



AI System Diagnostics Guide

Copyright © 2025 OneStream Software LLC. All rights reserved.

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

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

Table of Contents

Overview 1

Setup and Installation 3

 Dependencies 3

 Install AI System Diagnostics 5

 Set Up AI System Diagnostics 6

 Create Tables 6

Global Settings 8

 Security Settings 8

 Uninstall 8

Home 10

Active Jobs 13

Completed Jobs 14

 Completed Jobs Table 14

 Scan Report 15

 Scan Report Example 16

Conditions 17

 Conditions Table 18

 Create a New Condition 19

Table of Contents

 Edit a Custom Condition21

Help and Miscellaneous Information 23

 Display Settings23

 Package Contents and Naming Conventions23

 OneStream Solution Modification Considerations25

Overview

AI System Diagnostics builds on the capabilities of System Diagnostics by introducing the ability to scan custom code within the application for inefficiencies and poor coding practices that may negatively impact the environment's performance. By finding and eliminating these inefficiencies within the existing code, AI System Diagnostics helps ensure an environment's performance is optimized.

AI System Diagnostics includes a set of Predefined Conditions that identify common poor coding practices, complete with descriptions and examples, as well as a compiled list of OneStream-specific issues that can be found in custom code. To run a scan, users select the conditions they wish to evaluate, along with the option to specify which components to scan: Business Rules (here includes workspace assemblies and member formulas), Data Adapters, or both.

Once a scan is complete, a detailed report is generated containing the following:

- **Condition Name:** The name of the condition identified by the scan.
- **Scan Item Type:** The type of item containing the condition, either Business Rule or Data Adapter.
- **Scan Item:** The name of the file or item containing code that the scan detected with the condition.
- **Line Number Start:** Displays the first line number of the region of code where the scan has found its criteria.
- **Line Number End:** Displays the final line number of the region of code where the scan has found its criteria.

Overview

- **Severity Level:** Displays how detrimental this condition is to the performance of the system (Low, Medium, High). Some conditions are minor inconveniences to the system, while others play a vital role in the overall health of the environment. Any marked as High should be analyzed closely.
- **Explanation:** A Large Language Model (LLM)-generated explanation for why this region of code met the specified condition and why it is causing an inefficiency in their environment.
- **Suggested Fix:** An LLM-generated suggested fix for the region of code determined to meet the condition.

With AI System Diagnostics you can:

- Apply Predefined Conditions.
- Create and edit Custom Conditions.
- Run scans on the Home page.
- Run jobs asynchronously.
- Use multi-select tagging to filter and find relevant conditions easily.
- Download the report generated from a scan.

IMPORTANT: AI System Diagnostics is a paid solution and runs on Platform 9.0. To access the SensibleAI Studio and verify your Platform version, contact your Account Executive.

Setup and Installation

AI System Diagnostics now comes preinstalled alongside System Diagnostics in your OneStream environment. To use AI capabilities, the appropriate AI package must be enabled, containing Xperiflow and AI Plugin - OneStream System Diagnostics. Please contact your OneStream Account Executive for further instructions.

AI System Diagnostics can be installed within an existing OneStream application, yet it enables you to analyze the performance of all applications in a given environment. See [Install System Diagnostics](#) for details.

This section contains important details related to the planning, configuring, and installation of your solution. Before you install the solution, familiarize yourself with these details.

Dependencies

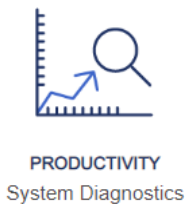
Component	Description
OneStream 9.0.0 or later	Minimum OneStream Platform version required to install this version of AI System Diagnostics.
Xperiflow 4.0.0 or later	Minimum Xperiflow Engine version required to install this version of AI System Diagnostics.

Component	Description
PSD Xperiflow 4.0.0 or later	Minimum Xperiflow Engine version required to install AI Plugin - OneStream System Diagnostics (PSD) which is needed for AI System Diagnostics.

Install AI System Diagnostics

AI System Diagnostics is an independent dashboard within System Diagnostics and it will install alongside it. Follow these steps to install the AI System Diagnostics dashboard:

1. In the OneStream Solution Exchange, go to **OneStream Solution > System Diagnostics**.



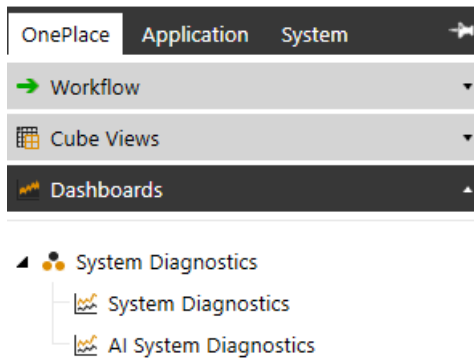
2. On the System Diagnostics Solution page, select the appropriate OneStream platform version from the **Minimum Platform Version** drop-down list.
3. Select the most recent version from the **Solution Version** drop-down list and click **Download**.
4. Log in to OneStream.
5. On the **Application** tab, click **Tools > Load/Extract**.
6. On the **Load** tab, locate the solution package using the **Select File** icons and click **Open**.
7. When the solution's file name appears, click **Load**.
8. Click **Close** to complete the installation.

Set Up AI System Diagnostics

IMPORTANT: AI System Diagnostics is an independent dashboard within System Diagnostics and will install alongside System Diagnostics. See *Set Up System Diagnostics* in the *System Diagnostics Guide*.

The first time you run AI System Diagnostics, you are guided through the table setup process.

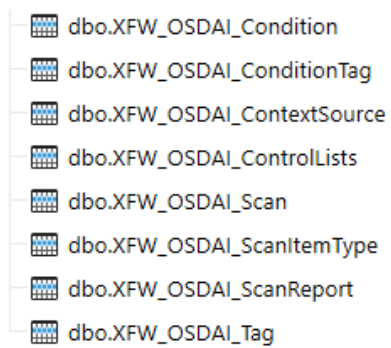
In OneStream, click **Community Solution > Dashboards > System Diagnostics > AI System Diagnostics**.



Create Tables

1. Click **Step 1: Setup Tables**.

The following tables are created in the OneStream framework database:



2. Click **Step 2: Launch Solution** to open AI System Diagnostics.

Global Settings



The **Global Settings** page contains key properties that guide administration are set in the **Security Settings** tab, as well as the **Uninstall** tab, which provides solution uninstall options.

Security Settings

Administrators can use the Security Management settings to determine which groups can manage different parts of the AI System Diagnostics solution.

1. Select from the **Security Role** drop-down menu and choose a security group.
2. Click the **Save** button.

Uninstall

Use the Uninstall feature to uninstall the AI System Diagnostics interface or the entire solution. If done as part of an upgrade, any modifications performed on standard AI System Diagnostics objects are removed.

1. On the **Global Options** page, click the **Uninstall** tile.
2. On the **Uninstall** page, select **Uninstall UI** or **Uninstall Full**.
 - Use the **Uninstall UI** button to remove AI System Diagnostics, including related dashboards and business rules, but retain the database and related tables.

- Use the **Uninstall Full** button to remove all related data tables, data, and AI System Diagnostics dashboards and business rules. Choose this option to completely uninstall AI System Diagnostics or to perform an upgrade that is so significant in its changes to the data tables that this method is required.

CAUTION: Uninstall procedures are irreversible.

3. On the **Confirm** dialog box, click the **Uninstall** button to proceed.

Home

Use the Home page to view both Predefined and Custom Conditions.

AI System Diagnostics

1 HomeActive JobsCompleted JobsConditions2

Home

3 Refresh456 Conditions Found: 4978 Run

Search Conditions

Tags

Scan Item Type

Severity Level

Reset

5

Condition Name	Condition Description	Scan Item Type	Severity Level
Error Logs	Error logging should be used only for meaningful debugging. Excessive or unnecessary logging increases system load and makes critical errors harder to identify.	Business Rule	Low
String-Based Member References	Using string literals for member names in business rules can lead to performance issues and increase the risk of runtime errors due to typos or member name changes. Instead, use integer member IDs, which are faster to evaluate and more reliable.	Business Rule	Low
Nested For Loops	Excessive nesting of 'for' loops increases time complexity, leading to high CPU usage and performance degradation, especially for large datasets. Refactoring with optimized algorithms or breaking down loops using helper functions can improve efficiency.	Business Rule	High
Looping Calls to App DB	Calling the App DB inside a loop can cause severe performance degradation due to multiple sequential database calls. Instead, batch processing or bulk queries should be used to optimize performance.	Business Rule	High
Large Methods	Methods exceeding 30 lines reduce readability, maintainability, and testability. Breaking them into smaller, well-defined methods improves code organization and reusability.	Business Rule	Low
Large SQL Queries	Unconstrained SQL queries that lack WHERE or HAVING clauses can return excessive rows and columns, leading to performance degradation and increased database load. Filtering data at the query level improves efficiency and prevents unnecessary data retrieval.	All	Low
Exceeding MaxDegreeParallelism	Setting MaxDegreeOfParallelism beyond the application's configured limit can cause severe performance degradation, excessive CPU usage, and resource contention. This is particularly problematic for computationally expensive processes like ETL workflows.	Business Rule	High
Large Comment Blocks	Excessive comment blocks clutter the code, reducing readability and making maintenance more difficult. Comments should be concise, relevant, and used only when necessary to clarify complex logic.	Business Rule	Low
Event Handler Use	Excessive or improperly scoped event handlers can cause serious application degradation by triggering unnecessary executions, leading to performance bottlenecks and unintended side effects.	Business Rule	Medium
Mismatched Data Types in SQL Query	Specifying the wrong data scan_type in a comparator can cause a SQL query to take much longer than it would without. This is due to the fact that the query engine has to convert your inputs to the correct data scan_type before they can be compared.	All	Medium
Using unnecessary 'DISTINCT'	Using 'DISTINCT' to remove duplicates instead of applying proper filtering or aggregation can lead to unnecessary processing and performance inefficiencies in SQL queries.	All	Low
Unused variables/parameters	Declaring variables or parameters that are never used in logic statements, methods, or classes decreases code readability and wastes memory allocation, leading to minor but unnecessary performance overhead. "Note": Code that has been commented out can be disregarded and does not trigger this condition.	Business Rule	Low
SQL Queries inside Loops	Executing SQL queries inside loops results in multiple database calls, leading to performance degradation and increased query execution time. Using batch queries significantly improves efficiency.	All	Low
Unencrypted Passwords/Secrets	Storing sensitive information such as passwords, API keys, or credentials in plain text within code creates a serious security vulnerability. These values should always be stored securely using environment variables, secret management tools, or encrypted configuration files.	Business Rule	Low
Double Instead of Decimal	Using 'Double' instead of 'Decimal' for financial or precise calculations can result in accuracy loss due to floating-point precision errors. 'Decimal' is the preferred choice for such scenarios.	All	Low

1. Navigational tabs:

- **Home:** View, search, and filter conditions; run scans
- **Active Jobs:** View all active jobs
- **Completed Jobs:** Download reports from completed jobs
- **Conditions:** Create and edit Custom Conditions

NOTE: The Conditions tab can only be viewed by admins.

2. Global Options:

- **GlobalSettings:** Contains Security Settings and Uninstall
- **Help:** Contains documentation for AI System Diagnostics

3. **Refresh:** Use to clear cache. Refreshes the conditions found, the conditions table, and will clear the search and tags.

4. **Filter:**

- **Search Condition:** Use the text box to search for conditions by name
- **Tags:** Set filters to search for selected conditions. The Conditions Table will refresh to show search results. Multiple tags can be selected from each category.
 1. **Scan Item Type:** All, Data Adapter, Business Rule
 2. **Severity Level:** All, High, Medium, Low

NOTE: Clicking **Search** with nothing in the text field will reset the search filter, but selected tags remain.

5. **Reset:** Clears the filters and search box. The Conditions Table will show all conditions.

6. **Conditions Found:** Displays the number of conditions

7. **Conditions Table:**

- **Name:** The condition name. This is set for Custom Conditions only on the Condition page, or are predefined by OneStream.
- **Description:** Displays the description of the condition. This is set for Custom Conditions only on the Conditions page, or are predefined by OneStream.
- **Scan Item Type:** Displays the Predefined or Custom Condition type. This is set for Custom Conditions only on the Conditions page, or are predefined by OneStream.

- **Severity Level:** Displays the priority that the condition should be addressed. This is set for Custom Conditions only on the Conditions page, or are predefined by OneStream

NOTE: All fields can be filtered.

8. **Run:** Runs a report of specified conditions.
 1. Clicking **Run** after you have selected one or more condition will prompt a dialog box asking if you wish to proceed. Click **Confirm**.
 2. Clicking **Run** without selecting conditions will populate a dialog box prompting you to select the conditions you wish to run.

Active Jobs

The Active Jobs page shows all active jobs with a status of Queued or Running. The number of Active Jobs will appear at the top of the page.

- **Start Time:** Displays the local time setting when the job begins
- **User:** Displays the name of the user that initiates the run
- **Condition Name:** Displays the uniquely assigned name of the conditions that are run in the given job
- **Status:** Displays the Job Status from the AI Engine

Click the **Refresh** button to refresh the page and display active jobs.

Active Jobs (3)



Refresh

Start Time	User	Condition Names	Status
6/12/2025 8:57:31 PM		Error Logs, Nested For Loops, Large Comment Blocks	Running
6/12/2025 8:58:16 PM		SQL in Finance Rule	Running
6/12/2025 8:58:40 PM		Utilize Sequential ID rather than GUID	Running

IMPORTANT: Run times may be impacted by varying factors, such as how many jobs are run at one time or the size of the packages. Additionally, while a job is running, you cannot create or edit Custom Conditions on the Conditions page.

Completed Jobs

Use the Completed Jobs page to download reports of completed scans. Only jobs with the status of Completed or Failed display in the Conditions Table. The number of Completed Jobs is shown at the top of the page.


On this page you can:


- Download a job by selecting it and clicking **Download**.

NOTE: If you select a failed condition and click Download, you are prompted to view the Error Log because there is no job report to download.

- Update the Completed Jobs table by clicking **Refresh**.
- Repeat a job by selecting it and clicking **Rerun**.

Completed Jobs (23)


Download


Refresh

Rerun

Completed Jobs Table

The table displays the following:

- **Start Time:** The local time when the job begins.
- **End Time:** The local time when the job ends.

Completed Jobs

- **User:** The name of the user that executed the run.
- **Condition Name:** The unique name of the conditions that the job ran.
- **Status:** The Job Status from the SensibleAI Studio.

Start Time	End Time	User	Condition Names	Status
6/6/2025 12:59:36 PM	6/6/2025 1:06:15 PM	OSDAI_edAdmin	Error Logs	Completed
6/6/2025 12:34:51 PM	6/6/2025 12:59:10 PM	OSDAI_edAdmin	Error Logs, String-Based Membe	Completed
6/6/2025 12:33:00 PM	6/6/2025 12:41:21 PM	OSDAI_edAdmin	Error Logs	Completed

Scan Report

From the Completed Jobs page, your downloaded report is saved in File Explorer.

When selecting and downloading a report, it opens as an Excel File (.xlsx file). A

Scan Report contains the following information:

- **Condition Name:** The unique name of the condition.
- **Scan Item Type:** The name of the type of scan the condition has run.
- **Scan Item:** The type of scan the condition has run.
- **Line Number Start:** The line of code where the scan detected the condition it was searching for.
- **Line Number End:** The final line of code where the scan detected the condition it was searching for.
- **Severity Level:** The severity of the condition.
- **Explanation:** Shows a detailed explanation of what the scan has picked up from searching the code it was applied to.

Completed Jobs

- **Suggested Fix:** Contains a detailed plan that you can perform to eliminate the errors in the code that the scan detected in the application where the condition was applied. Follow the steps outlined by the suggested fix to promote efficiency in the environment.

Scan Report Example

	A	B	C	D	E	F	G	H	I	J
		condition_name	scan_item_type	scan_item	line_number_start	line_number_end	severity_level	explanation	suggested_fix	
2	0	Error Logs	Business Rule	AISXDT_ParamHelper	174	177	Low	The code logs an error using BRApi.LogError.LogMessage. Refactor the code so that error logging occurs only within a catch block.		
3	1	Error Logs	Business Rule	AISXDT_SolutionHelper	1637	1654	Low	The method WriteQueryToLog uses an error logging call. Remove or conditionally execute the log call in WriteQueryToLog.		
4	2	Error Logs	Business Rule	AISXDT_SolutionHelper	2780	2780	Low	The error log on line 2780 is used for standard logging output. Remove or limit the error logging if not essential for monitoring.		
5	3	Error Logs	Business Rule	AISXDT_SolutionHelper	5243	5243	Low	The call to LogToOneStreamErrorLog on line 5243 is used. Remove the LogToOneStreamErrorLog call or refactor it to use a standard logging method.		
6	4	Error Logs	Business Rule	AISXDT_SolutionHelper	5244	5244	Low	The call to LogToOneStreamErrorLog on line 5244 is executed. Remove the LogToOneStreamErrorLog call or modify it to use a standard logging method.		
7	5	Error Logs	Business Rule	AISXDT_SolutionHelper	7330	7357	Low	The log call at line 7353 uses the error logging function. Wrap the logging call with a condition to only log when necessary.		
8	6	Error Logs	Business Rule	AISXDT_SolutionHelper	7360	7414	Low	At line 7372, the code logs a message about data table path. Consider either removing this log statement or wrapping it in a conditional statement.		
9	7	Error Logs	Business Rule	AISXDT_SolutionHelper	7427	7447	Low	The log call at line 7439 in the ListExt module is executed. Modify the logging to be conditional upon a debug configuration.		
10	8	Error Logs	Business Rule	TestVB	41	41	Low	The logging statement 'Brapi.LogError.LogMessage' is executed. Remove or conditionally execute the log statement if it is necessary.		
11	9	Error Logs	Business Rule	AISXFU - Setup.cs	269	269	Low	The call to BRApi.LogError.LogMessage is made outside of a catch block. Remove the error log call from this section or change it to be conditional.		
12	10	Error Logs	Business Rule	XBR - DatabaseCommon	950	972	Low	The WriteQueryToLog method logs detailed query information. Review if the logging in WriteQueryToLog is necessary; otherwise, remove it.		
13	11	Error Logs	Business Rule	XBR - DataTableExtensio	140	140	Low	The use of BRApi.LogError.LogMessage outside a catch block. Replace the error log in LogToOneStreamErrorLog with a standard logging method.		

- **Condition Name:** Error Log
- **Scan Item Type:** Business Rule
- **Scan Item:** AISXDT_ParamHelper
- **Line Number Start:** 174
- **Line Number End:** 177
- **Severity Level:** Low
- **Explanation:** The code logs an error using BRApi.LogError.LogMessage outside of a catch block, which may lead to excessive and unnecessary logging. This can increase system load and make it harder to identify truly critical errors.
- **Suggested Fix:** Refactor the code so that error logging occurs only within catch blocks or remove the logging call if it's not providing meaningful debugging information. Alternatively, use conditional logging to ensure that only significant errors are recorded.

Conditions

On the Conditions Page you can view Predefined Conditions, or create and edit Custom Conditions. All conditions can be viewed from the Conditions Table. There are a total of 38 Predefined Conditions, and the total number of conditions, including Custom can also be found here.

Conditions, either Predefined or Custom, will not scan OneStream Solutions and Partner Solutions. These are whitelisted.


On this page you can:

- **Create** a Custom conditions.
- **Edit** a Custom condition.
- **Delete** a Custom condition. This action cannot be undone.


NOTE: Predefined conditions cannot be deleted.

- Clear cache by selecting **Refresh**. The Conditions Table and the number of conditions refreshes.
- Filter conditions with the **Scan Item Type** drop-down menu.


Conditions (55)




Create



Edit



Delete



Refresh

Scan Item Type

Conditions Table

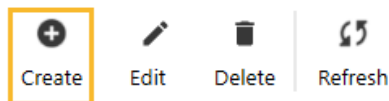
- **Condition Name:** The uniquely assigned name of the condition
- **Condition Description:** A user-legible definition of the condition. This will appear on the Home page for users to select for future scans.
- **Natural Language Definition:** The prompt informs the Large Language Model on how the condition acts once created, and how it scans code. This additional context will help aide the LLM in determining what to look for. It is best to be as specific as possible with the NLD, describing why the coding practice that the condition will look for should be avoided.
- **Condition Type:** Defines whether a condition is either Custom or Predefined
- **Scan Item Type:** Displays the type of scan the condition will run, Business Rule, Data Adapter, or All
- **Severity Level:** Shows the severity of a condition, either High, Medium or Low
- **Refresh:** Use to clear cache. The Conditions Table and the number of conditions refreshes.

Condition Name	Condition Description	Natural Language Definition	Condition Type	Scan Item Type	Context Source	Severity Level
Error Logs	Error logging should be used only for meaningful debugging. Excessive or unnecessary logging increases system load and makes critical errors harder to identify.	Unnecessary error logging increases system load and makes debugging harder by adding too much noise to logs, which can hide important errors. Instead, logs should be used only for meaningful debugging and be removed when no longer needed. Note: An error log inside of a "Catch", such as <code>throw ErrorHandler.LogWrite(s, new XException(s, ex))</code> , is safe. Do not flag this.	Predefined	Business Rule	Business Rule API Documentation	Low
String-Based Member Re	Using string literals for member names in business rules can lead to performance issues and increase the risk of runtime errors due to typos or member name changes. Instead, use integer member IDs, which are faster to evaluate and more reliable.	String literals representing member names in business rules should be avoided. Using strings can be error-prone and less efficient, especially when member names change or contain typos. Instead, reference members by their integer member IDs, which are more performant and reduce risk in production environments.	Predefined	Business Rule	Business Rule API Documentation	Low
Nested For Loops	Excessive nesting of 'For' loops increases time complexity, leading to high CPU usage and performance degradation, especially for large datasets. Refactoring with optimized algorithms or breaking down loops using helper functions can improve efficiency.	Nested For loops significantly slow down execution by increasing time complexity, often leading to exponential performance issues. This can cause unnecessary CPU usage, especially when handling large datasets. Instead, consider alternative approaches such as using caching, recursion, LINQ, or precomputing reusable values to improve efficiency.	Predefined	Business Rule	Business Rule API Documentation	High

IMPORTANT: You cannot create or edit Custom Conditions while a scan is running. This will avoid a concurrency error.

Create a New Condition

Conditions (55)



1. To add a new condition, click the **Create** button to be guided through the AI Component Workflow.
2. In the **Condition Name** text box, enter a unique name for your new condition.

NOTE: Names of conditions are unable to be edited later.

3. Enter a description in the **Condition Description** text box of what the bad practice in the code is that the scan trying to catch.
4. In the **Code Example** text box, enter a short code snippet of the bad practice described in the Condition Description. This field is optional, but will highly aid in accuracy of code scan results if used.
5. The **Scan Item Type** will determine what types of OneStream artifacts will be scanned for this condition. Use the drop-down menu to choose one of the following scan item types:
 - All
 - Business Rule
 - Data Adapter
6. In the **Natural Language Definition** text box, enter a prompt to inform the model of the condition requirements. The more detailed and specific a prompt is will result in a narrow and efficient scope when the condition scans the code.

Example: NLD prompt for an Error Log condition:

Unnecessary logging with `Brapi.ErrorLog.LogMessage` increases system load and makes debugging harder by adding too much noise to logs, which can hide important errors. Instead, logs should be used only for meaningful debugging and be removed when no longer needed.

7. From the **Severity Level** drop-down menu, select the severity of your condition:
 - Low
 - Medium
 - High
8. Some conditions need additional context apart from face-value code to determine if it meets the criteria for being a bad practice. Context should be used only when necessary to increase accuracy in scans. If you specify unnecessary context to be included, it can lead to confusion and inconsistent results in the model. Use the **Context Sources (optional)** drop-down to select the parameter that the condition will be applied to:
 - Business Rule API Documentation
 - Database Table Schema
 - Application Metadata
9. Click **Create** and your saved condition will appear in the Conditions Table.

NOTE: If you click **Cancel**, the AI Component Workflow field closes without saving any changes.

Conditions

New Condition

Condition Name:

Use of Custom Tables in Finance Rules

Condition Description:

Interacting with custom database tables from within a Finance business rule can create additional SQL connections, slow overall system throughput, and greatly increase the likelihood of dead-locks. Such data-layer activity should be handled in Data Adaptors or ETL scripts—not inside calculation logic.

Code Example (optional):

BRApi.Database.ExecuteSql(dbConn, sql, true)

Scan Item Type:

Business Rule

Natural Language Definition:

Any Finance rule or member formula that calls a method in the BRApi.Database namespace which reads or writes tables must be flagged. This includes: ExecuteSQL, ExecuteSQLUsingReader, ExecuteActionQuery, GetDataTable, GetCustomDataTable, SaveCustomDataTable, SaveDataTableRows, and InsertOrUpdateRow. These calls execute SQL directly during calculation, leading to performance bottlenecks and potential dead-locks.

Severity Level:

High

Context Sources (optional):


Business Rule API Documentation


Cancel


Create


Edit a Custom Condition

Conditions (55)

 Create

 Edit

 Delete

 Refresh

To edit a Custom Condition, select the condition and click the **Edit** button and the AI Component Workflow will open.

You can make changes to the following:

Conditions

- **Condition Description:** Edit the description describing the bad practice that informs how the condition will act
- **Code Example (optional):** Input a different example of code that the scan will look for
- **Scan Item Type:** Change the condition type
- **Natural Language Definition:** Change the prompt that sets the model requirements and informs how the condition will act and scan code
- **Severity Level:** Select the severity level of your condition
- **Context Sources (optional):** From the drop-down menu, can change the parameter that the condition will be applied to

Click the **Update** button and the Conditions Table will refresh with your saved changes. Clicking **Cancel** will cause the AI Component Workflow to close without saving any changes.

NOTE: You are unable to edit Predefined Conditions.

Help and Miscellaneous Information



This page contains solution documentation.

Display Settings

OneStream Solutions frequently require the display of multiple data elements for proper data entry and analysis. Therefore, the recommended screen resolution is a minimum of 1920 x 1080 for optimal rendering of forms and reports.

Additionally, OneStream recommends that you adjust the Windows System Display text setting to 100% and do not apply any custom scaling options.

Package Contents and Naming Conventions

The package file name contains multiple identifiers that correspond with the platform. Renaming any of the elements contained in a package is discouraged in order to preserve the integrity of the naming conventions.

Example Package Name: OSD_PV8.5.0_SV100_PackageContents.zip

Identifier	Description
OSD	Solution ID
PV9.0.0	Minimum Platform version required to run solution

Identifier	Description
SV100	Solution version
PackageContents	File name

OneStream Solution Modification

Considerations

A few cautions and considerations regarding the modification of OneStream Solutions:

- Major changes to business rules or custom tables within a OneStream Solution will not be supported through normal channels as the resulting solution is significantly different from the core solution.
- If changes are made to any dashboard object or business rule, consider renaming it or copying it to a new object first. This is important because if there is an upgrade to the OneStream Solution in the future and the customer applies the upgrade, this will overlay and wipe out the changes. This also applies when updating any of the standard reports and dashboards.
- If modifications are made to a OneStream solution, upgrading to later versions will be more complex depending on the degree of customization. Simple changes such as changing a logo or colors on a dashboard do not impact upgrades significantly. Making changes to the custom database tables and business rules, which should be avoided, will make an upgrade even more complicated.