



# BAI Parser

# Release Notes

Copyright © 2025 OneStream Software LLC. All rights reserved.

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

# Table of Contents

BAI Parcer Release PV8.0.0 SV105 .....	1
Fixed Issues .....	1
Upgrade Information .....	1
Documentation .....	1
Contact Support .....	1
Release PV8.0.0 SV104 .....	2
Prerequisites .....	2
Fixed Issues .....	2
Release PV8.0.0 SV103 .....	3
Prerequisites .....	3
Fixed Issues .....	3
Release PV800 SV102 .....	4
Prerequisites .....	4
New Features .....	4

## Table of Contents

---

Enhancements .....	4
Fixed Issues .....	4
Known Issues .....	5
Release PV800 SV101 .....	8
Prerequisites .....	8
New Features .....	8
Enhancements .....	8
Fixed Issues .....	8
Known Issues .....	9
Release PV800 SV100 .....	12
Prerequisites .....	12
Known Issues .....	13

# BAI Parcer Release PV8.0.0 SV105

The theme of this Hotfix release is bug fixes.

## Fixed Issues

- When importing files, they will load successfully into BAI Parser and all data is automatically parsed. (MP3-7896)

## Upgrade Information

- This solution is compatible with OneStream Platform v8.
- This release requires a minimum 8.0 Platform release.

## Documentation

- The solution Help icon now prompts a dialog where you can access help documentation. (MP3-7975)
- The *BAI Parser Guide* has not been updated except to change the release number and copyright if necessary.

## Contact Support

Contact OneStream Support by registering at:

[Support - OneStream Software](#)

# Release PV8.0.0 SV104

The theme of this Hotfix release is bug fixes.

**NOTE:** Only the Release Notes were updated for this release. The BAI Parser Solution Guide has not been updated except to change the release number and copyright if needed.

## Prerequisites

- This solution is compatible with OneStream Platform v8.
- This release requires a minimum 8.0 Platform release.

## Fixed Issues

- The BAI Parser Viewer does not freeze and processes all files accordingly regardless of the fund type of the accounts within a file. (MP3-6887)

# **Release PV8.0.0 SV103**

The theme of this Hotfix release is bug fixes.

**NOTE:** Only the Release Notes were updated for this release. The BAI Parser Solution Guide has not been updated except to change the release number and copyright if needed.

## **Prerequisites**

- This solution is compatible with OneStream Platform v8.
- This release requires a minimum 8.0 Platform release.

## **Fixed Issues**

- The BAI parser will successfully map files with fund type 'D'. All related formatting, including customer references and transaction detail values, will be mapped in the file accordingly. (MP3-6809)

# Release PV800 SV102

The theme of this release are enhancements.

**NOTE:** Only the Release Notes were updated for this release. The BAI Parser Solution Guide has not been updated except to change the release number and copyright if needed.

## Prerequisites

- This solution is compatible with OneStream Platform v8.
- This release requires a minimum 8.0 Platform release.

## New Features

None

## Enhancements

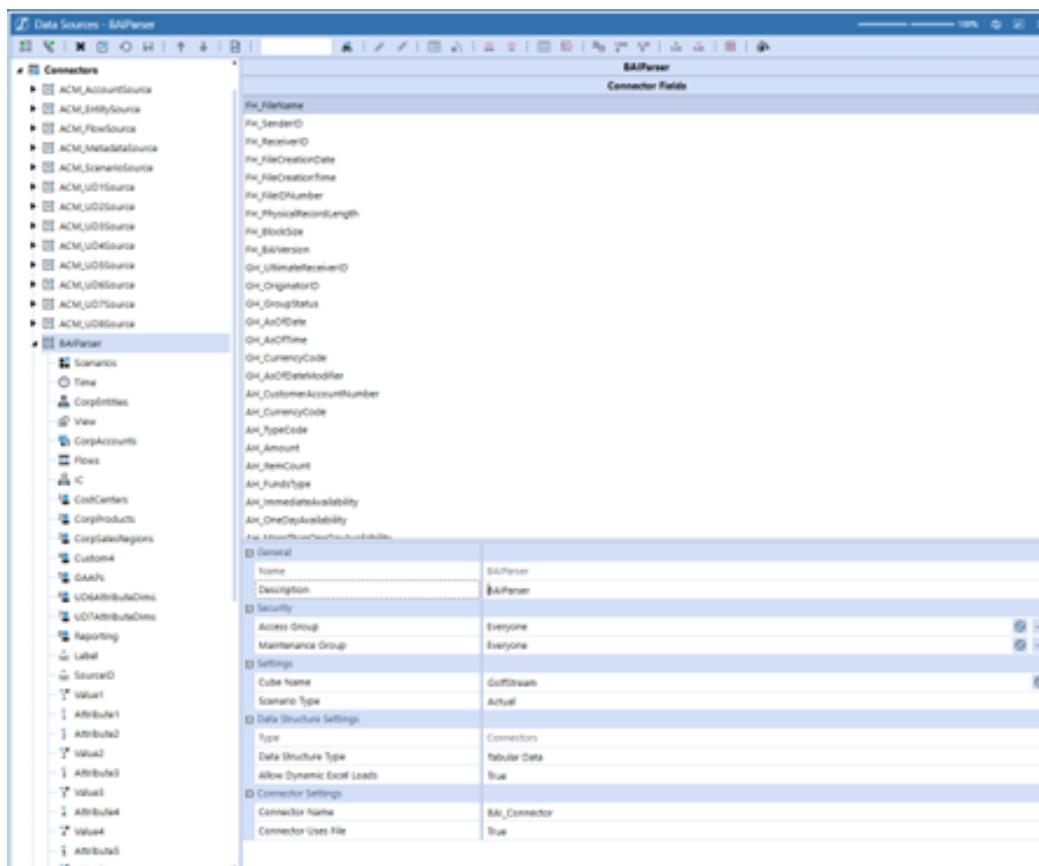
- When imported, files with warnings will still display all available information so you can review your data. (MP3-6350)
- When Account Identifier and Fund Type fields are not valid on imported files, the file will still be parsed with warnings. (MP3-6328, MP3-6348)

## Fixed Issues

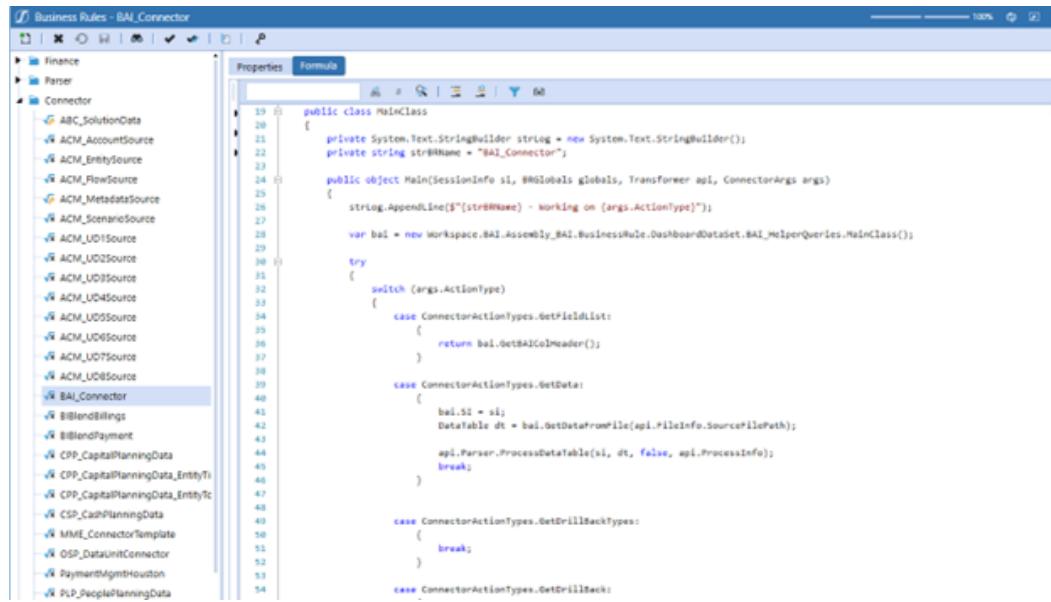
None

## Known Issues

- Automation of the data load is impacted by the Connector Uses File setting. To automate a file, follow the steps below:
  1. Navigate to the Data Source using the BAI Parser Solution.
  2. Set the **Connector Uses File** setting to **False**.



3. Create a copy of the BAI\_Connector Connector Business Rule.
4. Comment out line 42.



```
19 public class MainClass
20 {
21     private System.Text.StringBuilder string = new System.Text.StringBuilder();
22     private string strName = "BAI_Connector";
23
24     public object Main(SessionInfo si, #NGlobals globals, Transformer api, ConnectorArgs args)
25     {
26         string.AppendLine($"[{strName}] - Working on [{args.ActionType}]");
27
28         var bai = new Workspace.BAI.Assembly_BAI.BusinessRule.DashboardDataSet.BAI_HelperQueries.MainClass();
29
30         try
31         {
32             switch (args.ActionType)
33             {
34                 case ConnectorActionTypes.GetFieldList:
35                 {
36                     return bai.GetBAIColHeader();
37                 }
38
39                 case ConnectorActionTypes.GetDate:
40                 {
41                     bai.si = si;
42                     DataTable dt = bai.GetDataFromFile(api.FileInfo.SourceFilePath);
43
44                     api.Parser.ProcessDataTable(si, dt, false, api.ProcessInfo);
45                     break;
46                 }
47
48                 case ConnectorActionTypes.GetDrillbackTypes:
49                 {
50                     break;
51                 }
52
53                 case ConnectorActionTypes.GetDrillback:
54             }
55         }
56     }
57 }
```

**NOTE:** If you have an automated process that moves files to the Harvest Folder, you must add a step that also copies the file to a new folder in the OneStream File Explorer. This step must occur before the connector rule runs, so it is recommended to place this code in an extensibility rule.

5. Add code to the new connector rule that reads the file from your chosen OneStreamFile Explorer folder.

**NOTE:** Below is an example of how you can access the folder structure under a specific workflow name:

```
string IncomingFolder =
    BRApi.Utilities.GetFileShareFolder
    (si, FileShareFolderTypes.ApplicationIncoming,
    api.WorkflowProfile);
```

6. After defining the file path, use this code to process the file:

```
DataTable dt = bai.GetDataFromFile(SourceFilePath);
```

# Release PV800 SV101

The theme of this release is bug fixes.

**NOTE:** Only the Release Notes were updated for this release. The BAI Parser Solution Guide has not been updated except to change the release number and copyright if needed.

## Prerequisites

- This solution is compatible with OneStream Platform v8.
- This release requires a minimum 8.0 Platform release.

## New Features

None

## Enhancements

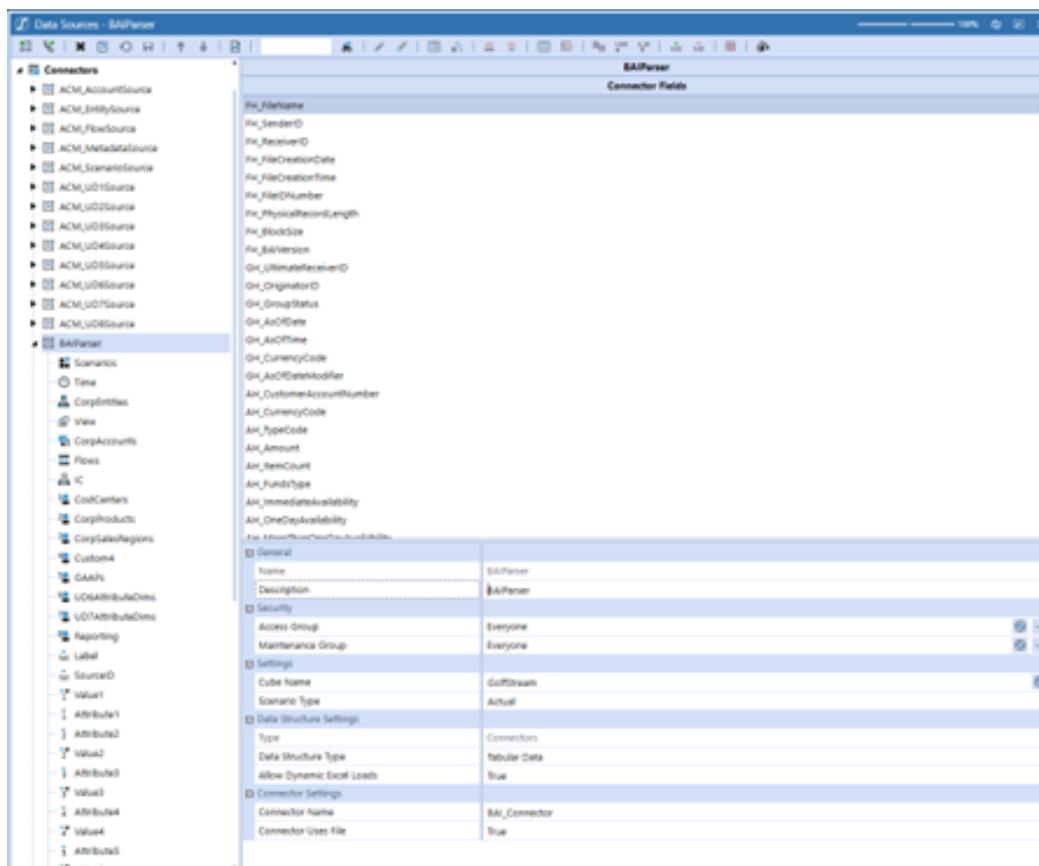
None

## Fixed Issues

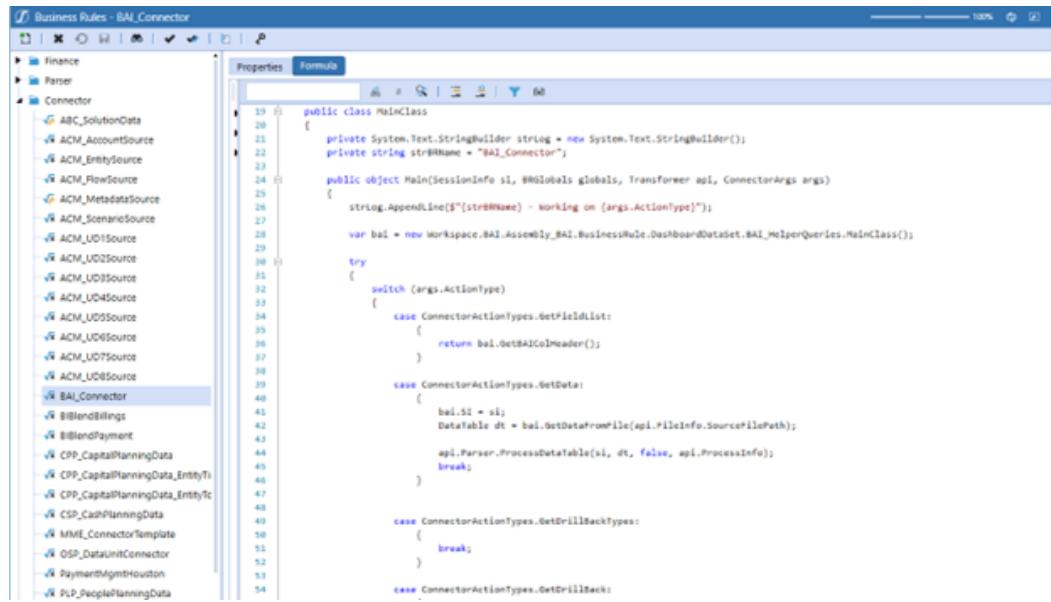
- BAI files uploaded without type codes parse based on all other codes available in the document. (MP3-4736)
- BAI files uploaded with empty, optional fields will parse without error. (MP3-4735)

## Known Issues

- Automation of the data load is impacted by the Connector Uses File setting. To automate a file, follow the steps below:
  1. Navigate to the Data Source using the BAI Parser Solution.
  2. Set the **Connector Uses File** setting to **False**.



3. Create a copy of the BAI\_Connector Connector Business Rule.
4. Comment out line 42.



```
19 public class MainClass
20 {
21     private System.Text.StringBuilder string = new System.Text.StringBuilder();
22     private string strName = "BAI_Connector";
23
24     public object Main(SessionInfo si, #NGlobals globals, Transformer api, ConnectorArgs args)
25     {
26         string.AppendLine($"[{strName}] - Working on [{args.ActionType}]");
27
28         var bai = new Workspace.BAI.Assembly_BAI.BusinessRule.DashboardDataSet.BAI_HelperQueries.MainClass();
29
30         try
31         {
32             switch (args.ActionType)
33             {
34                 case ConnectorActionTypes.GetFieldList:
35                 {
36                     return bai.GetBAIColHeader();
37                 }
38
39                 case ConnectorActionTypes.GetDate:
40                 {
41                     bai.si = si;
42                     DataTable dt = bai.GetDataFromFile(api.FileInfo.SourceFilePath);
43
44                     api.Parser.ProcessDataTable(si, dt, false, api.ProcessInfo);
45                     break;
46                 }
47
48                 case ConnectorActionTypes.GetDrillbackTypes:
49                 {
50                     break;
51                 }
52
53                 case ConnectorActionTypes.GetDrillback:
54             }
55         }
56     }
57 }
```

**NOTE:** If you have an automated process that moves files to the Harvest Folder, you must add a step that also copies the file to a new folder in the OneStream File Explorer. This step must occur before the connector rule runs, so it is recommended to place this code in an extensibility rule.

5. Add code to the new connector rule that reads the file from your chosen OneStreamFile Explorer folder.

**NOTE:** Below is an example of how you can access the folder structure under a specific workflow name:

```
string IncomingFolder =
    BRApi.Utilities.GetFileShareFolder
    (si, FileShareFolderTypes.ApplicationIncoming,
    api.WorkflowProfile);
```

6. After defining the file path, use this code to process the file:

```
DataTable dt = bai.GetDataFromFile(SourceFilePath);
```

# **Release PV800 SV100**

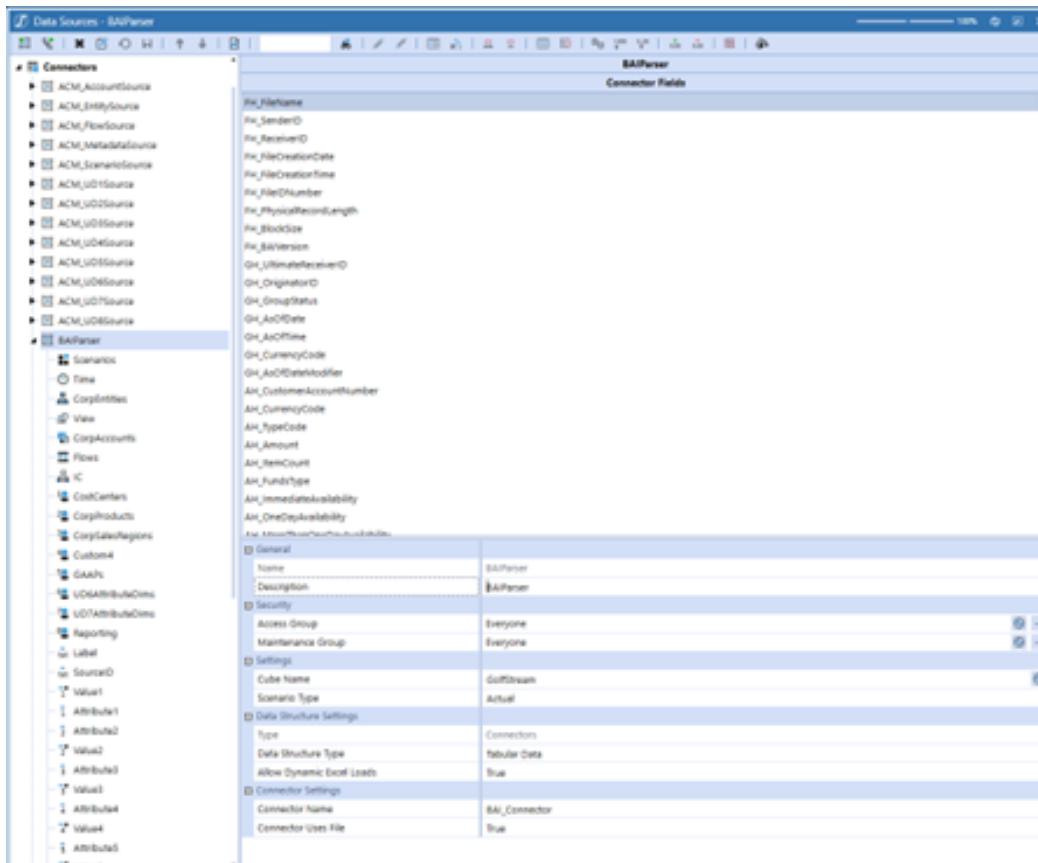
This is the initial release of the BAI Parser Solution and documentation for installation guidance.

## **Prerequisites**

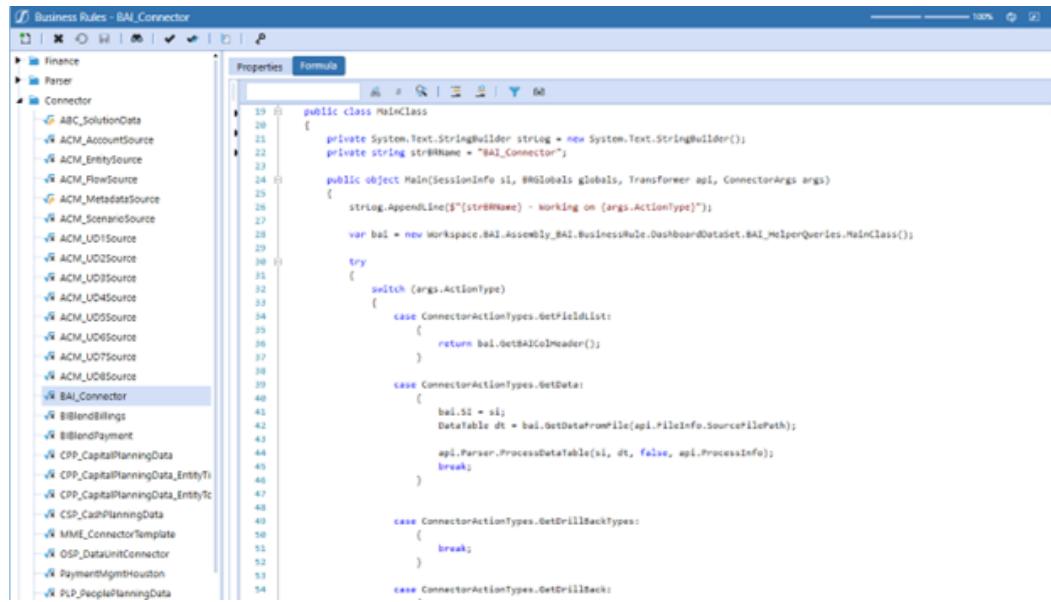
- This solution is compatible with OneStream Platform v8.
- This release requires a minimum 8.0 Platform release.

# Known Issues

- Automation of the data load is impacted by the Connector Uses File setting. To automate a file, follow the steps below:
  1. Navigate to the Data Source using the BAI Parser Solution.
  2. Set the **Connector Uses File** setting to **False**.



3. Create a copy of the BAI\_Connector Connector Business Rule.
4. Comment out line 42.



```
19 public class MainClass
20 {
21     private System.Text.StringBuilder string = new System.Text.StringBuilder();
22     private string strName = "BAI_Connector";
23
24     public object Main(SessionInfo si, #NGlobals globals, Transformer api, ConnectorArgs args)
25     {
26         string.AppendLine($"[{strName}] - Working on [{args.ActionType}]");
27
28         var bai = new workspace.BAI.Assembly_BAI.BusinessRule.DashboardDataSet.BAI_HelperQueries.MainClass();
29
30         try
31         {
32             switch (args.ActionType)
33             {
34                 case ConnectorActionTypes.GetFieldList:
35                 {
36                     return bai.GetBAIColHeader();
37                 }
38
39                 case ConnectorActionTypes.GetDate:
40                 {
41                     bai.SI = si;
42                     DataTable dt = bai.GetDataFromFile(api.FileInfo.SourceFilePath);
43
44                     api.Parser.ProcessDataTable(si, dt, false, api.ProcessInfo);
45                     break;
46                 }
47
48                 case ConnectorActionTypes.GetDrillbackTypes:
49                 {
50                     break;
51                 }
52
53                 case ConnectorActionTypes.GetDrillback:
54             }
55         }
56     }
57 }
```

**NOTE:** If you have an automated process that moves files to the Harvest Folder, you must add a step that also copies the file to a new folder in the OneStream File Explorer. This step must occur before the connector rule runs, so the code needs to be placed in an extensibility rule.

5. Add code to the new connector rule that reads the file from your chosen OneStreamFile Explorer folder.
6. After defining the file path, use this code to process the file:

```
DataTable dt = bai.GetDataFromFile(SourceFilePath);
```

**NOTE:** This is an example of how to access the folder structure under a specific workflow name:

```
string IncomingFolder =  
    BRApi.Utilities.GetFileShareFolder  
    (si, FileShareFolderTypes.ApplicationIncoming,  
    api.WorkflowProfile);
```