many of the dialog boxes display in a slide-out panel anchored to the right side of the page instead of the floating dialogs used in the Windows Client. This paradigm offers improved visibility and provides consistency for users accessing the browser from various devices. Slide-out panels can be expanded or constricted to suit user preference by dragging the left edge horizontally across the page.
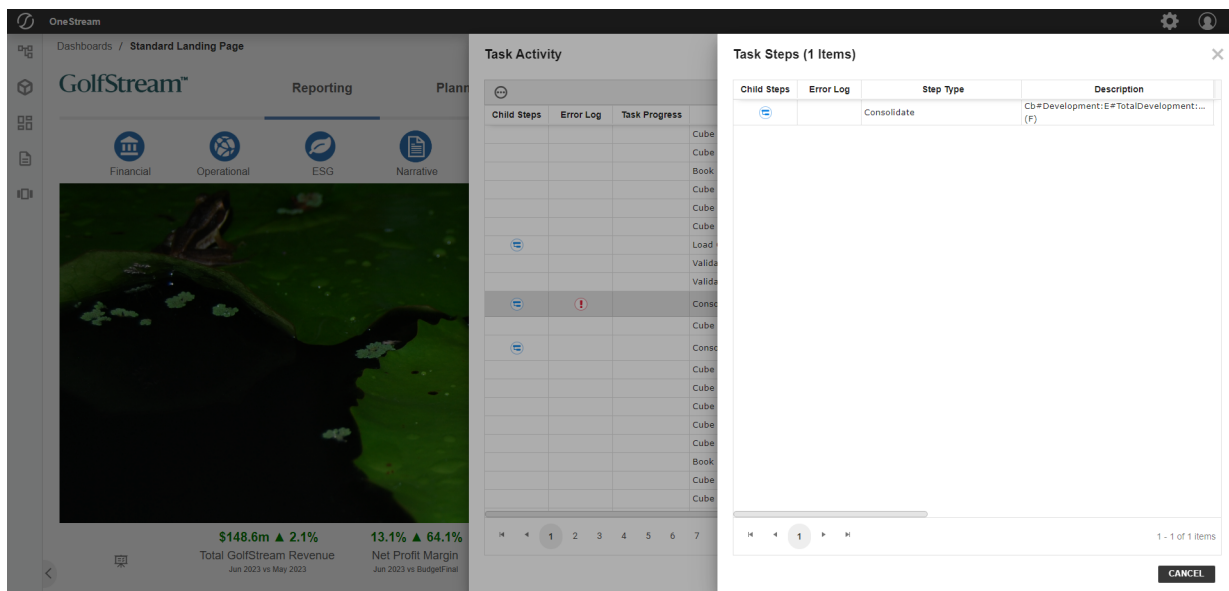
This anchored dialog box has been extended to dashboards that appear in dialog boxes. There is a larger default width compared to the Windows Client, but the width can be altered with the dialog box display format configurations. Since these dialog boxes are always the full height of the screen, the height configuration for dashboards in dialog boxes are ignored.

Smart Links

Smart Links are designed to make it easy to share dashboards. Built with a combination of security-aware features and declarative parameters, these links can open dashboards to specific pages or mimic a series of clicks.

Smart Links are generated and stored for a set period using a Business Rule. To access a shared link, users will simply click a hyperlink or paste the URL into a browser. On access, the Smart Link will:

- Verify if the user is logged in (if not, remain active through the login process)

- Check if the user has access to the shared dashboard

- Verify the dashboard can be rendered on the user's device

- Supply the dashboard with any declared parameters

- Prompt the user for any undeclared parameters

- Render the dashboard

Example Smart Link structure: https://example.onestream.com/share/ E1C52337EEE9430EAAF52D6A5846109B/DashboardProfile/Dashboard

To get started, review the code snippets in Appendix E.

Workflow

Workflow on the browser delivers the same functionality offered in the Windows Application with a few notable exceptions mentioned below. Some tactical user experience improvements have been made, while primarily focusing on delivering the functionality of the workflow steps.

Navigation

Similar to the Windows Application, Workflows are accessed through the OnePlace panel. Users are able to modify the Workflow POV by selecting the edit icon next to each dimension and navigate between Workflow Units under Time Periods. It is important to note a few differences between browser and the Windows Application:

- The Workflow POV Edit menu is displayed as an overlay menu, instead of a pop-up dialog. Users are able to update Workflow, Scenario, and Year using the available tabs. A Save button at the top of the section finalizes any changes.

- The new **Focus View** function allows users to view only the selected time period. In the example below, a user chose to view only June to simplify navigation. To return to the full list, simply disable the **Focus View** toggle.
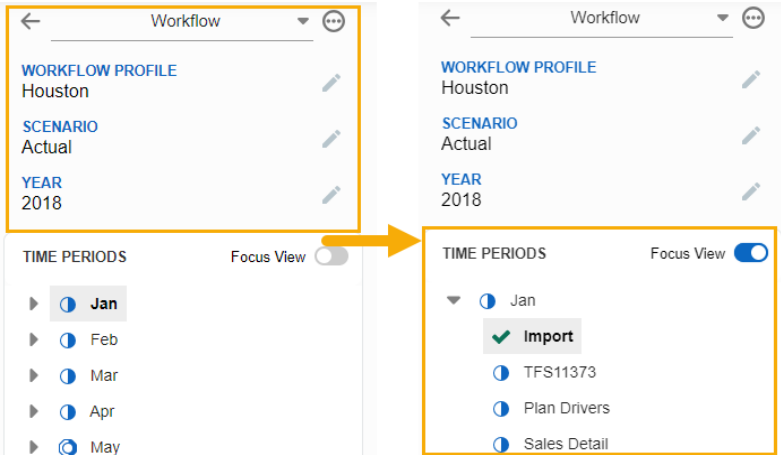
Status Icons have been updated, and in some cases, combined to make a more intuitive user experience.

Label	Windows Application	Browser UX
Complete	🟢	✔
Disabled	⚪	⊘
Error	🔴	!
In Progress	🔵	◑
Input Form Started	🔵	◑
Last Step Complete	🟢	◑
Last Step Disabled	⚪	⊘
Last Step Error	🔴	!
Last Step in Progress	🔵	◑
Optional Workflow Complete	✓	✓
Completed and Closed		◐
Completed and Locked	🔒	🔒
Incomplete Closed		◐
Incomplete Locked	🔓	🔓

Workflow Breadcrumbs

You can navigate to a specific month, time period, scenario, or workflow step of your choosing from the workflow navigation feature. Selecting an option will take you to the corresponding step. A new dropdown in the Workflow page title bar allows users to quickly toggle between Workflow Steps in a Time Period.
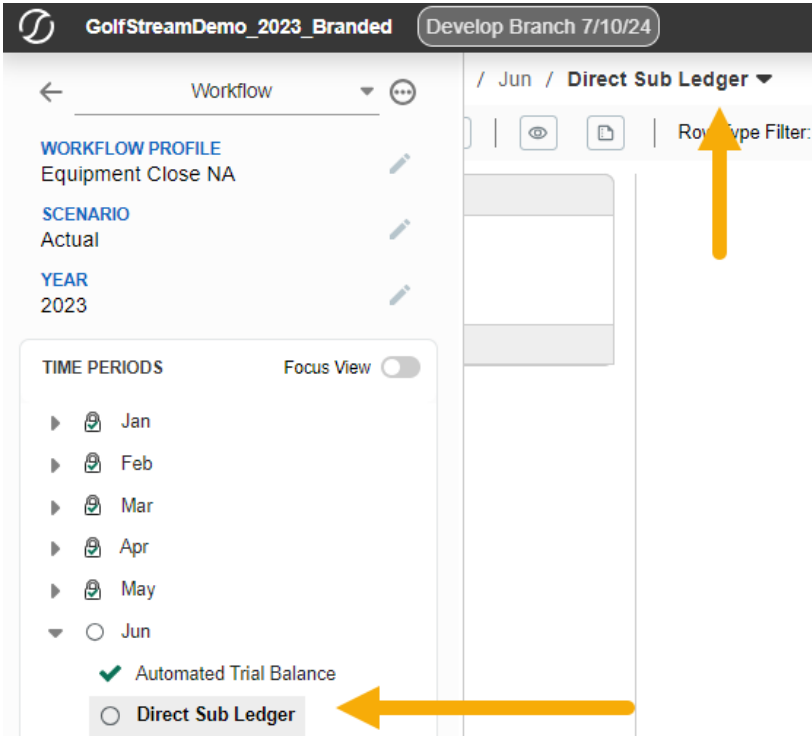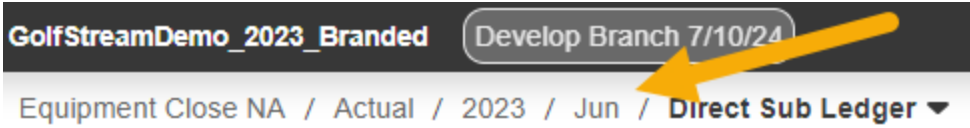
Supported Workflow Functionality

Workflow Steps

- Blend
- Central Form Input
- Central Import
- Central Journal Input
- Certify
- Confirm
- Direct
- Form

- Import
- Journal Input
- Load
- Pre-Process
- Process
- Validate
- Workspace

General Workflow Functionality

- Drill Back
- Drill Down
- Analysis Area
- Application Configurations
- Intercompany Matching
- Multi-Period Processing

- Time Range Periods
- View Transformation Rules
- View Processing Log
- View Source Document
- Time Period Actions
- Dependent Actions

Notable Differences from Windows Application

- Workflow Page Styling – Workflow Title, Workflow Steps, Grids, Status Area, and Analysis Area.
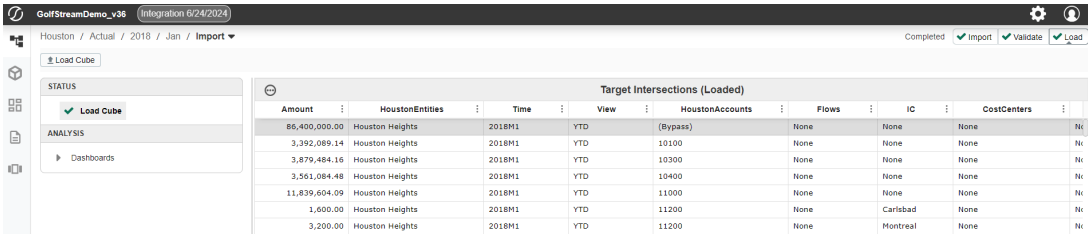
- Status Icons – Consistent with the OnePlace panel, the Workflow status icons have been simplified in the browser to make a more intuitive user experience.
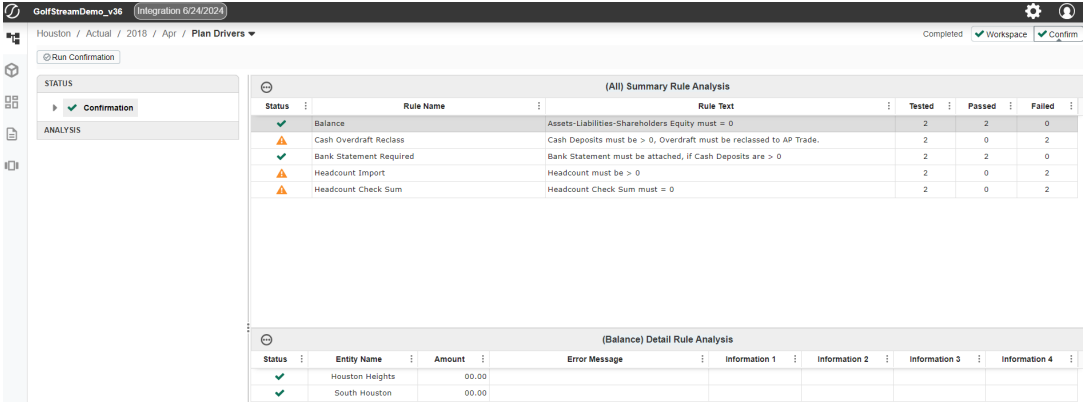
- Workflow Progress Buttons – Now displayed on the right side of the screen, the progress buttons have been updated with new icons and borders to improve visibility.
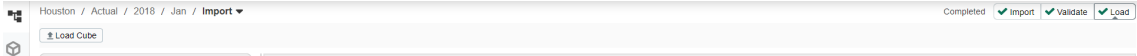
- Drag and Drop File Import – Files can be imported by selecting from your device or using a new drag and drop feature.

- Context Menus for Grids – All Workflow Grid functions have been moved from right-click menus to a context menu in the header of each grid.
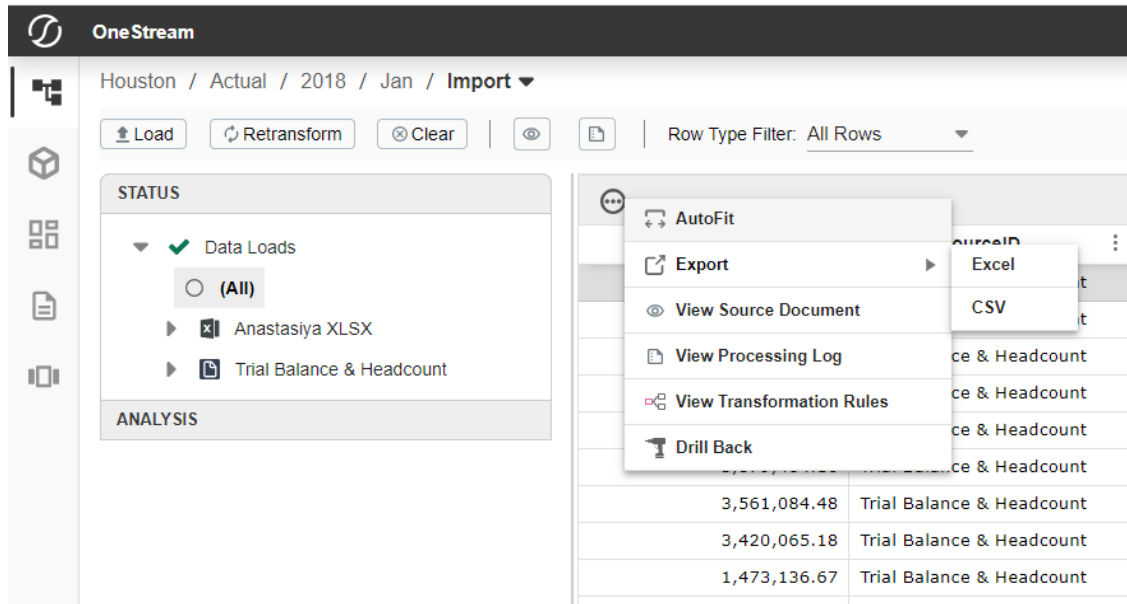
- Rename Journal – You can rename journals by editing the Journal Name property in the journal header, instead of a rename button and dialog.

Notable Workflow Functionality Gaps

Workflow Steps

- Manage Workflow

General Workflow Functionality

- Edit Transformation Rules

Navigating Workflow Step Gaps

Helpful tips have been added to the browser experience to identify unsupported workflow steps for both dashboard designers and end users. *Coming Soon* icons indicate a workflow step is not currently supported.
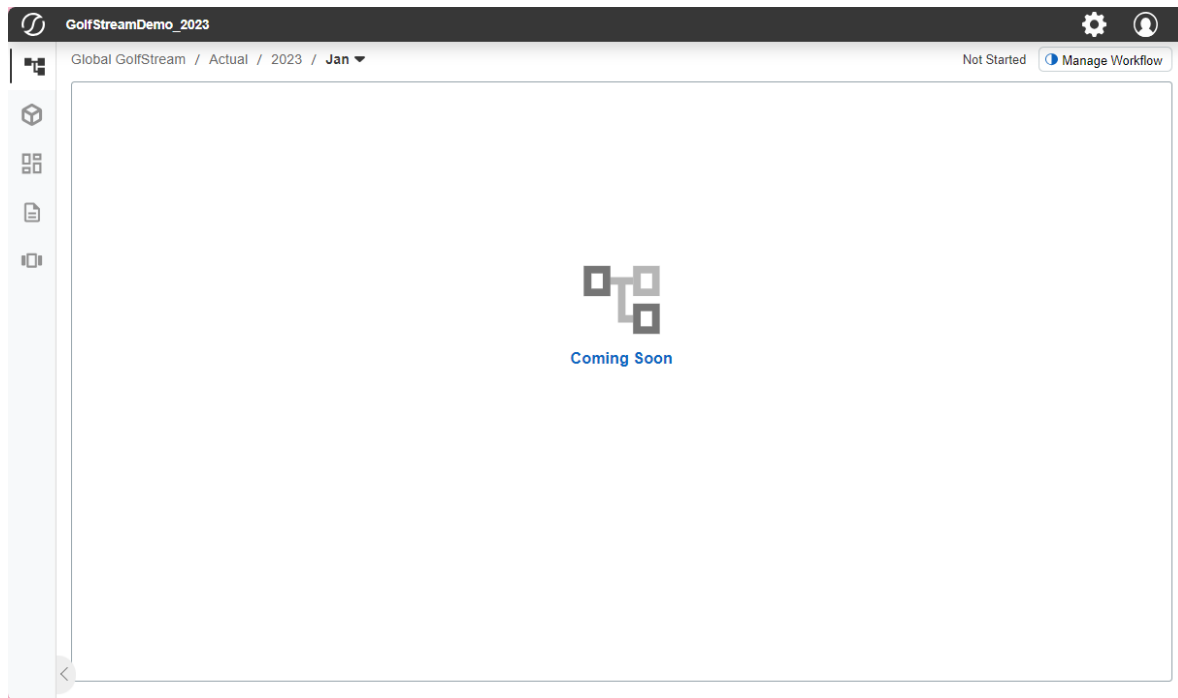
Even if all steps are supported in each workflow, thorough testing is still recommended to ensure processes are working as expected before production use.

Cube Views

Cube views are flexible reporting artifacts that enable cube data to be presented and edited in various ways by utilizing security, configuration, and formatting properties, member formulas, and calculations.

Nearly all cube view features provided in the Windows Application are available to users of the Modern Browser Experience. For cube views that utilize features or functionality that have not yet been implemented, or in cases when variations arise due to technical differences between desktop and web-based applications, there may be configuration changes or workarounds that can be leveraged until functional gaps can be closed.

Supported Cube View Functionality

Presentation

- Cube View and Export Format Configuration Properties

- Cell Data Types

- Cell Status Background Colors

- Row and Column Sharing

- Nested and Expandable Column and Row Headers

Toolbar Functions

- Context Menu Dropdown

- Save and Revert Changes

- Data Refresh

- Parameter Selection

Cube View and Cell-Level Context Menu Functions

- Calculate, Translate, Consolidate

- Data Attachments

- Spreading

- Cell Detail

- Cell POV Information

- Data Unit Statistics

- Cell Status

- Drill Down

- Row Suppression

- Export

- Cube View and Dashboard Navigation Links

Cube View Navigation

The General Visibility setting in Cube View profiles will be applicable for cube views and the Windows Client (OnePlace, Workflow, and Dashboards). The Client/Device Visibility setting in Dashboard profiles will also affect access to embedded cube views or data explorer components in Dashboards. See Dashboards.

Cube View Drill Down

The Drill Down interface has been moved from a page tab into a slide-out panel. You can drill down into cube data from a cell intersection by right-clicking the cell and selecting **Drill Down** from the cube view context menu. The cube view drill down interface opens in a slide-out panel, which can be resized horizontally.

Currently, only the **Open** in Panel option is supported. The **Open** in **New Tab** option will be considered in a future release.

A context menu was also added to the toolbar on the Drill Down page to consolidate some functionality in a way that better supports different screen sizes and devices.

A hidden context menu for drill down has been implemented to consolidate key drill down functions, such as Load Results for Imported Cell and Audit History for Forms or Adjustment Cells.

User Experience Differentiators

Row virtualization has been implemented to efficiently retrieve and render large datasets for cube views. Row virtualization retrieves data to display as the user scrolls vertically through cube view pages, improving performance of client-side data rendering and enabling a more seamless data review process. Row virtualization occurs in defined Cube View page boundaries.

Notable Functional Gaps

- Allocation
- Autofit

> **NOTE:** The modern browser experience uses column widths specified at design time, or a default width of 160 pixels when no specific width is defined for a column, or will autofit the column when "Auto" is used to set the width.

Dashboards

Browser dashboards are expanding functional coverage with the goal of offering equivalent use of layouts and components available in the Windows Client. Currently, the most commonly used features are available. For dashboards that contain elements that have not been delivered or potential gaps that can arise out of unique design techniques, there are typically substitutions, configuration changes, or workarounds that can be implemented until subsequent releases close the functional gap.

While browser capabilities move toward comprehensive end user dashboard functionality and matching rendering between the Windows Client and the browser experience, the underlying technologies between the clients will likely prevent perfect replication of dashboards in both clients. Additionally, since the user interface styling is different between the clients, dashboards designed for one client may look out of place when rendered in another.

If a preexisting dashboard contains only supported layouts and components, it can be enabled for use with a single configuration. However, for an optimized user experience on the browser or on mobile devices, designers may consider modifying the dashboard or creating new client/device specific experiences.

Enabling Dashboard Access

To ensure dashboards are only rendered within the client (Windows Application or browser) or device (PC, tablet, or phone), a new configuration has been added to the Dashboard Profile called Client/Device Visibility.

The Client/Device Visibility configuration enables dashboard designers to control which dashboard profiles, and the dashboards they contain, can be accessed and viewed on one or more clients and devices. Enabling this configuration will display the dashboard in the OnePlace Dashboard list of the respective client/device selection.

Preexisting and newly created dashboards will be set to the Windows Client only by default. Dashboards are not initially visible. Dashboards always render in Workflows.

Supported Dashboard Functionality

Dashboard Layouts

- Grid
- Uniform
- Tabs

- Horizontal Stack Panel
- Vertical Stack Panel

Dashboard Components

- Advanced Charts
- BI Viewer
- Book Viewer
- Button
- Check Box
- Combo Box
- Cube View
- Dashboard Filter Editor
- Data Explorer
- Data Explorer Report
- Date Selector
- Embedded Dashboard
- File Viewer
- Filter Editor
- Gantt View
- Grid View
- Image
- Label
- Large Data Pivot Grid
- List Box
- Logo
- Pivot Grid
- Radio Button Group
- Report
- SQL Table Editor
- Supplied Parameter
- Text Box
- Text Editor
- Text Viewer
- Tree View
- Web Content

General Dashboard Functionality

- Actions
- Extensible Documents
- User Supplied Parameters

Notable Differences from Windows Application

Dashboard Components

BI Viewer

You may observe style differences, like muted borders and headers, or alternative style visualizations.
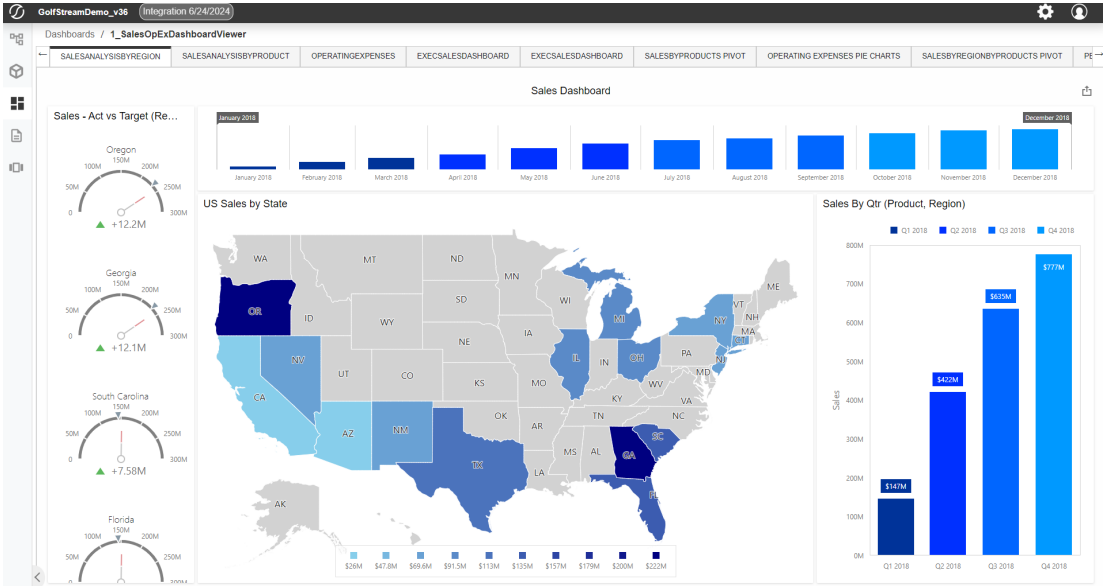
BI Viewer sometimes shows infinite loading screens in Browser that are driven by differences in how the component reads configurations.

Impacted Component	Observed Behavior	Cause	Resolution

BI Viewer	Infinite Loading Screen Appears	A filter is set to a question mark (?).	Update the filter to be set to an empty string. Validated in browser and WinApp.
BI Viewer	Infinite Loading Screen Appears	A Calculated Field is set to a decimal without a leading zero. (Example: .2).	Update the filter to include a leading zero. (Example: 0.2) Validated in browser and WinApp.
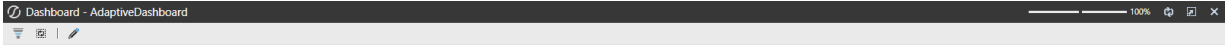
Embedded Dashboards

New client/device specific dashboards configurations allow a single root dashboard to render different embedded dashboards depending on the client or device type. (More details on the use of this configuration are described within the Multi-Client/Device Dashboard Design Approaches below)

Windows Client Dashboard

Web Dashboard

Tablet Dashboard

Phone Dashboard
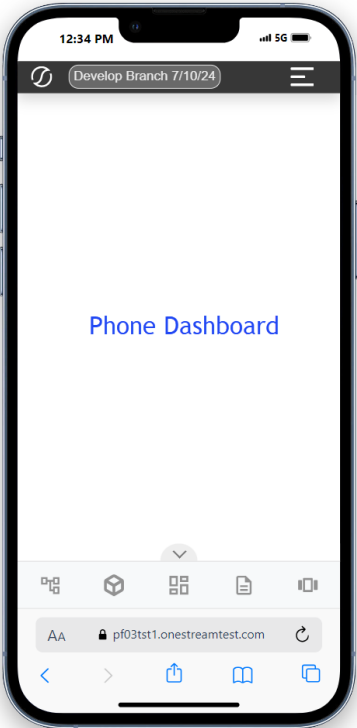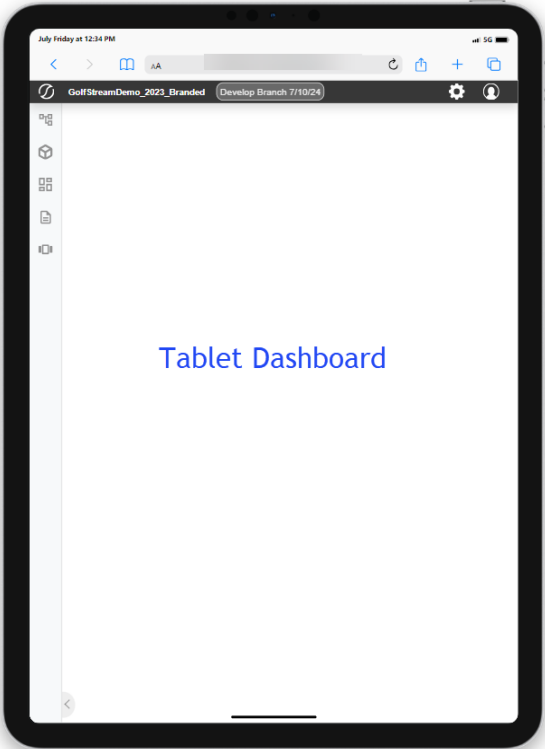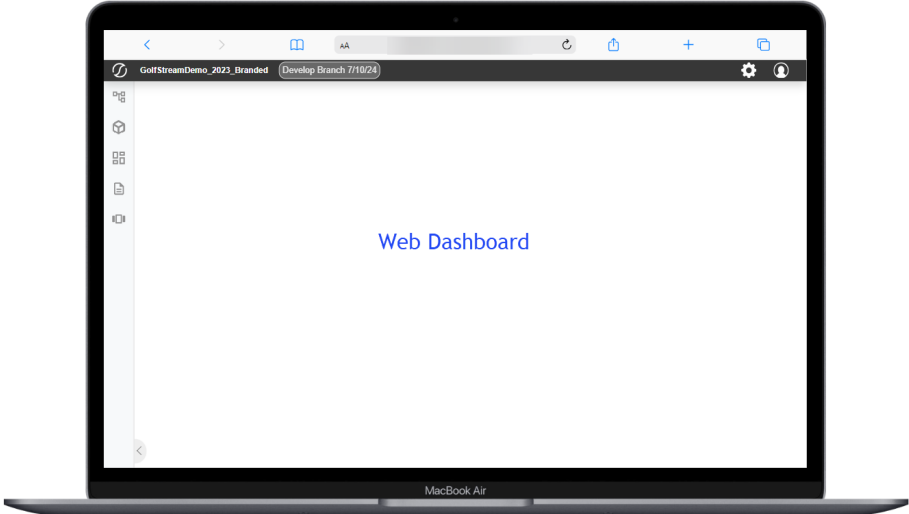
Grid View

Grid options are now listed in a context menu.

SQL Table Editor

Additional options are now listed in a context menu. Edited cells have minor style updates.

☐	1	0	Houston	Actual	2019M1	Chef master	Johnson	John
☐	122	0						
☑	12345	0	Profitability_Defau...	Actual	2017M1	*Manager*	Illescas	Rosa
☐	123455	0				QA Manager		
☐	132	0						
☐	2	0	Houston	Actual	2019M1	waiter	Andrews	Andrew

Grid Keyboard Functionality

The browser experience includes basic keyboard shortcuts. See Appendix A for more information.

Large Data Pivot Grid & Pivot Grid

Pivot Grids in the browser come with a host of new features including Saved Reports, Charts, Calculated Fields, and Keyboard Navigation. Both Pivot Grid and Large Data Pivot Grid benefit from updated data processing that ensures all data is available for aggregation.

Notable Differences in Pivot Grids
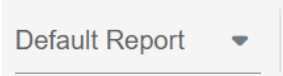
Field Controls are displayed in a floating dialog and launch using the *Show Fieldlist* button in the upper right corner of the pivot grid.

A count of total rows in the dataset is displayed on the top right of the pivot grid.

NOTE: This count will not change when filtering data, it will always display the total count of rows.

Button	Action
	Create new report
	Save a report
	Save as current report
	Rename a current report
	Delete a current report
Default Report ▼	Default Report dropdown
	Show table
	Chart types

	Export
	Subtotals
	Grand totals
	Formatting

New Features in Pivot Grids

Saved Reports – Easily reference past grid configurations by saving reports. The default report will return the Pivot Grid to its initial configuration, while Create, Save, Save As, and Rename buttons manage custom reports. Find reporting functions in the Browser UX Pivot Grid Toolbar.

Charts – This new feature lets users easily visualize the data from a customized pivot grid into a chart. Simply select the Chart icon in the toolbar, then configure the available options. Charts can be saved as reports for future reference using the Saved Reports feature. See Appendix B for a full list of available chart types.

- Options: Show Legend, Multiple Axis

- Explore larger charts using the zoom feature. Available using scroll control on mouse, the zoom toolbar on the chart, or by using keyboard shortcuts.

- Visualize the expansion and collapse of the pivot grid data from the chart using right mouse click menu.
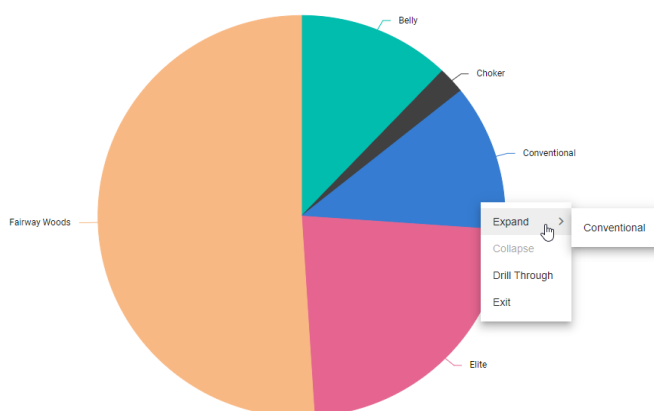
- Keyboard Functionality - A wide array of functions allow users to interact with Pivot Grids using keyboard shortcuts. See Appendix B for a full list of interactions.

- Calculated Fields - The calculated field feature allows users to create custom fields which are not present in the actual data. Users can create these fields using basic mathematical expressions collaborating with existing fields. See Appendix A for syntax.
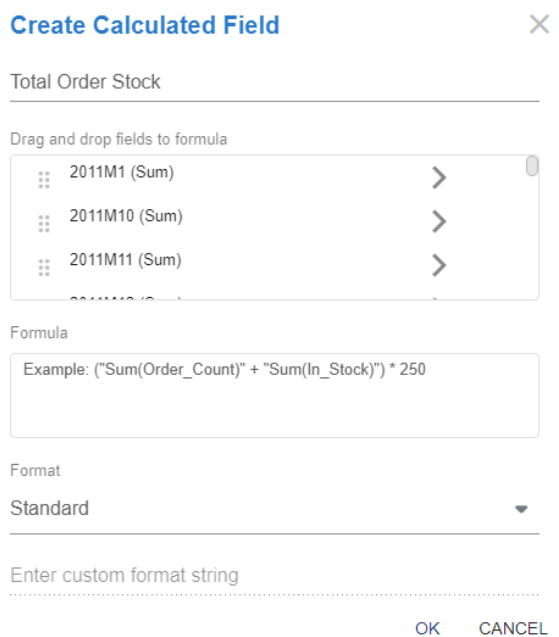
Advanced Charts

There are some charts and configuration properties that are not currently supported. See Appendix B for a detailed breakdown.

Text Editor

The text editor feature mirrors almost all of the windows application functionality including: create, save, open, edit, rich text editing, insert, and layout functions. Additionally,

OneStream functions such as refresh documents, show field codes, and hide field codes. The main difference of this feature is the review tab is not currently available.

Filter Editor

The Filter Editor component gives users a visual interface for modifying and applying advanced logic to filters. While most operators are available in browser, the following are not currently supported: *Not And*, *Not Or*, and *Not Between* (Numeric Values). Only Equals and Does Not Equal generate a data bound dropdown list. Other operators render a text box.

See Dashboard Filter Editor in the Design and Reference Guide for more information.

Spreadsheet

The Modern Browser Experience's Spreadsheet feature provides users with the ability to utilize spreadsheet functionality similar to the Excel Add-in directly through the browser. Spreadsheet behaves similar in the browser experience as it does the Windows Application. Refer to Appendix C for supported and unsupported functionality in Spreadsheet.

Web Content

By default, Web Content is opened in a new tab in the browser. The content may also be rendered directly in a dashboard using an iframe by modifying the following component configuration:

- Open Content in New Tab (Web Only) - Set the value of this field to "False" to open web content in an iframe.

> **NOTE:** Not all websites are compatible with iframes and some may be blocked from rendering in an iframe. Thorough testing of all use cases for Target URLs is strongly encouraged.
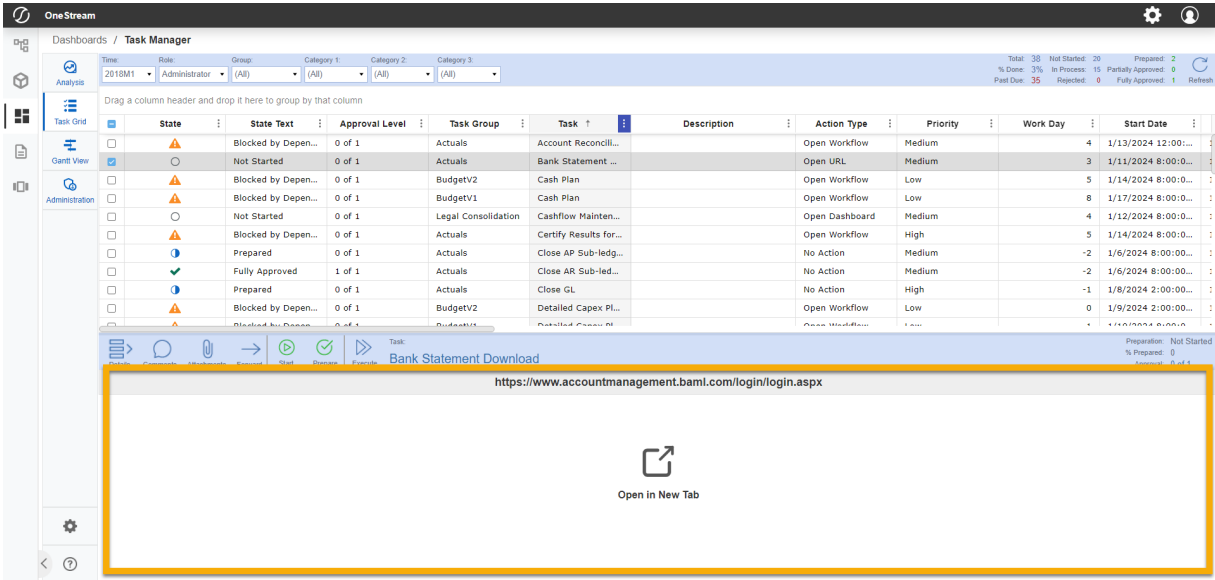
Notable Dashboard Functionality Gaps

Dashboard Layouts

- Canvas
- Dock
- Wrap

Dashboard Components

- Basic Charts
- State Indicator

- Map
- Password Box

- Member Tree
- Sankey Diagram

Navigating Dashboard Gaps

Helpful tips have been added to both the Windows Client and the modern browser experience to assist dashboard designers and end users in identifying dashboard layouts and components that are not yet supported on browser.

In the Windows Client, labels for Windows App Only have been added to both the Dashboard Properties Layout Type and Dashboard Component selection lists.

Dashboard Properties	Dashboard Components
General (Dashboard)	
Name	Cube Views_1
Workspace	IreneTestingWS
Maintenance Unit	IreneTesting
Dashboard Group	Cube Views
Description	
Page Caption	
Notes	
Behavior	
Dashboard Type	(Use Default)
Custom Control	
Required Input Parameters	
Processing	
Literal Parameter Values (e.g., Param1=Value1, ...)	
Template Parameter Values (e.g., Param1=Value1, ...)	
Component Template Repeat Items	(Collection)
Text 1	
Text 2	
Formatting	
Layout Type	Uniform
Is Initially Visible If Embedded	Canvas (Windows App Only)
Display Format	Dock (Windows App Only)
Show Title	Grid
Action (Primary Dashboard Only)	Horizontal Stack Panel
Server Task	Tabs
Load Dashboard Server Task	Uniform
Load Dashboard Server Task Arguments	Vertical Stack Panel
	Wrap (Windows App Only)

A placeholder image indicates if the dashboard layout or component is not currently supported. A hover tooltip specifies the layout or component that is not rendering.
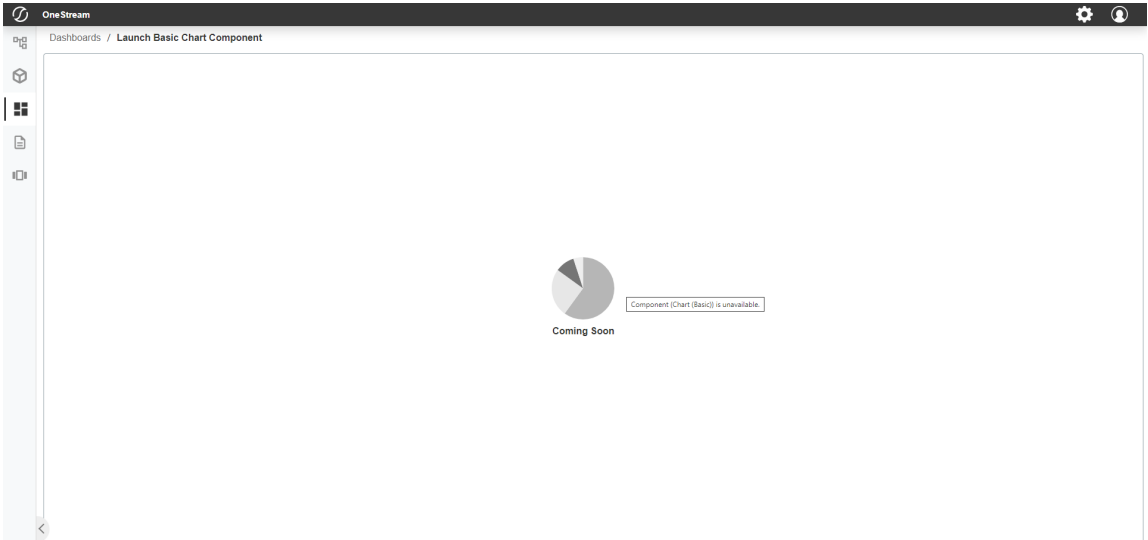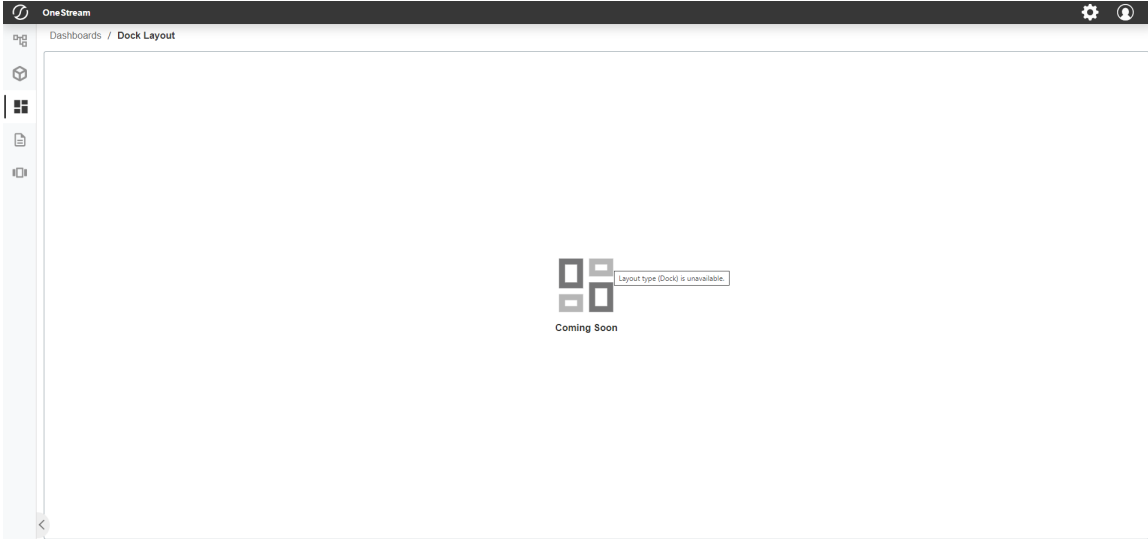
Even if all dashboard layouts and components are supported, we still recommend conducting a full regression test of any dashboard that is being transferred from the Windows Client to the browser to ensure functional parity before production use.

Dashboard Design Process

When building a dashboard for browser use, designers should open both their Windows and browser clients simultaneously. Using the Windows Client, designers can update and save dashboard configurations. To view reflected changes, simply refresh the browser.

When building a dashboard for use on a tablet or phone, designers can follow the same process, using either a physical device or mobile device emulator in a browser. Designers should make sure they log into their browser environment with the emulator properly sized to ensure the browser experience reacts to the device it is emulating.

As with the Windows Client, using dashboard design best practices can significantly improve the performance of a dashboard. The browser experience may have performance sensitivities that differ from the Windows Client.

Multi-client/Device Dashboard Design Approaches

When designing a dashboard that will be used across multiple clients or devices, dashboard designers can use three different approaches to find the right balance of upfront effort, end-user optimization, and maintenance effort that best suits their needs.

Approach 1: Unified Multi-client/Device Dashboards

This approach uses a single non-adaptive dashboard. The designer will most likely have to make compromises to make it usable for each client/device type. These compromises typically result in a simplified dashboard with efficient maintenance. This type of dashboard needs to have the proper combination Client/Device Visibility configurations to be accessible from the OnePlace Dashboard across clients and devices. It can be more difficult for multi-device use due to varying space for dashboard components, but it is an effective way to access dashboards initially. Designers can rely on this approach to

update existing Windows Client dashboards for use. For more complex scenarios, designers should consider building client/device specific dashboards or adaptive dashboards.

Approach 2: Client/Device Specific Dashboards

This approach uses multiple non-adaptive dashboards. Designers create separate dashboards that render equivalent information and functionalities but have different designs that are optimized for specific client and device types. These dashboards need to be added to separate dashboard profiles. Each of the dashboard profiles have different Client/Device Visibility configurations, which limits their visibility in the OnePlace Dashboard list to only the appropriate client or device they were designed for. This approach offers an optimized user experience and low design complexity, but it has a high maintenance effort.

Approach 3: Client/Device Adaptive Dashboards

This approach uses a single adaptive dashboard. Designers need an initial understanding of how their dashboard should adapt to each client or device type intended for use. The designer creates Embedded Dashboards with configurations that link to client/device specific dashboards. The Embedded Dashboards can be placed at any level within nested dashboards. Allowing designers to easily customize components shared across multiple devices or create unique experiences per device type. This approach offers a highly optimized user experience and lower maintenance effort, but it has high design complexity.

Current dashboards that have been designed previously for the OneStream Windows Client may need to be redesigned for an optimized user experience on the browser or alternative device types.

Documents

The Documents panel in the browser offers an intuitive interface has similar navigational patterns to the Windows Application. Files are accessed through the browser are downloaded to the user's local machine, instead of being opened automatically in a native application. Additionally, a context menu has been implemented added at the top of the Documents area, which provides a button to download selected documents. This context menu also provides an button to open the File Explorer and to view File Details. The File Details can also be viewed by double-clicking the filename. When a folder is selected, the context menu dynamically provides options to open the folder in File Explorer and or view the Folder Details.

The specific POV used to view files and folders cannot be modified in the Documents panel, but can be applied through the File Explorer.

> **NOTE:** When downloading files from OneStream, you may need to update the file associations on your device to recognize .xfdoc file extensions.
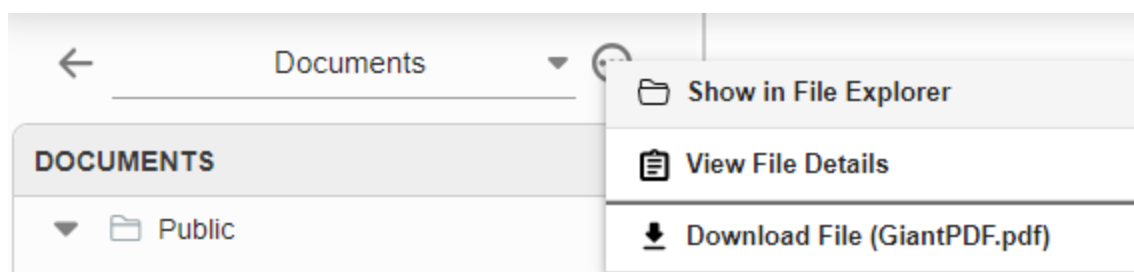
Additionally, by double-clicking the file, folder or selecting View File Details from the context menu, you can see the File Details or Folder Details.
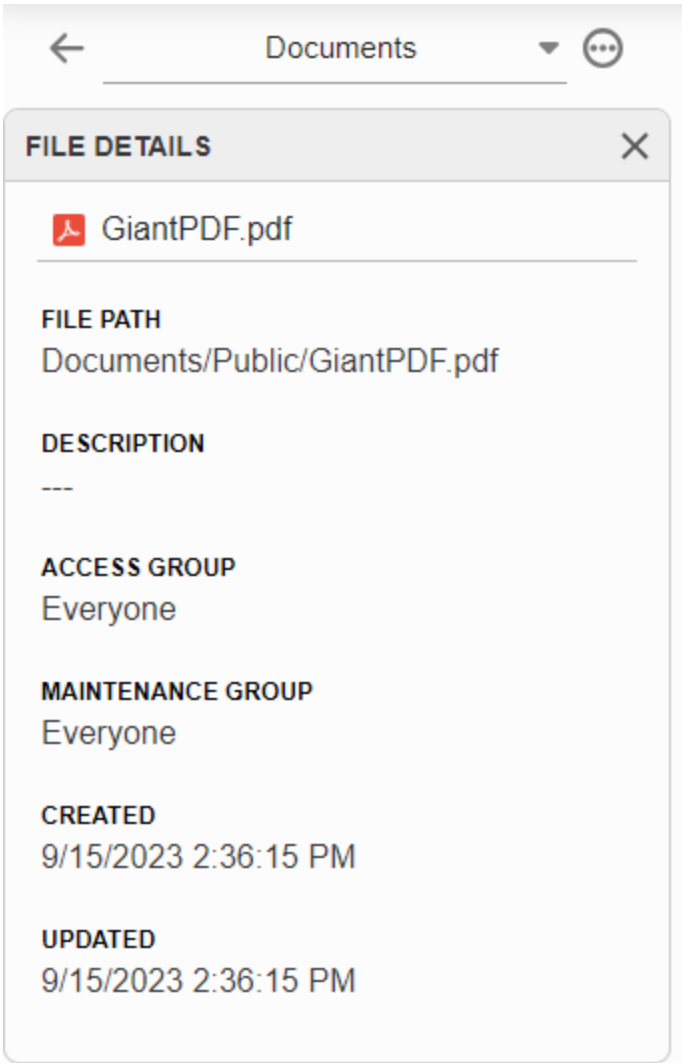
POV

POV is in the OnePlace panel. To view and modify the current Cube POV, select the POV menu from the OnePlace panel. Similar to Workflow POV, the Cube POV Edit menu is displayed as an overlay menu, instead of a pop-up dialog. It's also important to note that information from the Windows Application POV menu is displayed in a few different areas on the browser:

- **Global POV**: Select the gear icon in the header to view current Global POV settings.

- **Workflow POV:** Select Workflow in the OnePlace panel to view and modify the current Workflow POV.

Additional Features

Modify a POV

To modify a POV, click the pencil icon next to each field, make your modification, and click Save.
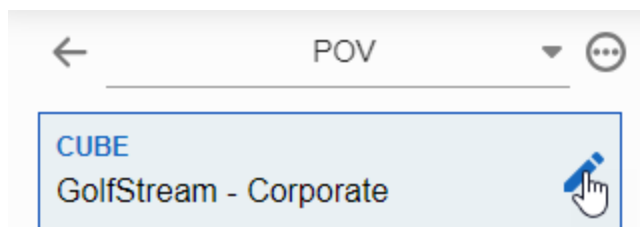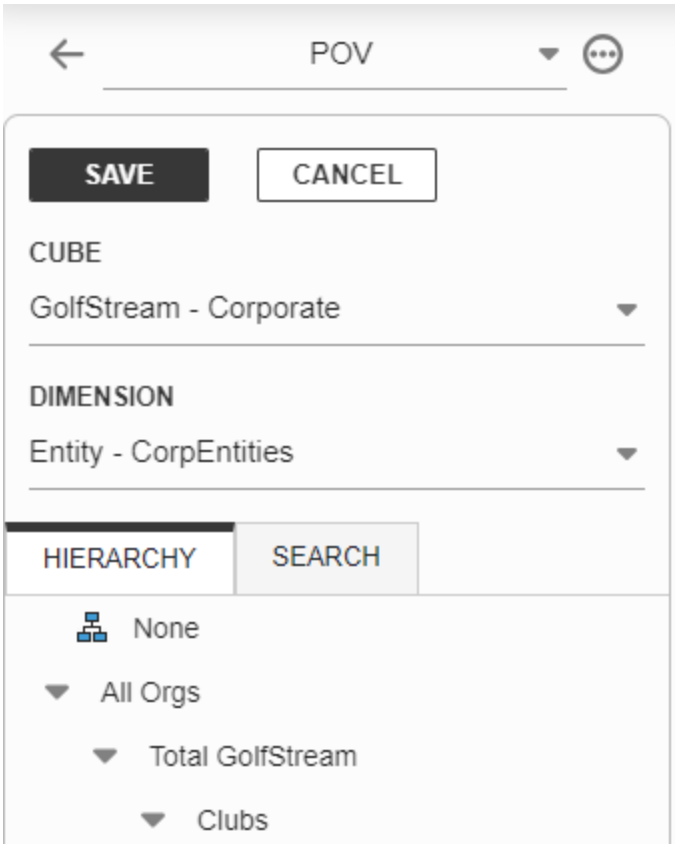
Save a POV

To save a POV, select Save POV from the POV context menu. File Explorer will open with an option to name and save the current POV settings.
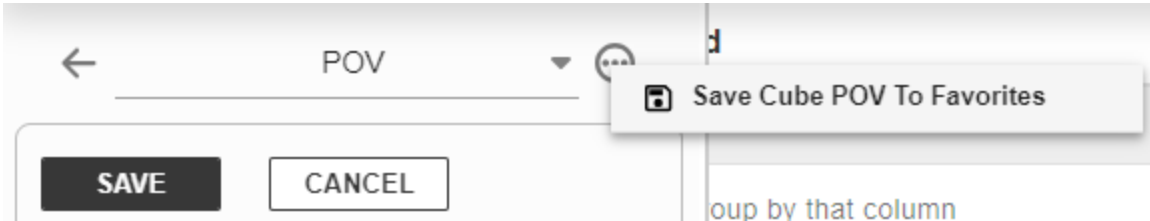
Set a Saved POV

Saved POVs are stored in File Explorer. To set a saved POV, use the **Apply POV** button in the File Explorer toolbar. You must have an item with a File Type of POV selected to see the button. POV can be applied through the File Explorer only.

File Explorer
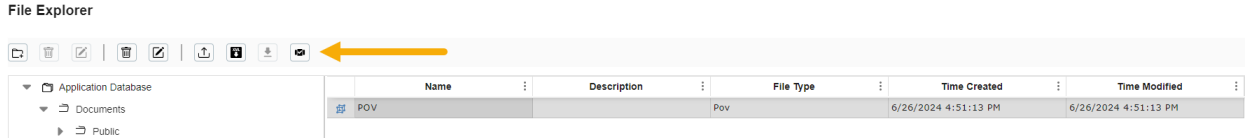
Third-Party Component Technology

OneStream is created using both OneStream's own tools and third-party tools in the authoring, installation, and running of OneStream's supported products. OneStream browser experience third- party developer components include Syncfusion. OneStream customers are not required to purchase these developer tools.

The copyright notice with respect to these third parties is as follows:

- Syncfusion. (c) 2001-2023 Copyright Syncfusion Inc. All rights reserved.

For a comprehensive list of all third-party component tools throughout the OneStream Windows Client application, see Third-Party Component Technology.

Appendix A: Pivot Grids

Calculated Fields Syntax

Add a Calculated Field to Pivot Grids by selecting Show Field List from the toolbar, then Calculated Field. The following syntax gives users several options for customizing fields.

Operator	Name	Syntax
+	addition	X + Y
-	subtraction	X - Y
*	multiplication	X * Y
/	division	X / Y
^	power	X^2
<	Less than	X < Y
<=	less than or equal	X <= Y
>	greater than	X > Y
>=	greater than or equal	X >= Y

==	equal	X == Y
!=	not equal	X != Y
|	OR	X | Y
&	AND	X & Y
?	conditional	condition ? then : else
isNaN	function that checks if the value is not a number	isNaN(value)
!isNaN	function that checks if the value is a number	isNaN(value)
abs	function that returns the absolute value of a number	abs(number)
min	function that returns the minimum value	min (number1, number2)
max	function that returns the maximum value	max (number1, number2)

Keyboard Navigation in Pivot Grids

Focus	
Alt + J	Focuses the Pivot Table component.
PivotTable	
Tab	Moves the cell focus right side. If no cells are focused, it moves to the next active element in the browser page.
Shift + Tab	Moves the cell focus left side. If no cells are focused, it moves to the previous active element in the browser page.
Down arrow	Moves the cell focus downwards.
Up arrow	Moves the cell focus upwards.
Left arrow	Moves the cell focus left side.
Right arrow	Moves the cell focus right side.
Home	Goes to the first cell in the current row.
End	Goes to the last cell in the current row.

Ctrl + Home	Goes to the first cell in the table.
Ctrl + End	Goes to the last cell in the table.
Enter	If the current cell is an expand/collapse cell, it performs expand/collapse operation (drill operation). If the current row/column header is in value sort state, it performs value sorting.
Shift + Enter	If value sorting is enabled in the pivot table and the current cell is a header with respect to its value axis, it performs value sorting to either ascending or descending order.
Grouping Bar	
Tab	Moves to the next active element (field's button) in the grouping bar. If no active elements present, it moves to the next active element in the browser page.
Shift + Tab	Moves to the previous active element (field's button) in the grouping bar. If no active elements present, it moves to the previous active element in the browser page.
Shift + F	If the current active element is a field's button and if it has a filter icon, the filter dialog will be opened to perform filtering.

Shift + S	If the current active element is a field's button and if it has a sort icon, the sorting will be performed to the selected field.
Shift + E	If the current active element is a calculated field's button and if it has an edit icon, the calculated field dialog will be opened to perform editing the selected calculated field.
Enter	Performs the selection operation of the current active element. If the current active element is a field's button and it has a dropdown icon, the aggregation menu will open to perform calculations using aggregation options to the selected value field.
Delete	If the current active element is a field's button, the selected field will be removed from the current report.
Down arrow	If the current active element is a dropdown list, the next item will be selected.
Up arrow	If the current active element is a dropdown list, the previous item will be selected.
Home	If the current active element is a dropdown list, the first item will be selected.
End	If the current active element is a dropdown list, the last item will be selected.

Alt + Down	If the current active element is a dropdown list, the popup will be opened.
Alt + Down	If the current active element is a dropdown list, the popup will be closed.
Esc or Escape	Closes the dropdown list.
Field List	
Ctrl + Shift + F	If the popup field list is enabled in either the pivot table or the pivot chart, the field list dialog will be opened.
Tab	Moves to the next active element in the field list. If no active elements present, it moves to the next active element in the browser page.
Shift + Tab	Moves to the previous active element in the field list. If no active elements present, it moves to the previous active element in the browser page.
Shift + F	If the current active element is a field's button and if it has a filter icon, the filter dialog will be opened to perform filtering.
Shift + S	If the current active element is a field's button and if it has a sort icon, the sorting will be performed to the selected field.

Shift + E	If the current active element is a calculated field's button and if it has an edit icon, the calculated field dialog will be opened to perform editing the selected calculated field.
Enter	Performs the selection operation of the current active element. If the current active element is a field's button and if it has a dropdown icon, the aggregation menu will be opened to perform calculations using aggregation options to the selected value field.
Delete	If the current active element is a field's button, the selected field will be removed from the current report.
Down arrow	If the current active element is a tree node, it moves to the next node.
Up arrow	If the current active element is a tree node, it moves to the previous node.
Left arrow	If the current active element is a tree node, it collapses the current node.
Right arrow	If the current active element is a tree node, it expands the current node.
Home	If the current active element is a tree node, it goes to the first node.
End	If the current active element is a tree node, it goes to the last node.

Space	If the current active element is a tree node or a checkbox element, it will be either checked or unchecked.
Esc or Escape	Closes the popup field list dialog.
Toolbar	
Tab	Moves to the next active option in the toolbar. If no active elements present, it moves to the next active element in the browser page.
Shift + Tab	Moves to the previous active option in the toolbar. If no active elements present, it moves to the previous active element in the browser page.
Enter	Performs the selection operation of the current active element.
Calculated Field	
Shift + E	If the current active element is a field's button and if it has an edit icon in either the field list or grouping bar UI, the calculated field dialog will be opened to perform editing the selected calculated field.
Tab	Moves to the next active element in the calculated field dialog. If no active elements present, it moves to the next active element in the browser page.

Shift + Tab	Moves to the previous active element in the calculated field dialog. If no active elements present, it moves to the previous active element in the browser page.
Enter	Performs the selection operation of the current active element. If the current active element is a tree node, it copies the selected field name/formula to the formula text area to perform calculations.
Down arrow	If the current active element is a tree node, it moves to the next node.
Up arrow	If the current active element is a tree node, it moves to the previous node.
Left arrow	If the current active element is a tree node, it collapses the current node.
Right arrow	If the current active element is a tree node, it expands the current node. If the current active element is a tree node and has a menu icon, the aggregation menu will be opened to select appropriate aggregation type to the selected field.
Home	If the current active element is a tree node, it goes to the first node.
End	If the current active element is a tree node, it goes to the last node.

Esc or Escape	Closes the filter dialog.
Drill Through	
Tab	Moves to the next active element in the drill-through dialog. If the current active element is a Grid cell, it moves the cell focus to right side. If no active elements present, then it moves to the next active element in the browser page.
Shift + Tab	Moves to the previous active element in the drill-through dialog. If the current active element is a Grid cell, it moves the cell focus to left side, If no active elements present, then it moves to the previous active element in the browser page.
Down arrow	Moves the row/cell focus downwards.
Up arrow	Moves the row/cell focus upwards.
Left arrow	Moves the cell focus left side.
Right arrow	Moves the cell focus left side.
Home	Goes to the first cell in the current row.

End	Goes to the last cell in the current row.
Ctrl + Home	Goes to the first cell in the table.
Ctrl + End	Goes to the first cell in the table.
Enter	Performs the selection operation of the current active element.
Esc or Escape	If the cell is in selected state, it deselects all rows/cells. If the row/cell is in edit state, it cancels the current entries in the row/cell. If the current active element is not a row/cell, it closes the drill-through dialog.
F2	If the cell is in selected state, it deselects all rows/cells. If the row/cell is in edit state, it cancels the current entries in the row/cell. If the current active element is not a row/cell, it closes the drill-through dialog.
Insert	Adds a new row/cell in the data grid.
Delete	Removes the selected row in the data grid.
Filter Dialog	
Shift + F	If the current active element is a field's button and if it has a filter icon in either the field list or grouping bar UI, the filter dialog will be opened to perform filtering.

Tab	If the current active element is a field's button and if it has a filter icon in either the field list or grouping bar UI, the filter dialog will be opened to perform filtering.
Shift + Tab	Moves to the next active element in the filter dialog. If no active elements present, it moves to the next active element in the browser page.
Enter	Moves to the previous active element in the filter dialog. If no active elements present, it moves to the previous active element in the browser page.
Down arrow	If the current active element is a tree node, it moves to the next node.
Up arrow	If the current active element is a tree node, it moves to the previous node.
Left arrow	If the current active element is a tree node, it collapses the current node. If the current active element is a tab, it moves focus to the previous tab element.
Right arrow	If the current active element is a tree node, it expands the current node. If the current active element is a tab, it moves focus to the next tab element.
Home	If the current active element is a tree node, it goes to the first node.

End	If the current active element is a tree node, it goes to the last node.
Space	If the current active element is a tree node or a checkbox element, it will be either checked or unchecked.
Esc or Escape	Closes the filter dialog.
Alt + Down	If the current active element is a DropDownList or DatePicker or DateTimePicker, the popup will be opened.
Alt + Up	If the current active element is a DropDownList or DatePicker or DateTimePicker, the popup will be closed.

Chart Types in Pivot Grids

These are the available chart types within Pivot Grids.

Column, Stacked Column, 100% Stacked Column	Bubble
Bar, Stacked Bar, 100% Stacked Bar	Pareto
Line, Stacked Line, Step Line, 100% Stacked Line	Radar
Area, Stacked Area, Step Area, 100% Stacked Area	Pie
Scatter	Doughnut

Polar	Funnel
Spline	Pyramid

Appendix B: Advanced Charts

The following chart types are currently supported by the Advanced Charts dashboard component.

XY2D	Area, Area Stacked, Area Full Stacked, Area Spline, Area Step, Bar, Bar Stacked, Bar Full Stacked, Bar Side By Side Stacked, Bar Side By Side Full Stacked, Line, Line Stacked, Line Full Stacked, Line Step, Waterfall
Simple2D	Nested Donut, Pie And Donut

The following chart types are not currently supported by the Advanced Charts dashboard component.

XY2D	Area Range, Bar Side By Side, Bar Range Overlapped, Bar Range Overlapped Waterfall, Bar Range Side By Side, Bar Range Side By Side Waterfall, Bubble, Candlestick, Line Scatter, Point, Spline, Stock
Simple2D	Funnel
Polar2D	Polar Area, Polar Line, Polar Point

Radar2D	Radar Area, Radar Line, Radar Point

Configuration Properties

The following chart configuration properties are not currently supported.

Chart	Show Toggle Size Button, Point Label Text Format, Enable Animations
Crosshair	Crosshair Enabled, Show Crosshair Lines, Show Crosshair Labels, Crosshair Label Mode, Crosshair Label Text Format
Legend	Show Check Boxes, Show Border
X-Axis, Y-Axis, Secondary Y-Axis	Text Format, Use Automatic Step, Step, Interlaced, Interlaced Color, Show Minor Grid Lines
Y-Axis, Secondary Y-Axis	Scale Break Style Type, Maximum Number of Scale Breaks
Series Properties	Model Display Type, Marker Size, Bar Width, Pie Hole Radius Percent
Waterfall Series Properties	Total Included in Series

Appendix C: Spreadsheet Features and Functionality

Supported Spreadsheet Features and Functionality

Component Integration	• Dashboards • Workflow Forms

Data Sources and Related Functionality	- Spreadsheet Files - Create, open, edit, and save - Cube View Connections - Create, display, edit, and remove - Refresh data and submit changes - Support for Cube View formatting - Table View Definitions - Create, display, edit, and remove - Refresh data and submit changes - Retrieve Functions - Supported Functions - XFGetCell, XFGetCell5, XFGetFXRate, XFGetCalculatedFxRate, XFGetMemberProperty, XFGetRelationshipProperty, XFGetHierarchyProperty, XFGetMemberInfo XFSetCell, XFSetFXRate - Refresh data and submit changes (XFSetCell only)

Ribbon Functions	• New (Create Spreadsheet)
	• Open (Local File, OneStream System File, Application Workspace File, System Workspace File)
	• Save As (OneStream System File, Application Workspace File, System Workspace File)
	• Refresh Sheet and Refresh Workbook
	• Submit Sheet and Submit Workbook

Unsupported Spreadsheet Features and Functionality

Charts	• Limited Chart Types
	• Color Customization

Functions	- Quick Views - Formula Bar - Macros - Solver - Sheet References - Document Properties - Context Menu Features and Functions - Insert Copied/Cut Cells - Keyboard Controls: - Shift + End - Ctrl + PageUp / Ctrl + PageDown (does not set tab focus) - Ctrl + N (overwrites spreadsheet with unsaved changes)
Formatting	- Preview Font Formats - Conditional Formatting - Cell Borders (last selection is reset after format is applied)

Appendix D: Supported Number Formats

The following Number Formats are supported:
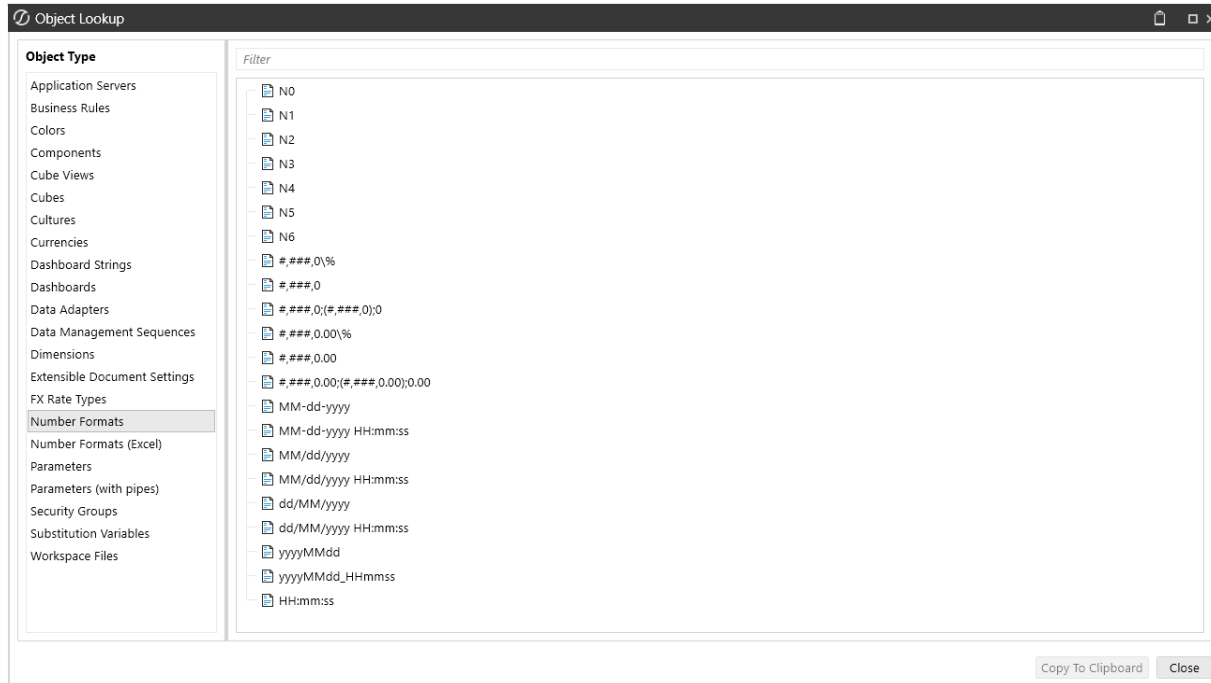
Appendix E: Smart Link APIs

Create Smart Link

The following business rules create Smart Links and allow developers to provide dynamic parameters (Custom Substitution Variables) using name-value pairs. In the code snippets below, link expiration is set to 240 hours (10 days) and outputs the Smart Link in a message box. Optionally, the link can direct a user to a different OneStream environment using *Host Override* or append a custom name to the end of the Smart Link URL using *Custom Name*.

C# Code Snippet:

```
public XFSelectionChangedTaskResult CreateSmartLink(SessionInfo si, BRGlobals
globals, object api, DashboardExtenderArgs args)
        {
            // Create new smart link and set expiration
            var smartlink = new XFSmartLink();
            smartlink.IsSystemlevel = true;
            smartlink.ExpirationDateUtc = DateTime.UtcNow.AddHours(240);

            // Optional Properties
            // Host override: Allows to change the host value. Must be a value
host
            //smartlink.HostOverride = "https://buxtest.com:444";

            // Custom Name: Adds a suffix at the end of the smart link.
Alphanumeric, 50 characters max.
            //smartlink.CustomName = "ThisIsACustomName";

            var dashboard = new XFSmartLinkDashboard();

            // Change values below to target different applications/ Dashboards
            dashboard.ApplicationName = "GolfStreamDemo_v36";
            dashboard.DashboardName = "SmartLinks";
            dashboard.WorkspaceName = "Default";
            dashboard.DashboardProfileName = "Default";
            dashboard.CustomSubstVars = new Dictionary<string, string>();

            // Add/ Remove the lines below to modify the custom subst vars that
```

```
will be applied.
            dashboard.CustomSubstVars.Add("parameter1", "value");
            dashboard.CustomSubstVars.Add("parameter2", "value2");

            smartlink.TargetObject = dashboard;

            string sValue = BRApi.State.CreateSmartLink(si, smartlink);
            var match = System.Text.RegularExpressions.Regex.Match(sValue,
@"share/([A-Z0-9]{32})");

            BRApi.Dashboards.Parameters.SetLiteralParameterValue(si, false,
"slid", match.Groups[1].Value);

            var result = new XFSelectionChangedTaskResult();
            result.IsOK = true;
            result.ShowMessageBox = true;
            result.Message = sValue;
            return result;
        }
```

VB Code Snippet:

```
Public Function CreateSmartLink(ByVal si As SessionInfo, ByVal globals As
BRGlobals, ByVal api As Object, ByVal args As DashboardExtenderArgs) As
XFSelectionChangedTaskResult
            ' Create new smart link and set expiration
            Dim smartlink = New XFSmartLink()
            smartlink.IsSystemlevel = True
            smartlink.ExpirationDateUtc = DateTime.UtcNow.AddHours(240)

            ' Optional Properties
            ' Host override: Allows To change the host value. Must be a value host
            'smartlink.HostOverride = "https://buxtest.com:444"

            ' Custom Name: Adds a suffix at the End Of the smart link.
Alphanumeric, 50 characters max.
            'smartlink.CustomName = "ThisIsACustomName"

            Dim dashboard = New XFSmartLinkDashboard()

            ' Change values below to target different applications/ Dashboards
            dashboard.ApplicationName = "GolfStreamDemo_v36"
            dashboard.DashboardName = "SmartLinks"
            dashboard.WorkspaceName = "Default"
            dashboard.DashboardProfileName = "Default"
            dashboard.CustomSubstVars = New Dictionary(Of String, String)()

            ' Add/ Remove the lines below to modify the custom subst vars that
will be applied.
            dashboard.CustomSubstVars.Add("parameter1", "value")
```

```
            dashboard.CustomSubstVars.Add("parameter2", "value2")

            smartlink.TargetObject = dashboard

            Dim sValue As String = BRApi.State.CreateSmartLink(si, smartlink)
            Dim match = System.Text.RegularExpressions.Regex.Match(sValue, "share/
([A-Z0-9]{32})")

            BRApi.Dashboards.Parameters.SetLiteralParameterValue(si, False,
"slid", match.Groups(1).Value)

            Dim result = New XFSelectionChangedTaskResult()
            result.IsOK = True
            result.ShowMessageBox = True
            result.Message = sValue
            Return result
        End Function
```

Retrieve Smart Link

Smart Links are stored for a set period of time. Using the following business rules,
developers can retrieve a stored Smart Link using slid (Smart Link ID)

C# Code Snippet:

```
public XFSelectionChangedTaskResult GetSmartLink(SessionInfo si, BRGlobals
globals, object api, DashboardExtenderArgs args)
        {
            var id = BRApi.Dashboards.Parameters.GetLiteralParameterValue(si,
false, "slid");

            var guidId = Guid.Parse(id);
            XFSmartLink sValue = BRApi.State.GetSmartLink(si, true, guidId);
            var text = System.Text.Json.JsonSerializer.Serialize(sValue);

            var result = new XFSelectionChangedTaskResult();
            result.IsOK = true;
            result.ShowMessageBox = true;
            result.Message = text;
            return result;
        }
```

VB Code Snippet:

```
Public Function GetSmartLink(ByVal si As SessionInfo, ByVal globals As BRGlobals,
ByVal api As Object, ByVal args As DashboardExtenderArgs) As
XFSelectionChangedTaskResult
        Dim id = BRApi.Dashboards.Parameters.GetLiteralParameterValue(si,
False, "slid")

        Dim guidId = Guid.Parse(id)
        Dim sValue As XFSmartLink = BRApi.State.GetSmartLink(si, True, guidId)
        Dim text = System.Text.Json.JsonSerializer.Serialize(sValue)

        Dim result = New XFSelectionChangedTaskResult()
        result.IsOK = True
        result.ShowMessageBox = True
        result.Message = text
        Return result
    End Function
```

Delete Smart Link

The following business rules delete stored Smart Links.

C# Code Snippet:

```
public XFSelectionChangedTaskResult DeleteSmartLink(SessionInfo si, BRGlobals
globals, object api, DashboardExtenderArgs args)
        {
        var id = BRApi.Dashboards.Parameters.GetLiteralParameterValue(si,
false, "slid");
        var guidId = Guid.Parse(id);

        BRApi.State.DeleteSmartLink(si, true, guidId);
        var text = "Deleted";

        var result = new XFSelectionChangedTaskResult();
        result.IsOK = true;
        result.ShowMessageBox = true;
        result.Message = text;
        return result;
    }
```

VB Code Snippet:

```
    Public Function DeleteSmartLink(ByVal si As SessionInfo, ByVal globals As
BRGlobals, ByVal api As Object, ByVal args As DashboardExtenderArgs) As
```

```
XFSelectionChangedTaskResult
        Dim id = BRApi.Dashboards.Parameters.GetLiteralParameterValue(si,
False, "slid")
        Dim guidId = Guid.Parse(id)

        BRApi.State.DeleteSmartLink(si, True, guidId)
        Dim text = "Deleted"

        Dim result = New XFSelectionChangedTaskResult()
        result.IsOK = True
        result.ShowMessageBox = True
        result.Message = text
        Return result
    End Function
```

Modify Smart Link

The following business rules are designed to overwrite an existing Smart Link while retaining its URL. Since modifying a link overwrites all stored values, all required values need to be provided when using this rule.

C# Code Snippet:

```csharp
public XFSelectionChangedTaskResult ModifySmartLink(SessionInfo si, BRGlobals
globals, object api, DashboardExtenderArgs args)
        {
        var id = BRApi.Dashboards.Parameters.GetLiteralParameterValue(si,
false, "slid");
        var guidId = Guid.Parse(id);

        var smartlink = new XFSmartLink();
        smartlink.IsSystemlevel = true;
        smartlink.ExpirationDateUtc = DateTime.UtcNow.AddHours(240);

        var dashboard = new XFSmartLinkDashboard();

        // Change values below to target different applications/ Dashboards
        dashboard.ApplicationName = "GolfStreamDemo_v36";
        dashboard.DashboardName = "Modified";
        dashboard.WorkspaceName = "Default";
        dashboard.DashboardProfileName = "Default";
        dashboard.CustomSubstVars = new Dictionary<string, string>();

        // Add/ Remove the lines below to modify the custom subst vars that
```

```
will be applied.
          dashboard.CustomSubstVars.Add("parameter1", "test");
          dashboard.CustomSubstVars.Add("parameter2", "value2");
          dashboard.CustomSubstVars.Add("parameter3", "value3");
          dashboard.CustomSubstVars.Add("parameter4", "value4");

          smartlink.TargetObject = dashboard;

          BRApi.State.ModifySmartLink(si, true, guidId, smartlink);

          var result = new XFSelectionChangedTaskResult();
          result.IsOK = true;
          result.ShowMessageBox = true;
          result.Message = "Modified";
          return result;
       }
    }
}
```

VB Code Snippet:

```
        Public Function ModifySmartLink(ByVal si As SessionInfo, ByVal globals As
BRGlobals, ByVal api As Object, ByVal args As DashboardExtenderArgs) As
XFSelectionChangedTaskResult
          Dim id = BRApi.Dashboards.Parameters.GetLiteralParameterValue(si,
False, "slid")
          Dim guidId = Guid.Parse(id)

          Dim smartlink = New XFSmartLink()
          smartlink.IsSystemlevel = True
          smartlink.ExpirationDateUtc = DateTime.UtcNow.AddHours(240)

          Dim dashboard = New XFSmartLinkDashboard()
          dashboard.ApplicationName = "GolfStreamDemo_v36"
          dashboard.DashboardName = "Modified"
          dashboard.WorkspaceName = "Default"
          dashboard.DashboardProfileName = "Default"
          dashboard.CustomSubstVars = New Dictionary(Of String, String)()

          ' Change values below to target different applications/ Dashboards
          dashboard.CustomSubstVars.Add("parameter1", "test")
          dashboard.CustomSubstVars.Add("parameter2", "value2")
          dashboard.CustomSubstVars.Add("parameter3", "value3")
          dashboard.CustomSubstVars.Add("parameter4", "value4")

          smartlink.TargetObject = dashboard

          BRApi.State.ModifySmartLink(si, True, guidId, smartlink)
```

```
            Dim result = New XFSelectionChangedTaskResult()
            result.IsOK = True
            result.ShowMessageBox = True
            result.Message = "Modified"
            Return result
        End Function
    End Class
End Namespace
```