# CONTENTS

# ONESTREAM & BUSINESS RULES – UPGRADING TO VERSION 8.0

Overview: There are several substantial changes as part of 8.0 that can impact business rules and custom solutions. These are being driven by

- The introduction of workspaces, associated security roles and XF Project changes
- The expansion of locations where cube views can be created, managed and stored
- Transitioning the business rule compiler from .NET Framework 4.8 to .NET6 (.NET Core)

The following document represents the changes and lessons learned made by OneStream MarketPlace while addressing these changes for the ~50 solutions maintained on Solution Exchange.

## WORKSPACES

A workspace is a framework for building software using software, creating a robust environment for developing products on the platform. It simplifies the development process and extends development capabilities for solution creators.

Workspaces store maintenance units and facilitate community development by providing an isolated environment for developers to segregate and organize solution objects. Maintenance units are stored, created, and maintained in workspaces, which vary by dashboard project need and application.

Workspaces represent a necessary evolution in how solutions are structured and managed in OneStream. More information on Workspaces can be found in the online OneStream Documentation.

### WORKSPACE ID

The Workspace ID is used to explicitly identify the Workspace that Dashboards, Components, Data Adapters, Parameters, Files etc. belong to. Many functions in the BRApi and Platform Classes have been enhanced to include a workspace ID in the function signature and existing function signatures have been marked as obsolete. The Default Workspace ID is Guid.Empty and can be referenced by SharedConstants.DefaultWorkspaceID.

There are multiple ways to obtain the Workspace ID.

- BRApi.Dashboards.Workspaces.GetWorkspaceIDFromName
    - By providing the Workspace name this function will return the Workspace ID or SharedConstants.UnknownWorkspaceID if not found.
- BRApi.Dashboards.Workspaces.GetWorkspace
    - By Providing the Workspace name this function will return the DashboardWorkspace object that contains the WorkspaceID
- BusinessRule DashboardExtender DashboardExtenderArgs
    - PrimaryDashboard.WorkspaceID is the outer dashboard and best way to identify the solution workspace
    - EmbeddedDashboard.WorkspaceID

## OBSOLETE FUNCTION WARNINGS

Multiple functions in the BRApi have changed to include a WorkspaceID and previous function signatures without a WorkspaceID have been marked as obsolete. Obsolete functions will utilize the default WorkspaceID. It's not uncommon to see many of these warnings post-upgrade and modifications to utilize the WorkspaceID from methods referenced in the section above should be performed for solutions supporting Platform 8.0.

1) Warning at line 202: 'Function GetLiteralParameterValue(si As SessionInfo, isSystemLevel As Boolean, parameterName As String) As String' is obsolete: 'This is a temporary function used by Marketplace Solutions for backwards compatibility with old XF versions. Please change your code to use supported functionality.'.

```
BRApi.Dashboards.Parameters.GetLiteralParameterValue(si, false, args.PrimaryDashboard.WorkspaceID, "[YOUR DASHBOARD PARAMETER NAME GOES HERE]"
```

## XF PROJECT / EXTRACTED XML CHANGES

XF Project extract is for application project designers who are building solutions that span many artifacts, such as workspaces, dashboard maintenance units, business rules, cubes, dimensions, cube views, and other artifacts. The application Extract and Load option collects defined objects, such as dashboards and business rules, as a single file export package that can be reloaded as a package.

XF Project is a convenient way to organize workspaces, data maintenance units, or similar solutions into a folder structure that can be integrated with a version control system, such as Git. Developers must create an XML file that is the definition for the contents of the project export.

More information on XF Project files can be found in the online [OneStream Documentation.]

Prior to workspaces, XF Project files would only need to define a Project Item Type of "DashboardMaintenanceUnit" and set Include Descendants = "true", which would bundle all the child items under a maintenance unit into the exported result. With workspaces, XF Project files will now need to define a Project Item Type of "DashboardWorkspace" and set Include Descendants = "true", which will bundle all the child items under a Workspace into the exported result. If the solution is contained within a Workspace other than the Default workspace the following update can be used:

### PREVIOUS

```
<projectItem projectItemType="DashboardMaintenanceUnit" folderPath="" name="XFW Help Desk (HDK)" includeDescendants="true" />
```

### NEW

```
<projectItem projectItemType="DashboardWorkspace" folderPath="" name="Help Desk" includeDescendants="true" />
```

For solutions installing into the Default Workspace then it will be necessary to add the DashboardWorkspace project item type for the Default Workspace and set includeDescendants to 'false' as well as updating the DashboardMaintenanceUnit to include the workspace="Default" as shown below.

### PREVIOUS

```
<projectItem projectItemType="DashboardMaintenanceUnit" folderPath="" name="XFW Help Desk (HDK)" includeDescendants="true" />
```

### NEW

```
<projectItem projectItemType="DashboardWorkspace" folderPath="" name="Default" includeDescendants="false" />

<projectItem projectItemType="DashboardMaintenanceUnit" folderPath="" workspace="Default" name="XFW Help Desk (HDK)" includeDescendants="true" />
```

**NOTE:** The NEW item is added because the maintenance unit requires a workspace to be loaded into.

Many of the projectItem types now require the 'workspace' attribute to ensure accuracy. For all items located under a Dashboard Maintenance Unit add the 'workspace' attribute after the 'folderPath' for the following:

- CubeViewGroup
- CubeView
- DashboardMaintenanceUnit
- DashboardFile
- DashboardString
- DashboardParameter
- DashboardGroup
- DashboardAdapter
- DashboardComponent
- Dashboard
- DashboardManagementGroup
- DashboardManagementSequence
- WorkspaceAssembly

## EXTRACTED XML

Extracted XML Changes

Prior to workspaces, extracting a solution into a zip file would result in the generation of an *ApplicationDashboards.xml* file, as well as other XML files depending on the solution requirements.

With workspaces, extracting a solution into a zip file will now result in the generation of an *ApplicationWorkspaces.xml* file rather than *ApplicationDashboards.xml* file. This file consists of a slightly different structure than the prior iteration.

Here is an example of the XML structure changes. In this example, attributes on XML nodes have been removed. A few notable changes are:

1. The root XML node of the file has changes from *applicationDashboardsRoot* to *applicationWorkspacesRoot*
2. The *maintenanceUnits* node has been moved inside of the new *workspaces* node
3. There is a new *workspaceAssemblies* node, if your application contains Workspace Assemblies
4. There is a new *cubeViewGroups* node, if your application contains Cube View Groups inside of its workspace
5. There is a new *cubeViewProfiles* node, if your application contains a Cube View Profile associated with the Cube View Group in your workspace

## APPLICATIONDASHBOARDS.XML

```
<?xml version="1.0" encoding="utf-8"?>
<OneStreamXF version="x.x.x...">
  <applicationDashboardsRoot>
    <businessRules> ... </businessRules>
    <maintenanceUnits>
      <maintenanceUnit>
        <fileResources> ... </fileResources>
        <components> ... </components>
        <dashboardGroups>
          <dashboardGroup> ... </dashboardGroup>
        </dashboardGroups>
      </maintenanceUnit>
    </maintenanceUnits>
  </applicationDashboardsRoot>
</OneStreamXF>
```

### APPLICATIONWORKSPACES.XML EXAMPLE

```
<?xml version="1.0" encoding="utf-8"?>
<OneStreamXF version="x.x.x...">
  <!--New root-->
  <applicationWorkspacesRoot>
    <businessRules> ... </businessRules>
    <!--New workspaces and workspace node-->
    <workspaces>
      <workspace>
        <maintenanceUnits>
          <maintenanceUnit>
            <fileResources> ... </fileResources>
            <components> ... </components>
            <dashboardGroups>
              <dashboardGroup> ... </dashboardGroup>
            </dashboardGroups>
            <!--New workspaceAssemblies node-->
            <workspaceAssemblies> ... </workspaceAssemblies>
            <!--New cubeViewGroups node-->
            <cubeViewGroups>
              <cubeViewGroup>
                <cubeViewItems> ... </cubeViewItems>
              </cubeViewGroup>
            </cubeViewGroups>
          </maintenanceUnit>
        </maintenanceUnits>
      </workspace>
    </workspaces>
    <!--New cubeViewProfiles node-->
    <cubeViewProfiles>
      <cubeViewProfile> ... </cubeViewProfile>
    </cubeViewProfiles>
  </applicationWorkspacesRoot>
</OneStreamXF>
```

## SECURITY CHANGES

### ROLE NAME CHANGES

As part of renaming dashboards to workspaces roles have been added and certain roles have had their names changed. Changes have been made to RoleTypeId as well as RoleTypes. The RoleType GetItem(string name) function has been updated for backward compatibility to return “workspace” role types for the following names {ManageApplicationDashboards, ManageSystemDashboards, DashboardAdminPage, SystemDashboardAdminPage}.

| | Platform Less Than 8.0.0 | Platform 8.0.0 Or Greater |
|---|---|---|
| **Level** | **RoleTypeId enumeration** | |
| System | N/A | AdministerSystemWorkspaceAssemblies |
| System | ManageSystemDashboards | ManageSystemWorkspaces |
| System-UI | SystemDashboardAdminPage | SystemWorkspaceAdminPage |
| Application | N/A | AdministerApplicationWorkspaceAssemblies |
| Application | N/A | ManageSmartIntegration |
| Application | ManageApplicationDashboards | ManageApplicationWorkspaces |
| Application-UI | DashboardAdminPage | WorkspaceAdminPage |
| | **RoleTypes** | |
| System | N/A | AdministerSystemWorkspaceAssemblies |
| System | ManageSystemDashboards | ManageSystemWorkspaces |
| System | N/A | ManageIdentityProviders |
| System-UI | SystemDashboardAdminPage | SystemWorkspaceAdminPage |
| Application | N/A | AdministerApplicationWorkspaceAssemblies |
| Application | ManageApplicationDashboards | ManageApplicationWorkspaces |
| Application-UI | DashboardAdminPage | WorkspaceAdminPage |

## DASHBOARD GROUP SECURITY SETTING CHANGE

The dashboard group security setting has been moved to the maintenance group level. This is true even for existing solutions that are automatically being migrated into the Default workspace.

8.0 – Security Moved to Maintenance Unit

| General (Maintenance Unit) | |
|---|---|
| Name | Application Control Manager Installer (ACMI) |
| Workspace | Default |
| Description | Application Control Manager Installer |
| Is Mobile | False |
| Assemblies | |
| Workspace Assembly Service | |
| Security | |
| Access Group | Everyone |
| Maintenance Group | Everyone |

| General (Dashboard Group) | |
|---|---|
| Name | Main Workspace Dialogs (ACMI) |
| Workspace | Default |
| Maintenance Unit | Application Control Manager Installer (ACMI) |
| Description | |

## PROGRAMMATIC EXTRACT AND LOAD FUNCTIONALITY

The shift in the hierarchy that came about by introducing workspaces as the parent of a solution also impacted programmatic extract and load functionality.

## EXTRACTXFPROJECTZIPFILE

In 8.0, the 'ExtractXFProjectZipFile' functionality has been moved to the Framework exclusively, which means it is no longer a call from XFProjectWcf object. Correct this by defining a Framework object and issuing the call from there.

**Pre 8.0**

```
editedDataRow.Item("Archive") = XFProjectWcf.ExtractXFProjectZipFile(dbconnfw, dbconnApp, XfProjObj)
```

**Post 8.0**

```
Dim fw As IFramework = New Framework()
editedDataRow.Item("Archive") = fw.ExtractXFProjectZipFile(si, XfProjObj)
```

## EXTRACTXML FOR DASHBOARDS

Extracting dashboards now requires reference to workspaces rather than dashboards. ApplicationDashboards becomes ApplicationWorkspaces and DashboardsXmlExtractItemHierarchy becomes WorkspacesXmlExtractItemHierarchy

**Pre 8.0**

```
var dashboardsHier = DashboardsXmlExtractItemHierarchy.GetHierarchy(dbConnFW, dbConnApp, false);
AddPksToDictionary(si, ref dictDashboards, dashboardsHier, solutionCode, false);
string dashboardsXmlExtract = XmlExtractController.ExtractXml(dbConnFW, dbConnApp, exWcfClient, xmloptions,
dictDashboards, XmlLoadExtractType.ApplicationDashboards);
```

**Post 8.0**

```
var dashboardsHier = WorkspacesXmlExtractItemHierarchy.GetHierarchy(dbConnFW, dbConnApp, false);
AddPksToDictionary(si, ref dictDashboards, dashboardsHier, solutionCode, false);
string dashboardsXmlExtract = XmlExtractController.ExtractXml(dbConnFW, dbConnApp, exWcfClient, xmloptions,
dictDashboards, XmlLoadExtractType.ApplicationWorkspaces);
```

## ENUMERATION CHANGES

The OneStream.Shared.Wcf.RoleTypeId Enumeration replaced the DashboardAdminPage and SystemDashboardAdminPage members with new WorkspaceAdminPage and SystemWorkspaceAdminPage members, respectively.

**Pre 8.0**

```
If BRApi.Security.Authorization.IsUserInRole(si, RoleTypeId.DashboardAdminPage) Then
GeneralHelpers.WriteNameValuePairRow(si, dt, "DashboardAdmin", "DashboardAdmin")
If BRApi.Security.Authorization.IsUserInRole(si, RoleTypeId.SystemDashboardAdminPage) Then
GeneralHelpers.WriteNameValuePairRow(si, dt, "SystemDashboardAdmin", "SystemDashboardAdmin")
```

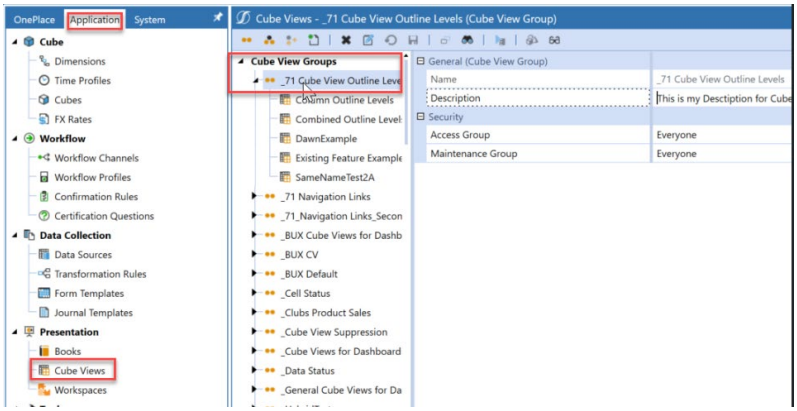
**Post 8.0**

```
If BRApi.Security.Authorization.IsUserInRole(si, RoleTypeId.WorkspaceAdminPage) Then
GeneralHelpers.WriteNameValuePairRow(si, dt, "WorkspaceAdmin", "WorkspaceAdmin")
If BRApi.Security.Authorization.IsUserInRole(si, RoleTypeId.SystemWorkspaceAdminPage) Then
GeneralHelpers.WriteNameValuePairRow(si, dt, "SystemWorkspaceAdmin", "SystemWorkspaceAdmin")
```

## CUBE VIEWS

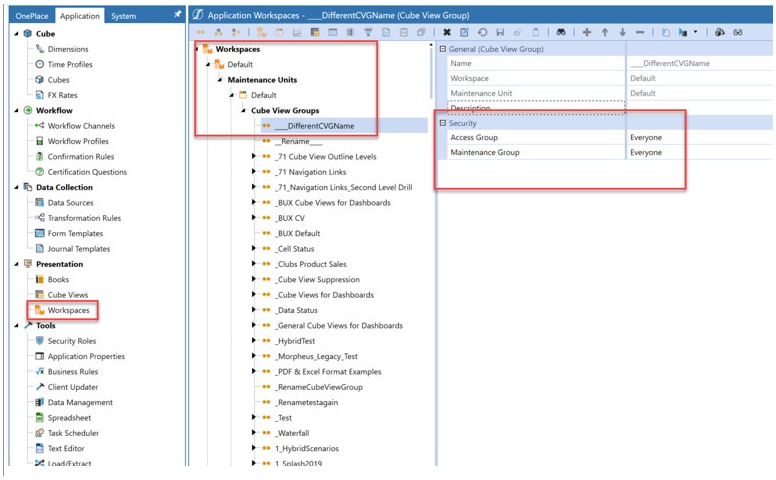
### CUBE VIEW LOCATIONS AND SECURITY

Cube views are now available at several different locations in the application and have different properties available depending on that location.



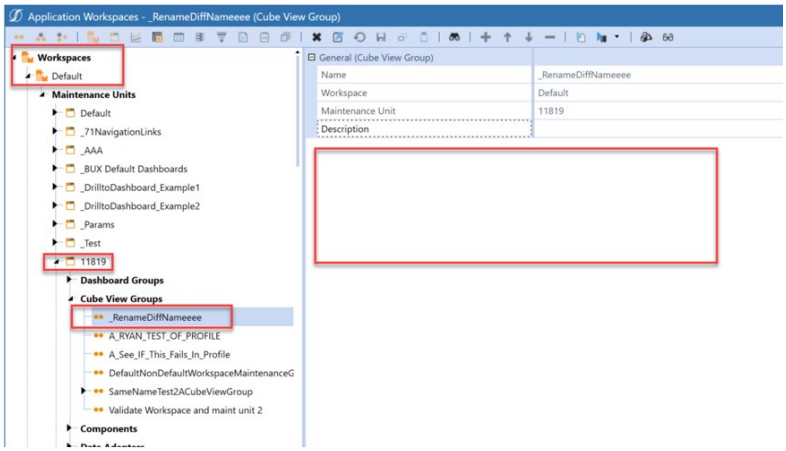

In the pre-8.0 OneStream instances Cube Views were available under the presentation layer as Cube View Groups.

Cube View Groups within this location will have the security properties of *Access Group* and *Maintenance Group* to be editable by the user if the user has the ManageCubeViews role. Otherwise, these properties are read-only.



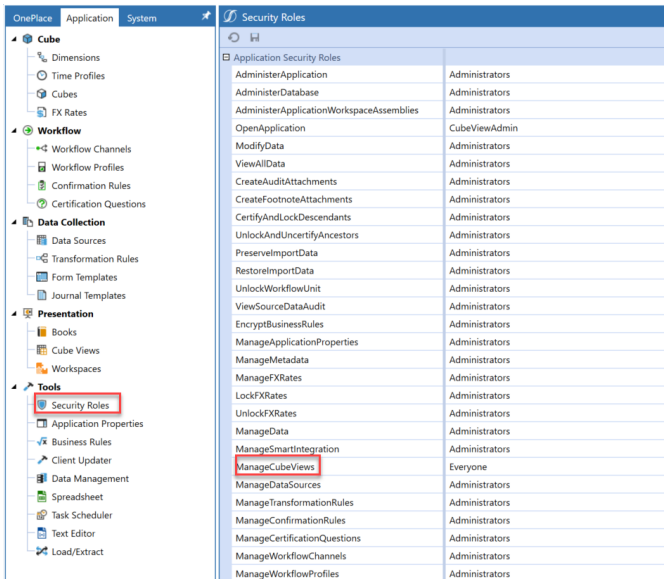

Cube View Groups under workspaces as seen here at Default Workspace/Default Maintenance Unit. These Cube View Groups are the same as those in the pre-8.0 OneStream instances except for their location.

Cube View Groups within this location will have the security properties of *Access Group* and *Maintenance Group* to be editable by the user if the user has the ManageCubeViews role. Otherwise, these properties are read-only.



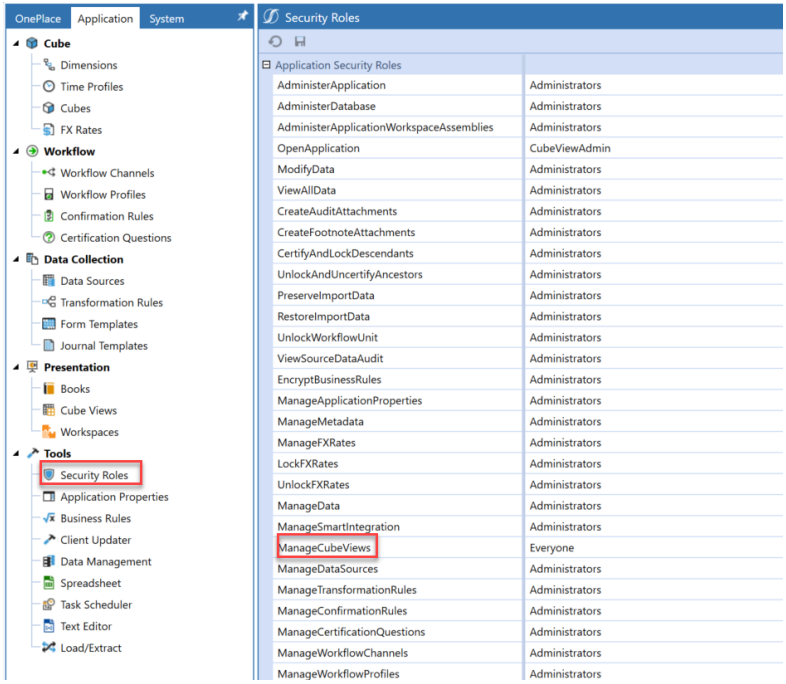

Cube View Groups outside the workspace/default maintenance unit will have the security properties of *Access Group* and *Maintenance Group* completely hidden.

## ADMINISTERING CUBE VIEWS




The *ManageCubeViews* Security Role will give full rights to edit Cube View Groups in the legacy cube views and default workspace/ default maintenance unit cube view groups.

Users with *ManageCubeViews* role can edit Cube Views in Legacy Cube Views but will also need to have the *WorkspaceAdminPage* role to see and edit those same Cube Views in the Workspace page. Edits made in the Presentation/Cube Views will automatically update the Cube Views in the Default Workspace/Default Maintenance Unit and vice versa.




The application security role ManageApplicationWorkspaces will give full rights to make edits everywhere else outside the default workspace/default maintenance unit.

This includes:

- Dashboard Groups
- Cube View Groups
- Components
- Data Adapters
- Parameters
- Files
- String
- Assemblies (also requires the AdministerApplicationWorkspaceAssemblies role)

## CUBE VIEWS IN CODE

Many cube view instantiation calls (such as for CubeViewGroup or CubeViewItem) now have an additional parameter, Workspace ID. If a cube view is part of the Default workspace, the ID is SharedConstants.DefaultWorkspaceID. If the cube view is part of another workspace, that workspace ID can be obtained by the following call:

```
Dim newCubeViewGroup As New CubeViewGroup(Guid.NewGuid, groupName, String.Empty, Guid.Empty, Guid.Empty)
```

Would become:

```
Dim gValue As Guid = args.PrimaryDashboard.WorkspaceID (See WorkspaceID above)

Dim newCubeViewGroup As New CubeViewGroup(Guid.NewGuid, groupName, gValue, String.Empty, Guid.Empty, Guid.Empty)
```

## .NET6

The introduction of .NET6 brought significant gains in terms of computational power and memory usage for many common OneStream operations, not least among them, the execution of business rules. It also introduced a few potential issues that will need to be addressed.

.NET6 applies to Platform versions 8.0 and 8.1.

## BREAKING CHANGES

### BUSINESS RULES

Platform release 8.0 includes updates and enhancements to the Business Rules compiler for .NET 6 compliance. Administrators must resolve these errors to fully compile business rules. Warning messages identify line items that will function but that you should update to support the latest compiler's requirements. How to resolve these warnings varies. You may be able to use a provided replacement function or change to a function's properties.

**Missing Enum.Parse Overload**

Error: BC30519 - Overload resolution failed because no accessible 'Parse' can be called without a narrowing conversion.
REMEDIATION: Refactor the Business Rule to use .NET Standard 2.0 overload of Enum.Parse method.

**Microsoft.VisualBasic.Devices.Computer**

Error: BC30002 - Type 'Microsoft.VisualBasic.Devices.Computer' is not defined.
REMEDIATION: Refactor Business Rule using the actual computer name obtained from the Environment.MachineName Property.

**Microsoft.VisualBasic.MyServices**

Microsoft.VisualBasic.MyServices namespace is not supported in .NET 6. Error: BC30456 - 'Computer' is not a member of 'My'
REMEDIATION: Refactor Business Rule to use and available .NET 6 alternative.

**Microsoft.VisualBasic.CompilerServices.Conversions.ChangeType**

Error: BC35000 - Requested operation is not available because the runtime library function 'Microsoft.VisualBasic.CompilerServices.Conversions.ChangeType' is not defined.
Example: Pre v8.0 - Using objDT As DataTable = objReport.DataSource.Tables(tableName)
REMEDIATION: Refactor code to reinforce intended type by applying .ToString() method.
Example: v8.0 - Using objDT As DataTable = objReport.DataSource.Tables(tableName.ToString())

**JavaScriptSerializer Class**

JavaScriptSerializer is no longer available in .NET 6. Although OneStream provides an adapter for implementation, this adapter does not provide 100% compatibility with JavaScriptSerializer.
REMEDIATION: It is strongly advised to refactor Business Rules that use JavaScriptSerializer Class to use the System.Text.Json Namespace

**System.Windows.Forms Namespace**

System.Windows.Forms Namespace is no longer available in .NET 6.Remediation: OneStream platform 8.0 includes a stub System.Windows.Forms namespace where the namespace is referenced, not used. In instances in which the System.Windows.Forms namespace is being used, Business Rules will have to be refactored using an alternative method.

**System.Data.Entity is interpreted as a namespace in .NET 6.**

Error: 'System.Data.Entity' is a namespace and cannot be used as an expression. 'Entity' can no longer be used as an expression (ex: entity.Member.Name).
REMEDIATION: Give the variable a unique naming such as 'myentity' as in the following example:

**Pre 8.0**

```
For Each entity In BRApi.Finance.Members.GetMembersUsingFilter(si, BRApi.Finance.Dim.GetDim(si,
"CorpEntity").DimPk, "E#Top", True)
dt.Rows.Add(entity.NameAndDescription, entity.Member.Name)

Next
```

**Post 8.0**

```
For Each myentity In BRApi.Finance.Members.GetMembersUsingFilter(si, BRApi.Finance.Dim.GetDim(si,
"CorpEntity").DimPk, "E#Top", True)
dt.Rows.Add(myentity.NameAndDescription.ToString, myentity.Member.Name.ToString)


Next
```

**Deserialization from CodeDOM report format which is not supported**
Error: "Deserialization from CodeDOM format not supported in .NET Core applications" occurs when reports stored in a legacy binary CodeDOM format are viewed.
REMEDIATION: As part of the overall Platform upgrade, a utility has been provided to update these reports. This utility is to be run just after the DB upgrade. See "Run the Upgrade Assistant Utility" in the Upgrade Guide.

**Type Safety**
Enhanced type safety is now being recognized and enforced by the compiler.
Example: Error: BC30311: Value of type 'Char' cannot be converted to 'Decimal'.
REMEDIATION: Refactor Business Rule to maintain type within variables.

**Cast object of the same Type**

Error: [A]* cannot be cast to [B] from the same compiled assembly.
When executing a business rule that references another business rule, if there is an attempt to cast the type of an object to a class from the referenced business rule using DirectCast or a similar method, the .NET compiler will provide an error similar to the one that follows:
[A]* cannot be cast to [B] from the same compiled assembly.
REMEDIATION: Impacted applications can work around this issue by moving the target classes to a separate, unrelated business rule that is referenced by both of the original business rules

## INFERRED CASTING

One subtle exception that may occur in code is with the casting of objects from one type to another. Certain default castings that users may have taken for granted will now raise exceptions. In most cases, like the two examples below, explicit casting resolves the issue.

**Pre 8.0**

```
Dim result As String = Nothing
If Not inputString = String.Empty Then
    If Integer.TryParse(inputString, result)
```

**Post 8.0**

```
Dim intResult As Integer = 0
Dim decResult As Decimal = 0
If Not inputString = String.Empty Then
    If Integer.TryParse(inputString, intResult)
```

Passing an Object variable as the result parameter of an Int16.TryParse call no longer compiles in Platform 8.0.0 due to .NET 6 not being able to resolve the call to a specific TryParse override.

**Pre 8.0**

```
Dim numberString As String = "5"
Dim numberValue As Object = Nothing
If Not Int16.TryParse(numberString, numberValue) Then
    errors.AppendLine(StringHelper.FormatMessage("numberString not a number", Int16.MinValue, Int16.MaxValue))
End If
```

**Post 8.0**

```
Dim numberString As String = "5"
Dim numberValue As Short = 0
If Not Int16.TryParse(numberString, numberValue) Then
    errors.AppendLine(StringHelper.FormatMessage("numberString not a number", Int16.MinValue, Int16.MaxValue))
End If
```

## PLATFORM 8.0 CONDITIONAL COMPILATION

Beginning in Platform 8.0 new preprocessor symbols have been added to enable conditional compilation of business rules for version specific needs. This allows developers to update versions prior to Platform 8.0.0 that will also work when the OneStream Platform is upgraded. Version constants will be updated and added for each new Platform release.

- ONESTREAM8_0_0
  - Will be present during compilation of business rules on Platform version 8.0.0
- ONESTREAM8_0_0_OR_GREATER
  - Will be present during compilation of business rules on Platform version 8.0 and greater. This means it will be present on 8.0.1, 8.1.0, 8,2.0, etc.

When compiling a business rule on a version lower than Platform 8.0.0 the top of the if block will not be compiled or executed and the bottom block will be.

```
#If ONESTREAM8_0_0_OR_GREATER Then
  If BRApi.Security.Authorization.IsUserInRole(si, RoleTypeId.WorkspaceAdminPage) Then
    GeneralHelpers.WriteNameValuePairRow(si, dt, "WorkspaceAdmin", "WorkspaceAdmin")
  End If
```

```
  If BRApi.Security.Authorization.IsUserInRole(si, RoleTypeId.SystemWorkspaceAdminPage) Then
    GeneralHelpers.WriteNameValuePairRow(si, dt, "SystemWorkspaceAdmin", "SystemWorkspaceAdmin")
  End If
#Else
  If BRApi.Security.Authorization.IsUserInRole(si, RoleTypeId.DashboardAdminPage) Then
    GeneralHelpers.WriteNameValuePairRow(si, dt, "DashboardAdmin", "DashboardAdmin")
  End If

  If BRApi.Security.Authorization.IsUserInRole(si, RoleTypeId.SystemDashboardAdminPage) Then
    GeneralHelpers.WriteNameValuePairRow(si, dt, "SystemDashboardAdmin", "SystemDashboardAdmin")
  End If
#End If
```

When compiling a business rule on a version equal to or greater than Platform 8.0.0 the top of the if block will be compiled and executed and the bottom block will not be.

```
#If ONESTREAM8_0_0_OR_GREATER Then
  If BRApi.Security.Authorization.IsUserInRole(si, RoleTypeId.WorkspaceAdminPage) Then
    GeneralHelpers.WriteNameValuePairRow(si, dt, "WorkspaceAdmin", "WorkspaceAdmin")
  End If

  If BRApi.Security.Authorization.IsUserInRole(si, RoleTypeId.SystemWorkspaceAdminPage) Then
    GeneralHelpers.WriteNameValuePairRow(si, dt, "SystemWorkspaceAdmin", "SystemWorkspaceAdmin")
  End If
#Else
  If BRApi.Security.Authorization.IsUserInRole(si, RoleTypeId.DashboardAdminPage) Then
    GeneralHelpers.WriteNameValuePairRow(si, dt, "DashboardAdmin", "DashboardAdmin")
  End If

  If BRApi.Security.Authorization.IsUserInRole(si, RoleTypeId.SystemDashboardAdminPage) Then
    GeneralHelpers.WriteNameValuePairRow(si, dt, "SystemDashboardAdmin", "SystemDashboardAdmin")
  End If
#End If
```

## .NET8

As with .NET6, the introduction of .NET8 brings additional potential issues that will need to be addressed.

.NET8 applies to Platform version 8.2.

### TYPE SAFETY

The IsMatch method is typically used to validate a string or to ensure that a string conforms to a particular pattern without retrieving that string for subsequent manipulation. If you want to determine whether one or more strings match a regular expression pattern and then retrieve them for subsequent manipulation, call the Match or Matches method.

Prior to .NET8, coding an object as an object and then using the IsMatch method would not cause any exceptions, however by utilizing the recommended method to define objects, casting of those objects should not cause an exception. Below is an example of moving code from pre-Net8 to .NET8.

### Instead of:

Dim objInt as Object = 123

...CType (objInt, String) ...

### Recommended:

Dim strValue as String = "123"

*Casting should be considered only if control of the variable definition is unavailable.