Installation and Configuration Guide

7.0.1 Release

Table of Contents

About This Guide

This guide describes how to install and configure the platform. We suggest that these tasks are performed by the Information Technology professional responsible for maintaining and supporting your implementation.

This guide also:

- Identifies the best software configuration for an application's particular requirements.

- Describes how to configure an external Identity Provider so you can use single sign on.

- Provides troubleshooting for common issues.

- Identifies considerations for enhancing performance.

About Upgrading

Refer to the *Upgrade Guide* for information about upgrading to the latest release, and to learn about upgrade considerations, migration, and best practices.

Accessing Related Documentation

Refer to the *System Requirements and Architecture Guide* for information about:

- The platform's architecture and components such as the Web Client, Web Server, Application Sever, and Database Server.

- Third-party software.

- Hardware and software requirements and recommendations.

- Virtual environments.

- Environment configuration guidelines.

The full documentation suite is available in the application or on the MarketPlace.

Infrastructure Requirements and Preparation Checklist

The first step of your installation or upgrade is to contact OneStream Support by registering at: http://support.onestreamsoftware.com. File a ticket for assistance or email support@onestreamsoftware.com.

Use the checklist to prepare your environment before making an installation appointment with My Company Name, LLC Support and review the *System Requirements and Architecture Guide* for:

- Hardware and software requirements, considerations, and configuration best practices for the Web Server, Application Server, Data Server and Client Workstation.

- Minimum environment requirements and system infrastructure guidelines.

Web Server Requirements and Considerations

Ensure that you have:

- Determined the number of web servers.

- Prepared for the Windows OS version that is installed with all updates. Refer to the *System Requirements and Architecture Guide* for supported Windows Server versions.

- Enabled a High Performance Power Plan.

- Installed the .NET Framework. Refer to *System Requirements and Architecture Guide* for supported .NET Framework versions.

- Configured load balancing. Press F5 to specify load balancer settings.

- Disabled any Anti-Virus services. You can re-active anti virus after installation and exclude the OneStream installation directory.

- Discussed and planned for the size of each Web Server and implement the CPU and Memory minimums identified in the *System Requirements and Architecture Guide.*

> **NOTE:** Web Server mostly acts as an intermediary between the client and Application servers, so the number of processors and amount of memory does not need to be as significant as the Application Servers.

- Enabled Network Discovery so all OneStream servers can communicate.

- Implemented the file sharing best practices.

- Emailed OneStream Support so you can download the installation package from the MarketPlace and put the installation package on the desktop or a folder on each server.

- Windows Process Activation Service (WAS).

Installing and Setting up Internet Information Server

You must install a supported version of Internet Information Server (IIS). Refer to the *System Requirements and Architecture Guide*.

After installing, enable the required server roles as follows:

- Go to Add roles and features.

- Select **Next** until you can select **Web Server (IIS)**.
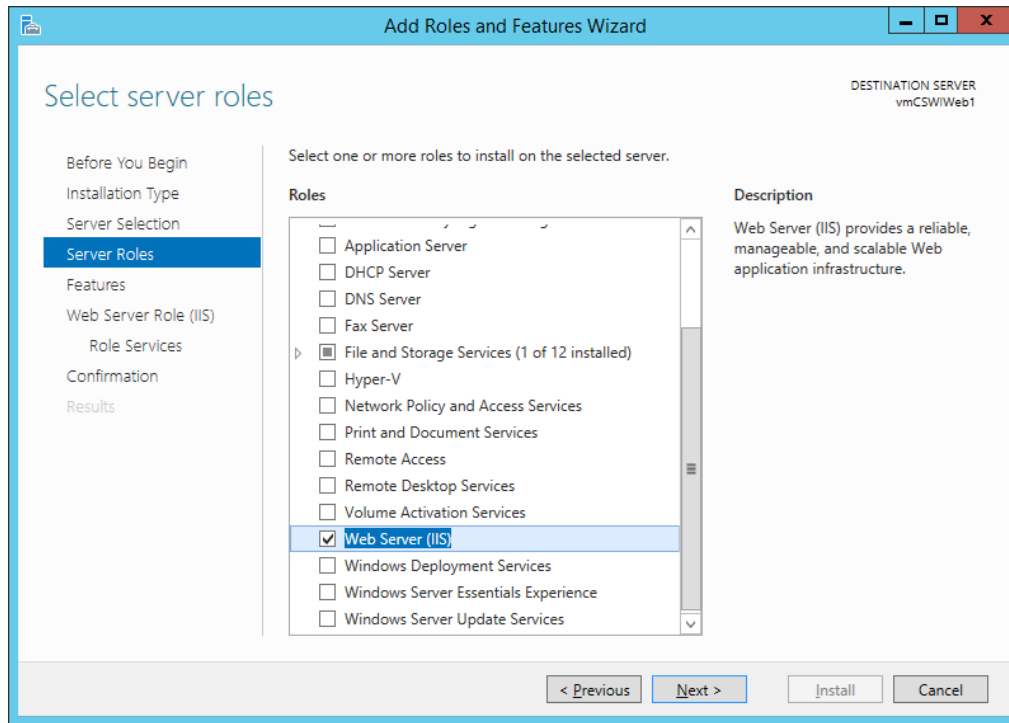
- If the following windows appears during installation, click **Add Features**.
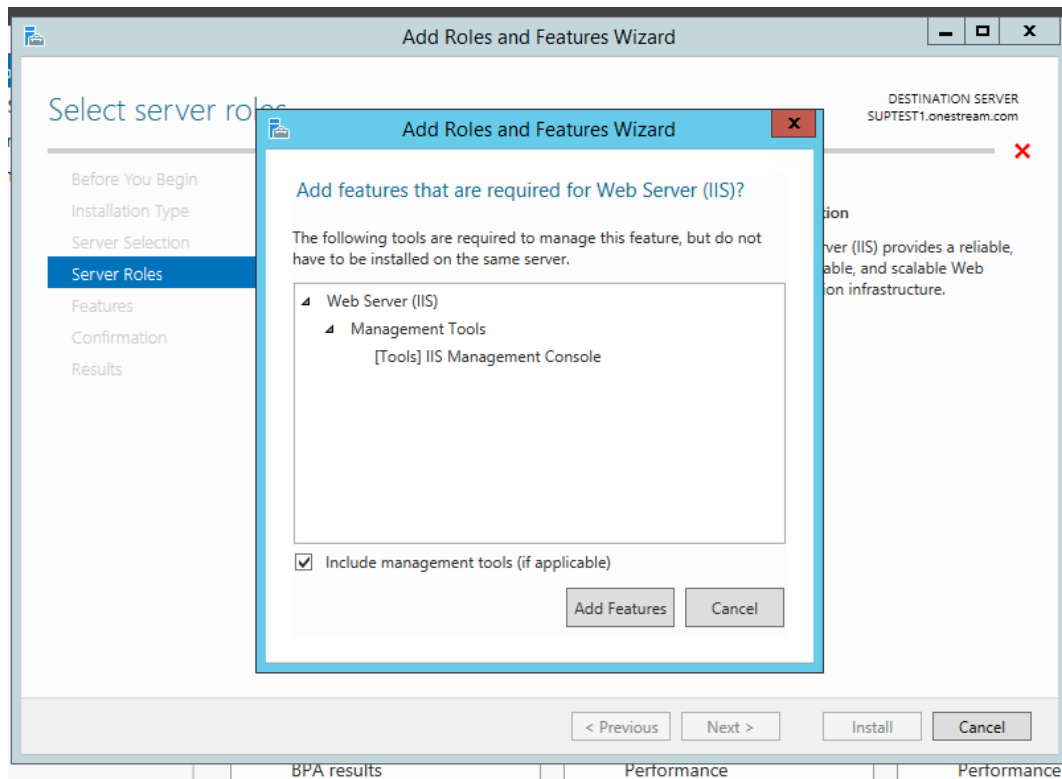
- Click **Next** and select all **.NET Framework 4.5** features including **WCF Services** Features as shown.
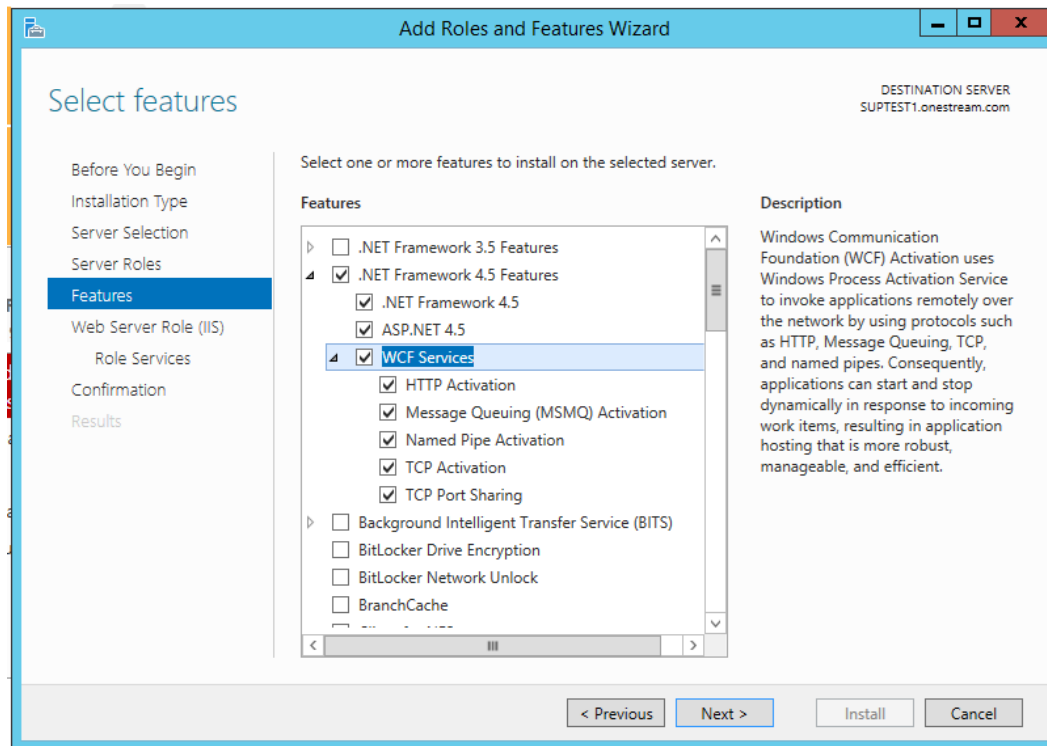
- Select all options under **Windows Process Activation Services**.
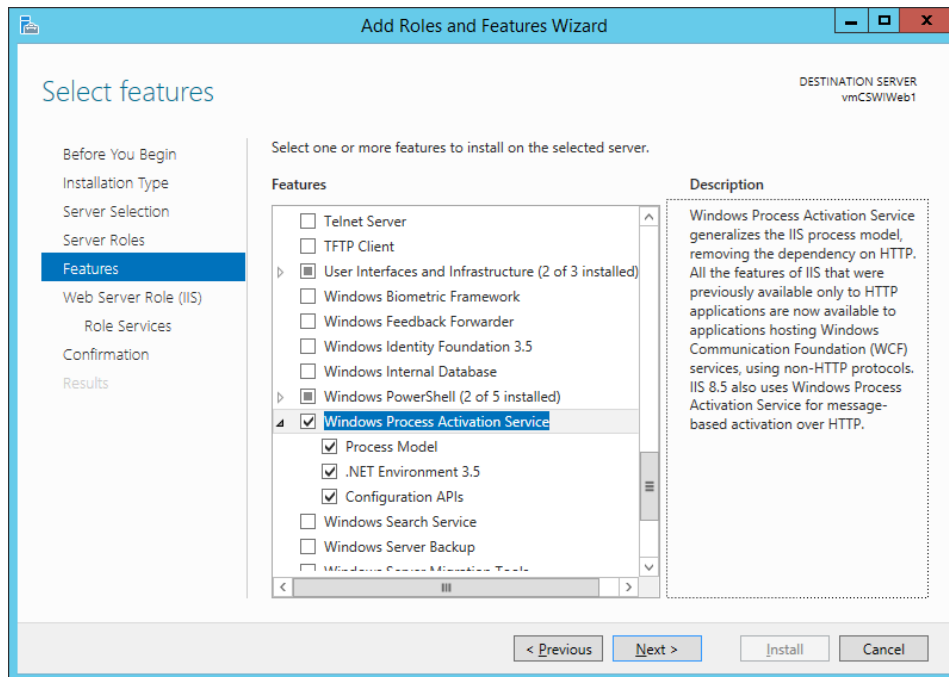
- Click **Next** and select these role services:

  ○ **Common HTTP Features**: All except HTTP Redirection and WebDAV Publishing.

  ○ **HTTP Logging**
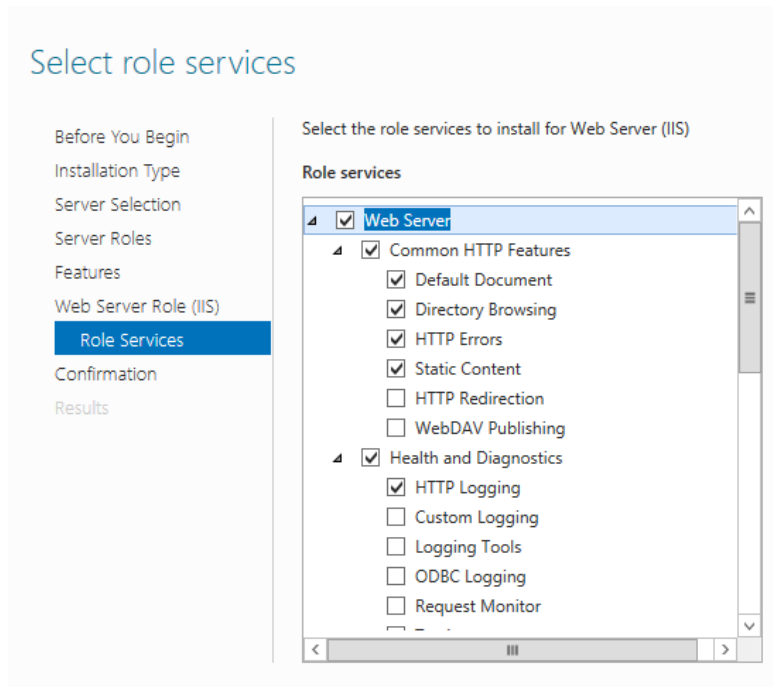
- **Performance: Static Content Compression**

    ○ **Security: Request Filtering**

    ○ **ASP.NET 4.5**

    ○ **ISAPI Extensions**

    ○ **ISAPI Filters**

    ○ **Management Tools: IIS Management Conso**le

- Click **Next** to install the IIS Feature.

Application Server Requirements and Considerations

Before installing the application server, perform these tasks and review these guidelines:

- Determine the quality of the application server.

- Allow Windows to manage the page file, which may mean moving the page file to a dedicated disk drive on the machine to accommodate growth.

- Ensure you meet the requirements for the Windows OS version that is automatically installed with updates. See the *System Requirements and Architecture Guide* for supported Windows Server versions.

- Install the .NET Framework. Refer to *System Requirements and Architecture Guide* for supported .NET Frame versions.

- Disable any Anti-Virus service. This can be re-activated after installation and set to exclude the installation directory.

- Discuss and plan for Application Server clustering. Do you know:

  - How many General and Staging Application Servers will be deployed?

  - How many Consolidation Application Servers will be deployed?

- Discuss and plan for Application Server types such as the following:

  - General request Application Servers such as reporting, user navigation, and workflow.

  - Dedicated Data Load (Stage) Application Servers

  - Dedicated Consolidation Application Servers

  - Combination Application Servers (Data Management)

- Discuss and plan the sizing of each Application Server and you are aware of the CPU and memory requirements identified in the *System Requirements and Architecture Guide.*

- Ensure there is enough space on the temp drive for swapping or a temporary file I/O.

- Know which level of authentication you use such as Basic, Secure, SSL, or FastBind.

- Have the information in this string used to connect to your user directory:
  `CN=UserDirectoryName,DC=myCompany,DC=com`

- For the database server connection:

- Ensure that Application Servers can connect to Database Server.

- Have the information below required to connect the Application Server to the Database.

  ```
  Server:  Data Source=hostname\MSSQLServerDatabaseServerName
  ;Initial Catalog=OneStream_Framework;username=<OneStream
  SQLAuthID>;password=???;Max Pool Size=3000;Connect Timeout=60
  ```

- Enable Network Discovery so My Company Name, LLCservers can communicate.

- Download the My Company Name, LLC Software installation package from MarketPlace and put it on the desktop or a folder on each server.

- Install and configure the following components for IIS:

- Enable Application Initialization role on the Application Servers.

- Install and enable Windows Process Activation Service (WAS) and WCF Services.
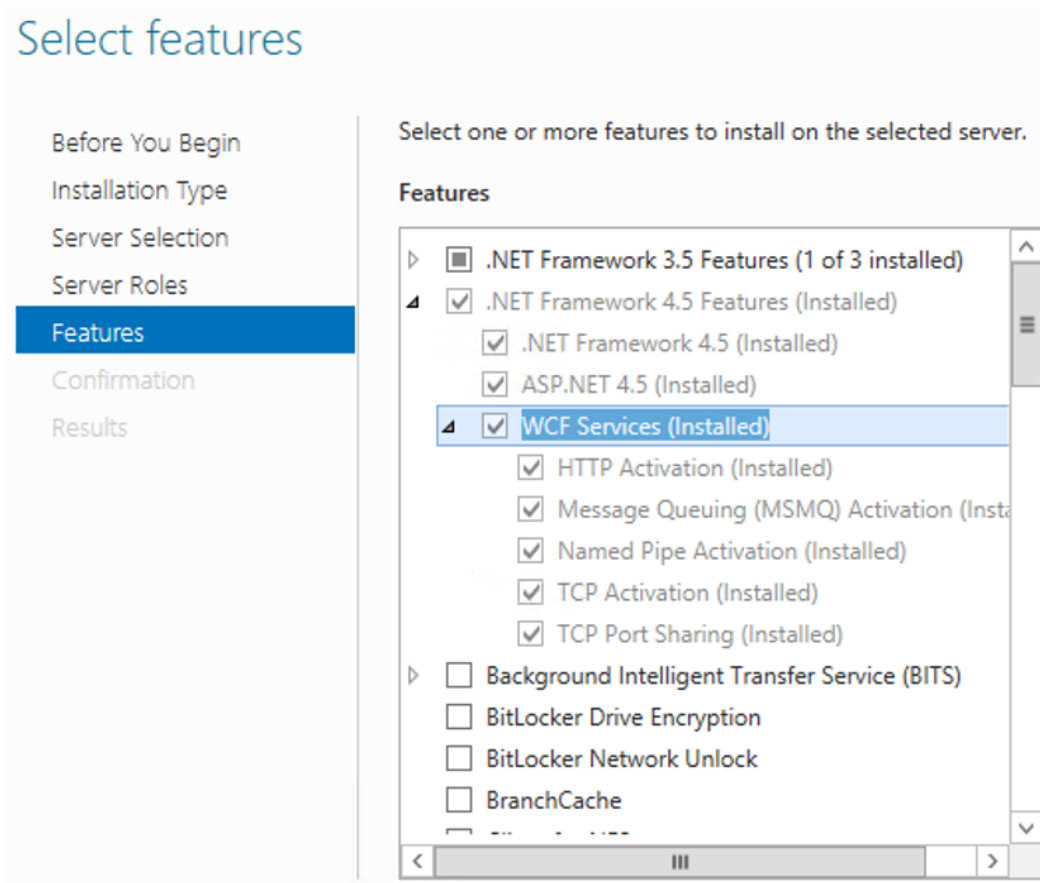
Database Server Requirements and Considerations

- Windows OS version installed with all updates. Refer to the OneStream System Requirements and Architecture Guide under OneStream Hardware and Software Requirements for supported Windows Server versions and Database Configuration Requirements.

- Proper version of 64-bit Microsoft SQL Server installed with all updates applied. SQL Server Enterprise Edition or versions beyond SQL Server Standard 2016 sp1 are recommended for larger, more complicated deployments due to support for table partitioning. See Data Server Recommendations in System Requirements and Architecture Guide for additional guidance and all supported versions.

    - Required Components:

        - Database Engine

        - Management Tools

- High Performance Power Plan has been enabled.

- Verify that the Microsoft SQL Server Database Server has the recommended amount of free disk space available and room to grow overtime.

- Update the SQL Server Database Transaction Log to grow by file size and not by percentage.   OneStream recommends setting the growth to grow in 100 MB Increments.

- To enhance performance over time in the Application Databases, OneStream recommends re-indexing the Application Databases periodically.

- Sizing is based on Customer Data Center Specs for CPU and Memory with minimums stated in System Requirements and Architecture Guide.

- Authentication and Rights

- Windows Integrated vs. SQL Server Native:

    - Use Native SQL Server Service Account instead of a Windows Integrated Account for the database to ensure fast connection and reduce network traffic for our hundreds of connections to the database.

    - If using Windows Integrated Security for SQL, the service account used in IIS will be the Database Account.

  - The OneStream Database account needs admin rights to the Master database in SQL Server in order for the administrator to be able to create new databases via the OneStream Database Configuration Utility.

  - The OneStream Database account should be given full user rights to Public and Sysadmin under Roles in SQL server.

- Backup / Recovery

  - All critical information is stored in the Framework and Application databases.

  - Verify that the SQL Server Database for the OneStream Framework and Application (s) are backed up per company backup policy. OneStream recommends a minimum of daily backups for the Application and Framework Databases. This can be handled by creating a SQL Server Maintenance Plan using the New Maintenance Plan wizard in SQL Server Management Studio. It is recommended to also backup Framework and Application databases used for development and test purposes, especially during initial build-out.

  - Verify that the SQL Server Database Transaction Logs are backed up on a regular basis in conjunction with the databases to avoid the Transaction Log growing to a full state.

Security Considerations

- Create IIS service account(s) before installation.

  - Can be separate accounts for Web and Application servers.

  - The service account used to run the OneStream Web Server IIS Web Application Pool requires file share privileges and the following Windows group access: IIS_ IUSRS

- The service account used to run the OneStream Application Server IIS Application Pool requires file share privileges and the following Windows group access: IIS_ IUSRS, Performance Log Users, Performance Monitor Users.

- The service account used to run the OneStream Management Service IIS Application Pool requires administrator privileges on the Application Server (OneStream recommends using "LocalSystem").

File Share Considerations

We suggest that you follow these best practices:

- Create a folder called OneStream on the network that is accessible and shared by each OneStream server.

- Give the following Read & Execute permissions to the OneStream folder(s):

  - Service Account running the OneStream Web Server IIS Web Application Pool

  - Service Account running the OneStream Application Server IIS Application Pool.

- Create a folder called FileShare under the OneStream folder. Your OneStream IIS service account ("NETWORK SERVICE") must have full rights to this folder.

- Create a folder under OneStream called Config where you can save and share the Web and Application Server configuration files.

- Give the following Read & Execute permissions to the OneStream Configuration folder:

  - Service Account running the OneStream Web Server IIS Web Application Pool

  - Service Account running the OneStream Application Server IIS Application Pool.

- Ensure that applications servers have the File Share for application server workspace (Logs, File Upload / Down Load).

- Ensure that each Web and Application Server can connect to the File Share.

- Regularly back up the OneStream Application Server and Web Server Configuration Files (such as XFAppServerConfig.xml and XFWebServerConfig.xml).

Firewall Considerations

- Web server requires port 50001 to be open as the web URL is accessed on port 50001.

- Application server uses standard web port, which is 50002 by default.

Virtualization Considerations

See System Requirements and Architecture Guide for detailed information related to Virtualized Environments.

For Database Servers, Application Servers and Web Servers:

- Use dedicated VM hosts.

- Sharing or over committing CPU's with other virtual guests is not supported. If you over-commit, performance impacts during processes like data load and consolidations may occur.

- Directly assign dedicated, logical CPU's to each OneStream VM guest.

- Dynamic Memory Management by the VM Host is not recommended. Because RAM usage increases quickly, VM guests without committed RAM immediately available may experience performance impacts.

- Ensure you update to the latest Windows Server patch level.

Anti-Virus Considerations

Verify that:

- IIS is being excluded by any antivirus programs installed on My Company Name, LLC Servers. See this Microsoft article for information: https://support.microsoft.com/en-us/kb/821749

- The My Company Name, LLC installation directory (C:\Program Files\OneStream Software\...) is being excluded from any virus program live active scanning.

About Installation and Configuration

Existing customers should refer to the *Upgrade Guide* for information about installing the latest release.

Refer to the *System Requirements and Architecture Guide* for:

- An overview of the platform architecture, technology and components such as the Web Client, Web Server, Database Server and more.

- Third-party software requirements.

- Hardware and software requirements for components such as the Web Server, Application Server, Data Server and Client Workstation.

- Information about virtualized environments.

- Environment configuration guidelines.

OneStream Component Technology

Client

Web

OneStream Mobile

Microsoft Office

OneStream Excel Add-in (optional)

Windows Client

OneStream Windows App (including Spreadsheet and Text Editor features)

OneStream Studio for report authoring (optional)

Administration

OneStream Server Configuration Utility for initial product configuration

OneStream Database Configuration Utility for initial product configuration

Web Server

- Microsoft Internet Information Services (IIS), Windows Process Activation Service (WAS) and .NET Framework using Windows Communication Foundation (WCF)

- OneStream Web Server software installation

Application Server

- Microsoft Internet Information Services (IIS), Windows Process Activation Service (WAS) and .NET Framework using Windows Communication Foundation (WCF)

- OneStream Application Server software installation.

Database Server

- Microsoft SQL Server

  NOTE: Use Table Partitioning (requires SQL Server version that supports Table Partitioning, such as Enterprise, Azure or SQL Standard version 2016 SP1 or higher)

- Transaction logs and data at a minimum should be stored on separate drives to spread I/O across drives. For more advanced configurations and improved throughput, create additional File Groups to spread I/O across devices.

Supported Authentication Providers

You can configure an environment for external authentication in one of two ways:

- Using the OneStream Identity Server that supports combinations of most OIDC compliant and SAML compliant external identity providers (IdPs) or native authentication coupled with an external IdP. This enhances user authentication by supporting multiple providers in one environment. See the *Identity Management Guide*.

- Using one of these providers:

    ◦ Native authentication

    ◦ Azure AD

    ◦ MSAD

    ◦ LDAP

    ◦ Okta

    ◦ PingFederate

    ◦ SAML 2.0

Application Folder Permissions

User logs are stored in the My Company Name, LLC application folder directory, which also acts as the Application Server's workspace. Ensure that the Application Server identity specified in Microsoft IIS has full access to this folder structure.

About the Installation Packages

You install and deploy My Company Name, LLC using separate installation packages that contain different components. One of these packages contains the primary server components, which also includes the web application.

The other packages contain supplemental client applications used for report design and Microsoft Excel based analysis.

Installation Package Content

The lists below identify the contents of the installation packages. Each system component is described in a later section. All software can be downloaded from the MarketPlace.

OneStream Servers Installation Package

- Application Server (optional)

- Web Server (optional)

- Rest API (optional)

- Mobile Server (Optional)

- Configuration Tools (optional)

  - Database Configuration Utility

  - Server Configuration Utility

OneStream Windows App Client Installation Package

OneStream Windows App (optional installation package as an alternative to ClickOnce deployment)

OneStream Studio Client Installation Package

Report Designer

OneStream Excel Add-In Client Installation Package

OneStream Excel Add-In

Installation Overview

The following sections identify the basic tasks you will perform to install the product.

Step 1: Install the Application Servers

1. New installations: Review the requirements.

2. Install the My Company Name, LLC Servers Installation Package on each application server. Perform a **Custom Install** and clear the **Web Server optional component**.

Step 2: Install the Web Servers

1. Install the My Company Name, LLC Servers Installation Package on each application server.

2. Select **Custom Install** and clear the **App Server & Database Configuration Utility** optional components.

Step 3: Create Database Schemas and Connection Strings

1. As an Administrator, run the Database Configuration Utility on one of the application servers.

2. Create the database schemas:

    a. Create an empty Framework database and its tables.

    b. Create an empty Application database and its tables.

    c. Create an empty State database (tables are created by the application).

3. Configure and export connection strings:

    a. Modify the Framework connection string (Advanced Options). Set Pooling, Max Pool, and Connection timeout values.

    b. Click **Tools** to export the encrypted connection string to an XML file. This file is imported to application server configuration file.

Step 4: Configure the Application Servers

1. As an Administrator, run the Server Configuration Utility on one of the application servers.

2. Create an application server configuration file (XFAppServerConfig.xml) in the file share in a folder called **ConfigurationtFiles**.

3. Import the encrypted database connection string file you created in step 3.

4. Follow the application server configuration process.

Step 5: Configure the Web Servers

1. As an Administrator, run the Server Configuration Utility on one of the web servers.

2. Create a Web Server Configuration file (XFWebServerConfig.xml) in the file share under a folder called **ConfigurationFiles**.

3. Follow the web server configuration process.

Step 6: Test

1. Make sure that any firewalls are OFF or that exceptions were added for My Company Name, LLC port traffic.

2. Launch the web server:
   `http://<Servername>:50001/OneStreamWeb/OneStreamXF.aspx.`

Step 7: Log onto the Framework Database (System Administration)

Username: Administrator

Passowrd: OneStream

Step 8: Add Application References

Select **Applications** and create a reference to the application database created in Step 3 "Create Database Schemas...".

Installing Server and System Components

Installing the OneStream Servers Package

This is the primary OneStream installation package. This is a wizard-based package used to install a Complete server setup that includes the web server, application server and all utilities. A Custom install allows the appropriate components to be selected for the server type being built.

Building an All-In-One Server (Combined Web and Application Server)

To build an All-In-One Server, follow the installation wizard prompts and select the Complete installation option. This will instruct the wizard to install all server components required for a server to function as both a web and application server.

Building an Application Server

To build an Application Server, follow the installation wizard prompts and select the Custom installation option.

Next, only select the following installation options:
Application Server
Server Configuration Utility

This will instruct the wizard to install all server components required for a server to function as an Application Server.

Building a Web Server

To build a Web Server, follow the installation wizard prompts and select the Custom installation option.

Next, only select the following installation options:
Web Server
Server Configuration Utility
Mobile Server (optional)
Rest API (optional)

This will instruct the wizard to install all server components required for a server to function as a Web Server.

Database Configuration Utility

The Database Configuration Utility is a component that can be installed on any server and is used to access and configure the SQL Server database server used to host the OneStream system databases.

This utility was purposely kept separate from the Server Configuration Utility to allow for separation of duties between IT resources. The Database Configuration Utility can be installed for database administrator personnel only if desired and the resulting database connecting information can be delivered to the application server administrator personnel via an XML file containing encrypted database connection string information. For more details on this utility, see Configuring System Components.

Uninstalling the OneStream Servers Package

1. Click **Control Panel > Programs > Uninstall a Program** and search for the OneStream items to remove.

2. Right-click items and select **Uninstall**.

Uninstall and Re-install on Another Drive

1. Confirm the configuration in a few additional areas in IIS. Once the new drive is installed, go into Internet Information Services (IIS) Manager:
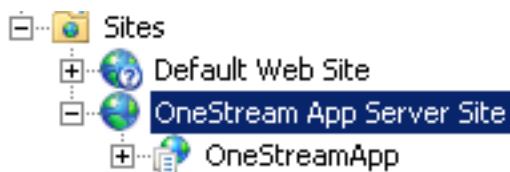
2. Click the OneStream App Server Site and choose **Advanced Settings** on the right side. In the pop-up window, confirm the Physical Path is correct and, if not, update it accordingly.

ID

Name

Physical Path

Physical Path Credentials

Physical Path Credentials Logon Type

Start Automatically

3. Click **OneStreamApp** and choose **Advanced Settings** on the right side. Confirm the physical path is correct or update it.

Application Pool

Physical Path

Physical Path Credentials

Physical Path Credentials Logon Type

Virtual Path

4. Recycle the App Pool, recycle IIS and test.

Installing the Application Server

> **NOTE:** Refer to the Configuring System Components in the Installation and Configuration Guide before proceeding with this section.

1. Launch OneStream Server Setup.msi by right-clicking and choosing **Install**. This launches the Installation Wizard Welcome Screen.

Name		Type
Setup	**Install**	Windows Installer Package
Setup.v	Repair	WIXPDB File
	Uninstall
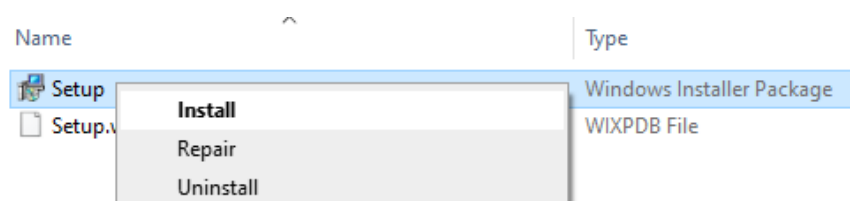
2. Click **Next** to continue the installation operation.

3. Accept the terms of the license agreement and click **Next**.
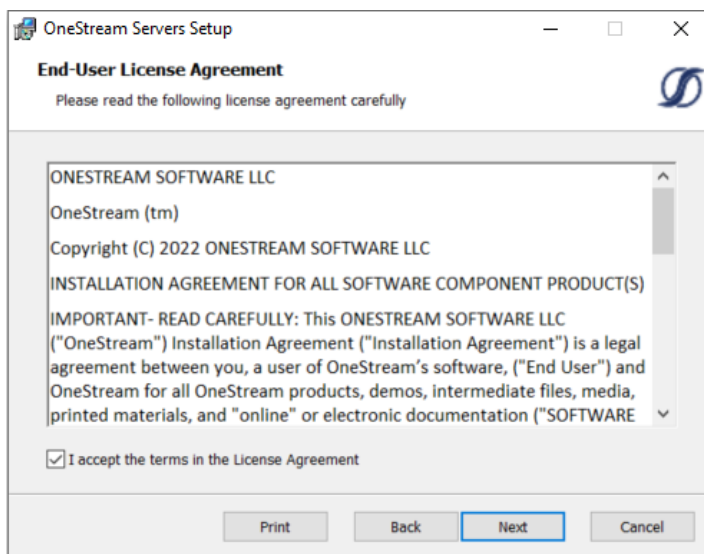
4. Select the **Custom** setup type and click **Next**.
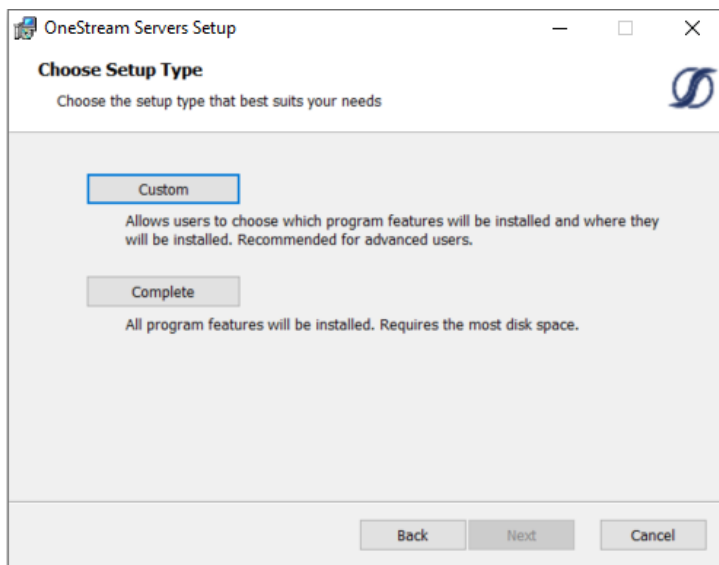
5. Install the Application Server only, click **Application Server** and choose Will be installed on local hard drive. Repeate for Server Configuration Utility. Click **Next**.

    > **NOTE:** (Optional) We recommend that the Database Configuration Utility be installed on the Application Server.
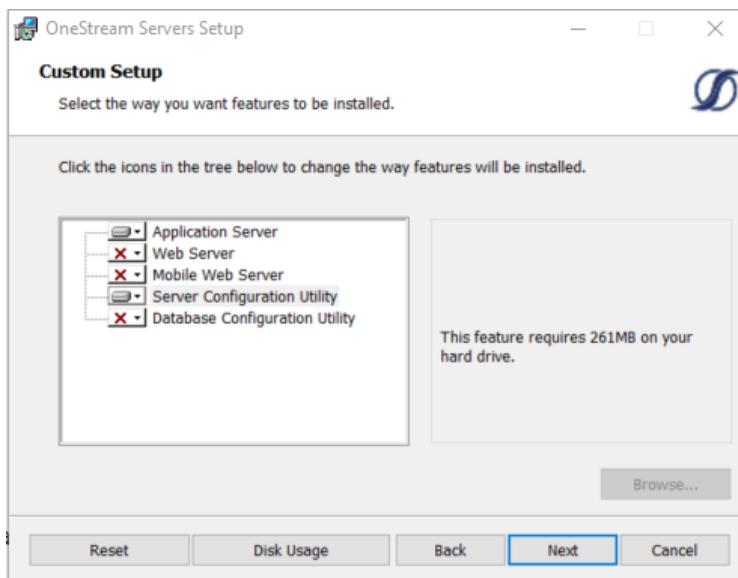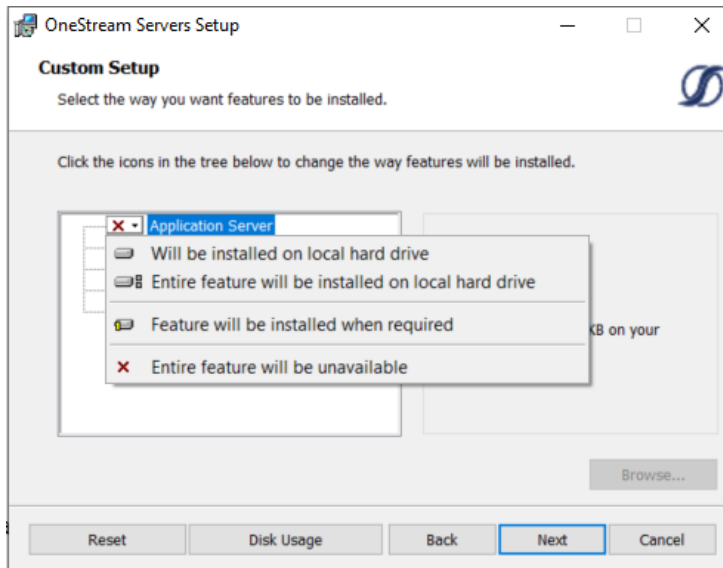
6. If needed, change the folder path and click Next.
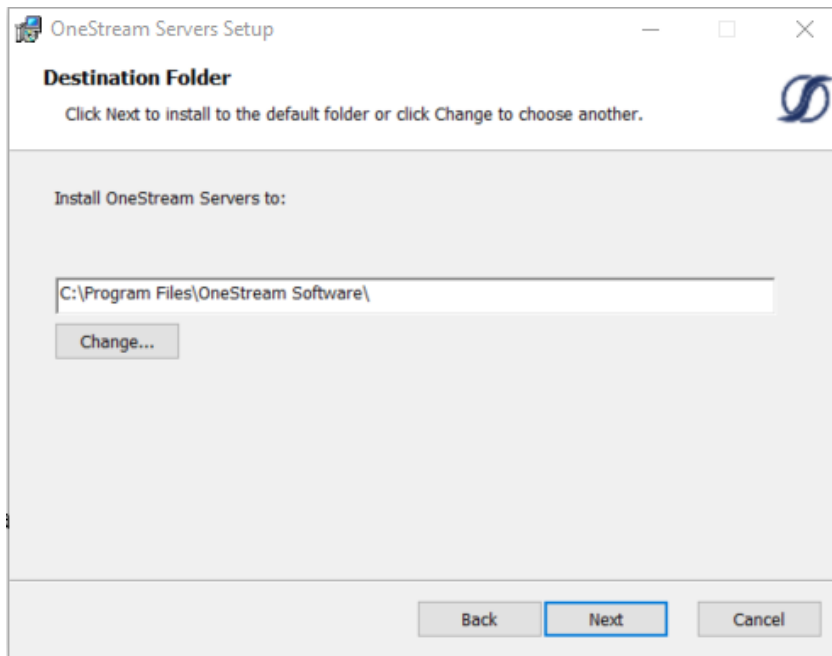
7. Click Install to begin the installation.
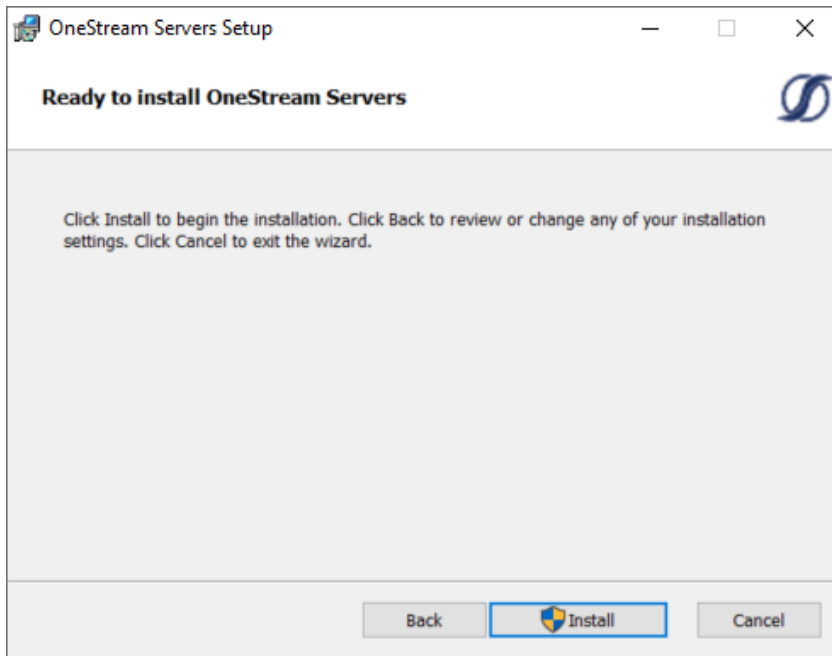
8. Click **Finish** to complete the installation.
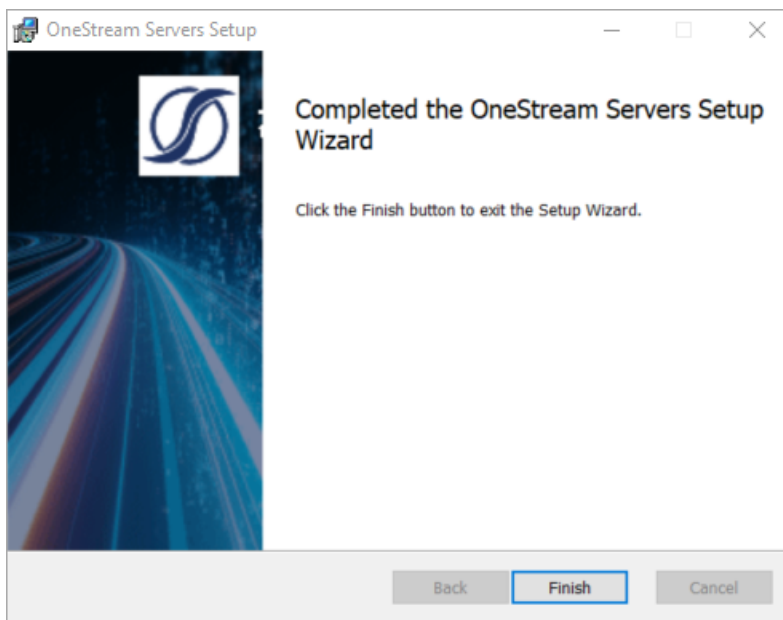
Configuring the Application Server

For more information regarding configuration, refer to the Configuration section of this guide.

1. Launch the OneStream Server Configuration Utility using Run As Administrator.
    Go to **Start > OneStream Software > OneStream Server Configuration Utility**. Right-
    click and choose **Run as Administrator**.

2. Update the OneStream Application Server ASP.NET configuration file to point to the
    location of the OneStream Application Server Configuration File in the environment.

    a. Choose **File > Open ASP.NET Configuration File**.
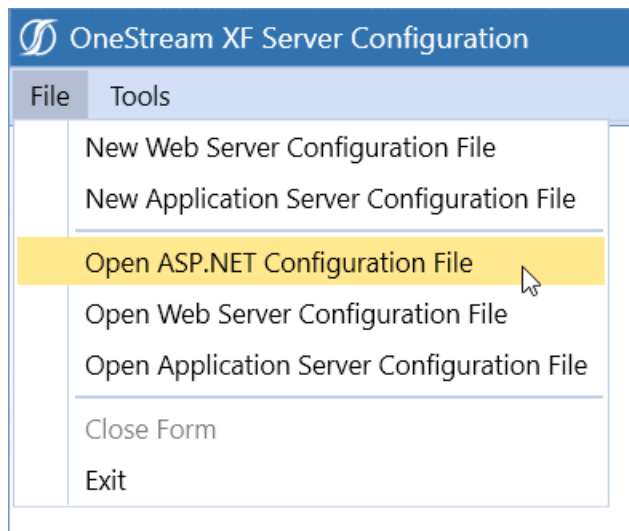
    b. Browse to the following location: `<installdrive>\Program Files\OneStream`
        `Software\OneStreamAppRoot\OneStreamApp`, select the **Web.config** file and
        click **Open**.
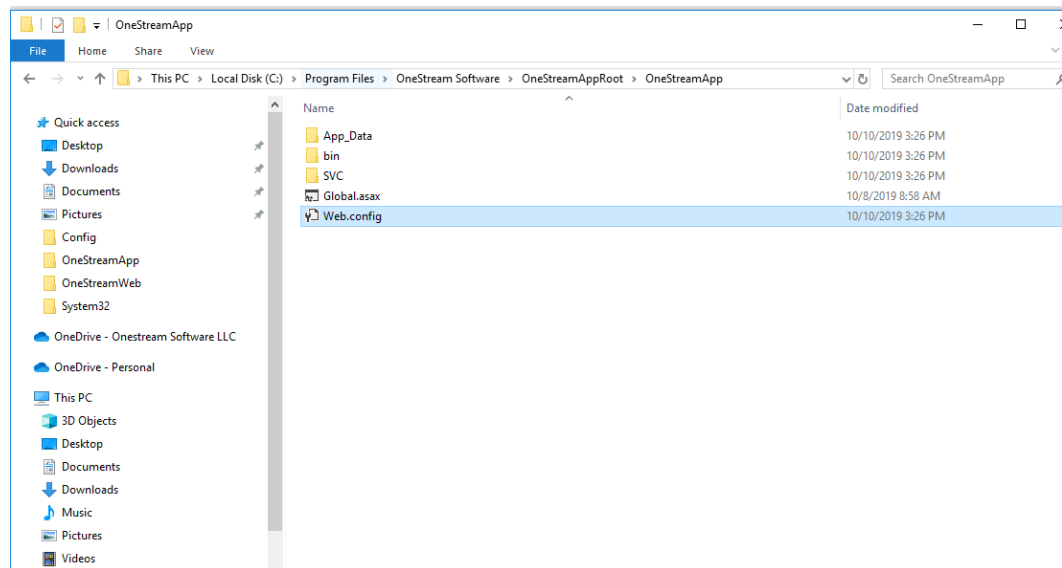
c. Update the path to the file location of the OneStream Application Server Configuration File (XFAppServerConfig.xml). By default this file is located at `<installdrive>\Program Files\OneStream Software\OneStreamAppRoot\OneStreamApp\App_Data`.
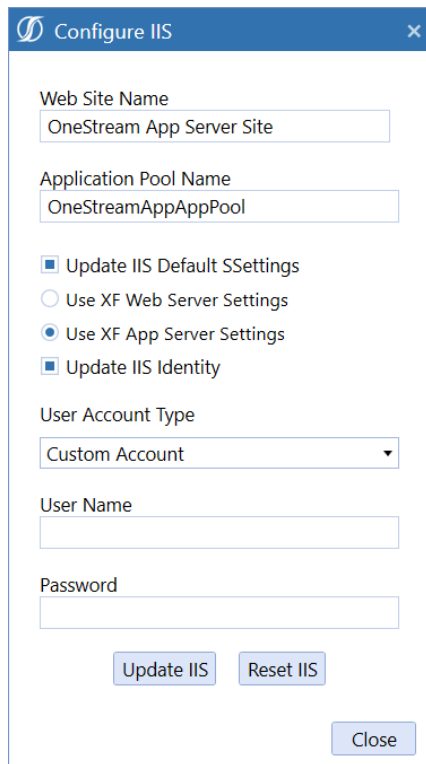
    Best Practice: Move a copy to the OneStream\Config folder, which can be used by all Application Servers.
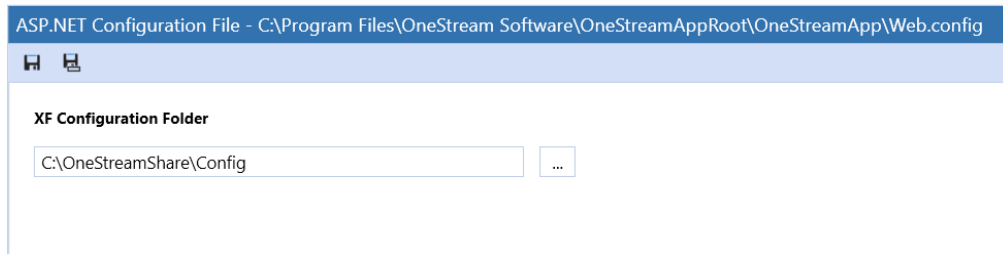
d. Save the configuration file.

Update the Application Server IIS Settings using Configure IIS Tool

1. Choose **Tools > Configure IIS**.

2. Enter the following values:
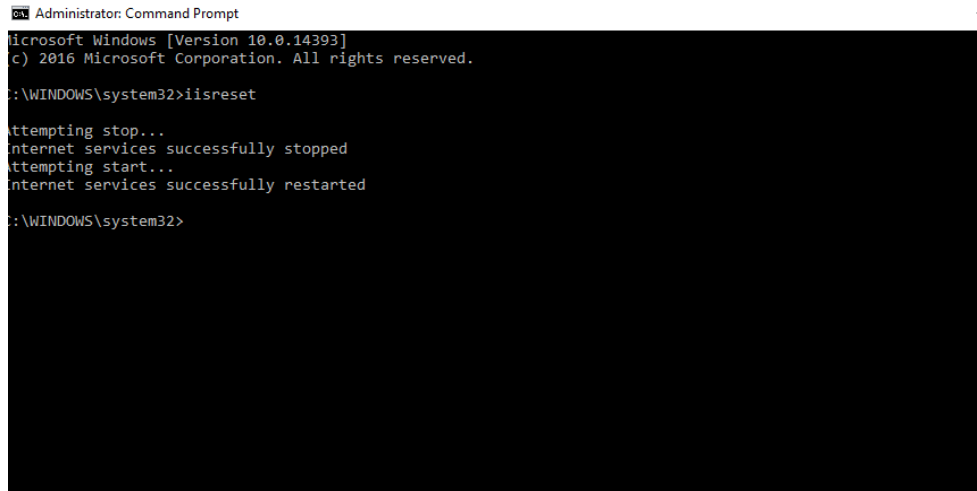
   - Web Site Name: OneStream App Server Site

   - Application Pool Name: OneStreamAppAppPool

3. Check Update IIS Default Settings.

4. Select Use App Server Settings.

5. Check Update IIS Identity.

6. Set the User Account Type to the proper value from the drop down list. (It should be "Custom Account" if using a domain service account.)

    a. UserName: Enter the OneStream Service Account as (Domain\UserName).

    b. Password: Enter the Password.

7. Click **Update IIS Settings** to set the IIS Application Pool settings and click **OK**.

8. Click **Reset IIS** to recycle IIS.

> **NOTE:** You can also recycle IIS by stopping and restarting the web server in IIS, or by using an IISRESET Command via an administrator command prompt.

> **NOTE:** If users do not use the "Configure IIS" utility to set the settings for the OneStream App Server, they will need to make sure that they:

- Manually update the username in IIS.

- Set Ping Enabled to False.

- Set the following settings for the application server Website and AppAppPool:

Start Mode: Always Running
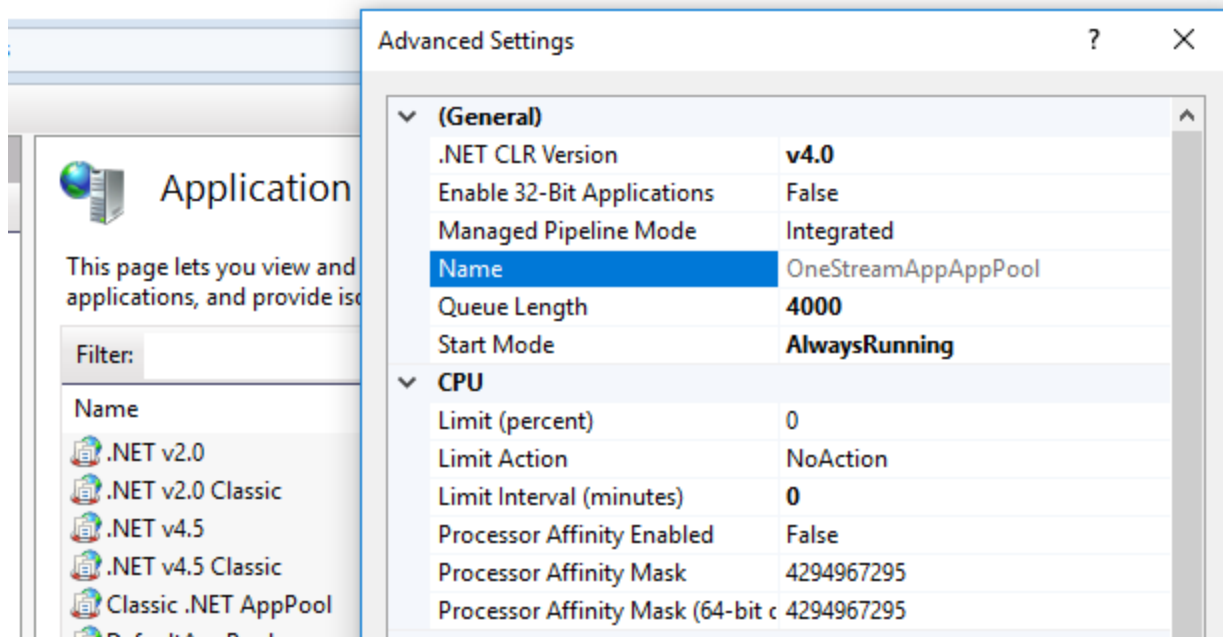
PreLoad Enabled: True on the OneStream App Server Site
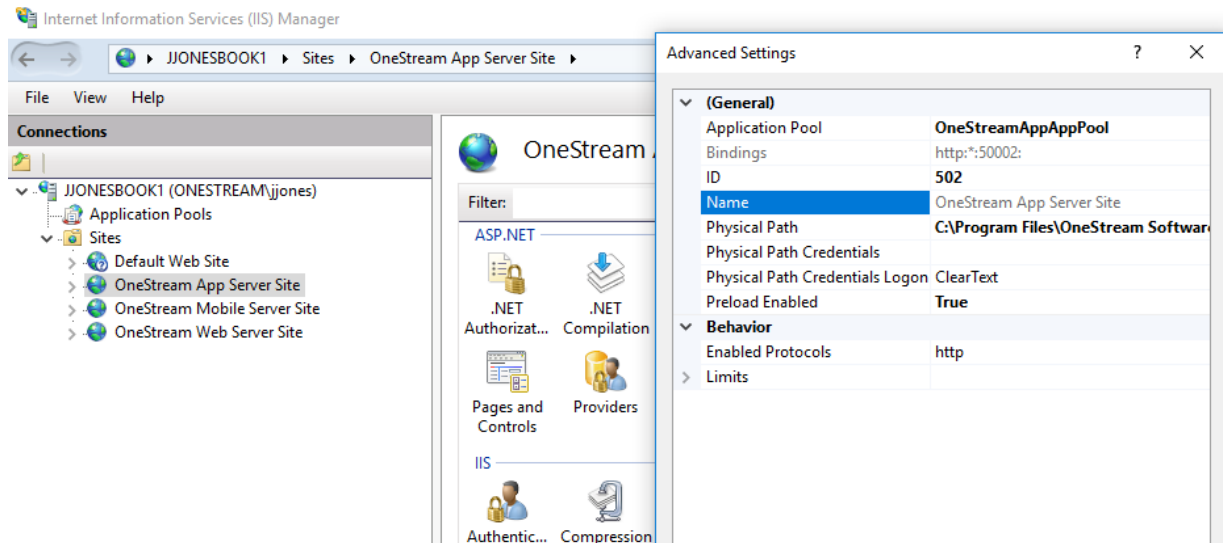
Preload Enabled True on the OneStreamApp registration:
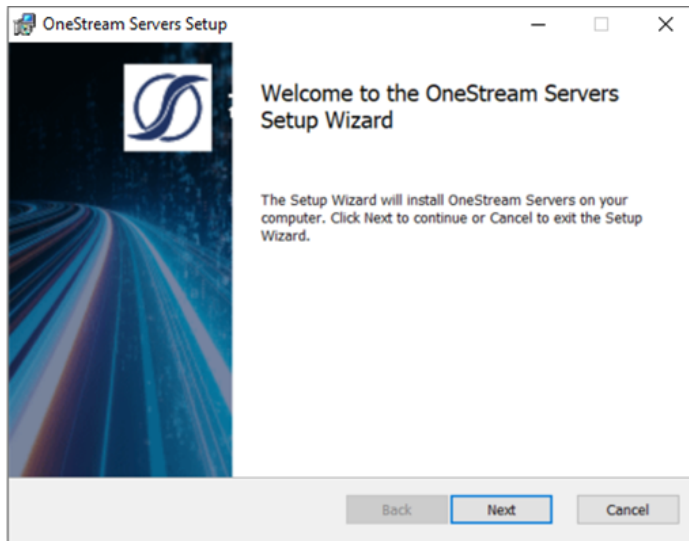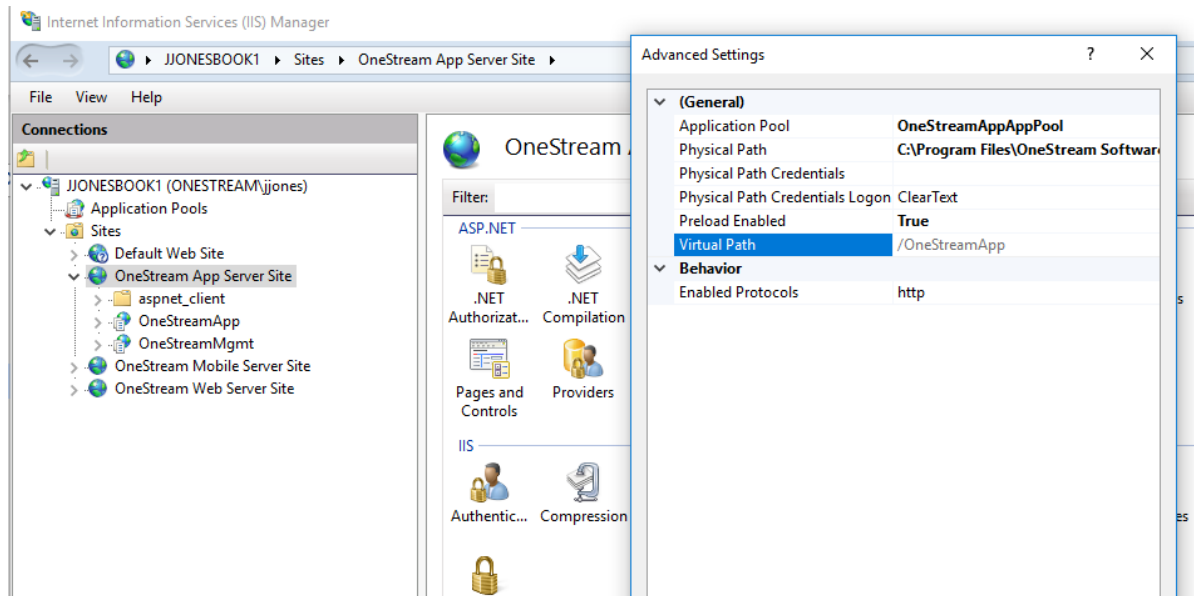
Installing the Web Server

> **NOTE:** See *Configuring System Components* on in the Installation and Configuration Guide before proceeding with this section.

1. Launch OneStream Server Setup.msi by right-clicking and choosing **Install**.
   This launches the Installation Wizard Welcome Screen.
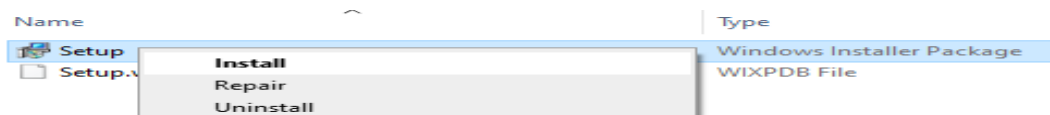
2. Click **Next** to continue with the installation operation.
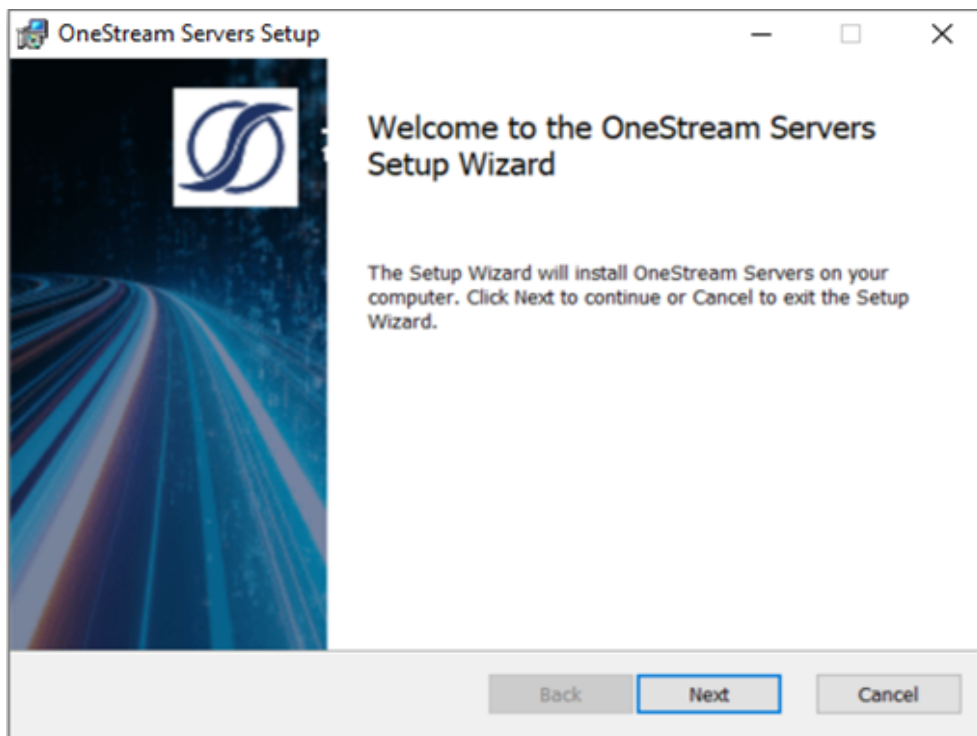
3. Accept the terms of the license agreement and click **Next**.
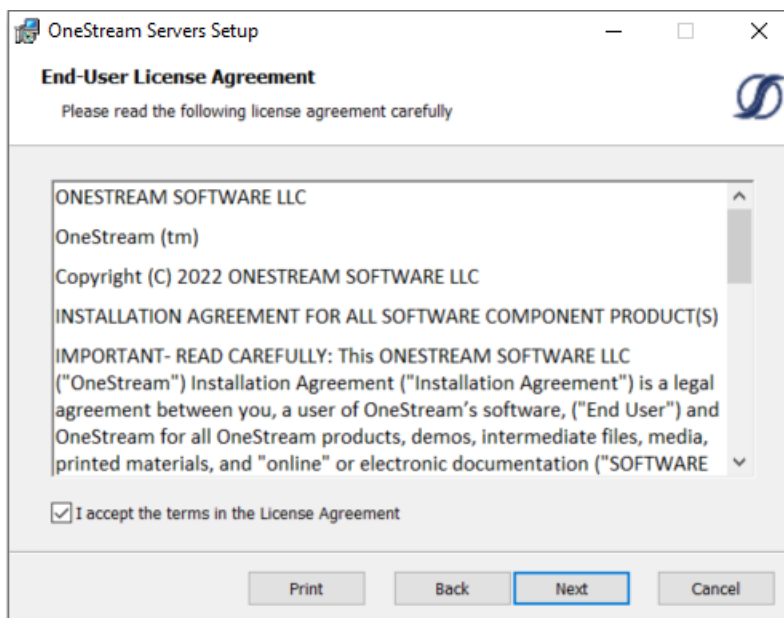
4. Select **Custom** as the setup type.
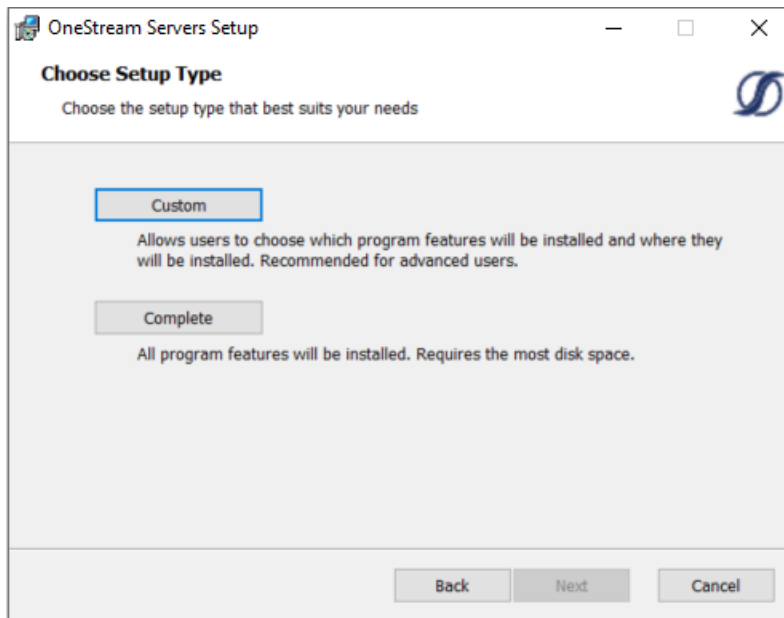
5. To install the Web Server only, click **Web Server** and choose **Will be installed on local hard drive**. Repeat for **Server Configuration Utility**. Click **Next**.
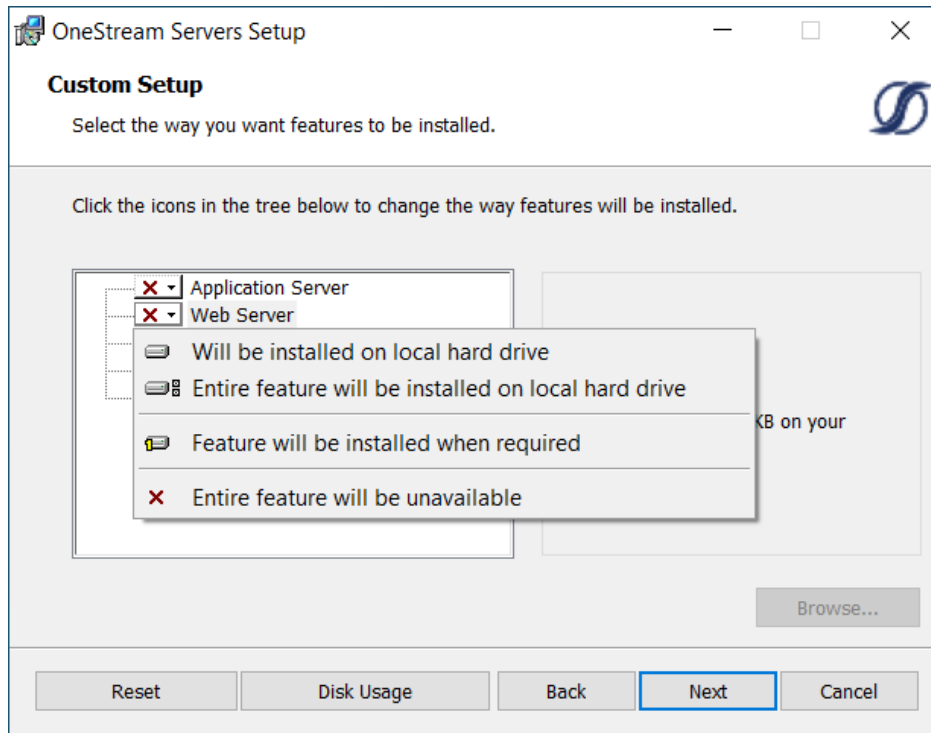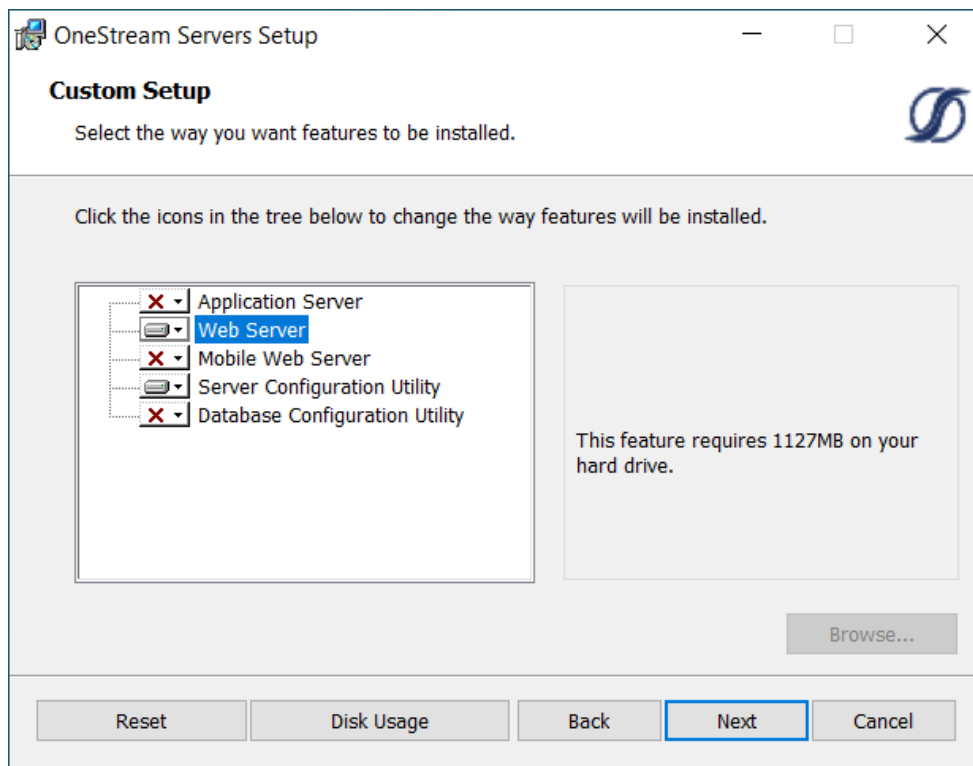
6. Change the folder path, if needed, and click **Next**.

    > **NOTE:** The default installation path is C:\Program Files\OneStream Software. Select Change to change the drive for the installation. For example, D:\Program Files\OneStream Software.

7. Click **Install** to start.

8. Click **Finish** to complete the installation.

Configuring the OneStream Web Server

Update the OneStream Web Server IIS Settings using Configure IIS Tool

1. Choose Tools > Configure IIS.

2. Enter the following values:

    a. Web Site Name: OneStream Web Server Site

    b. Application Pool Name: OneStreamWebAppPool

3. Check **Update IIS Default Settings**.

4. Select **Use Web Server Settings**.

5. Check **Update IIS Identity**.

6. Set the User Account Type to the proper value from the drop down list. (It should be "Custom Account" if using a domain service account.)

    a. UserName: Enter the OneStream Service Account as (Domain\UserName).

    b. Password: Enter the Password.

7. Click **Update IIS Settings** to set the IIS Application Pool settings and click **OK**.

8. Click **Reset IIS** to recycle IIS.

    > **NOTE:** You can also recycle IIS by stopping and restarting the web server in IIS, or by using an IISRESET Command via an administrator command prompt.

Installing the OneStream Mobile Server

1. Launch OneStream Server Setup.msi by right-clicking and choosing Install. This launches the Installation Wizard Welcome Screen.
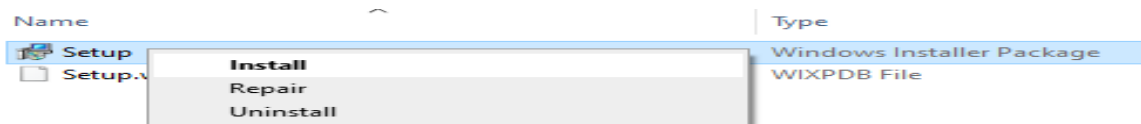
2. Click **Next** to continue with the installation operation.
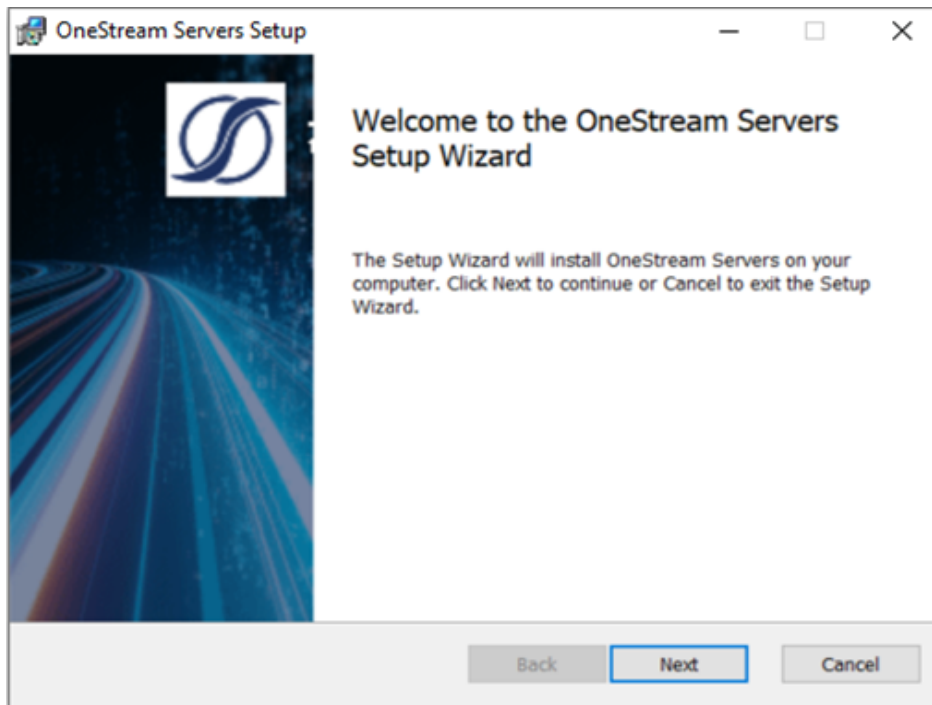
3. Accept the terms of the license agreement and click **Next**.
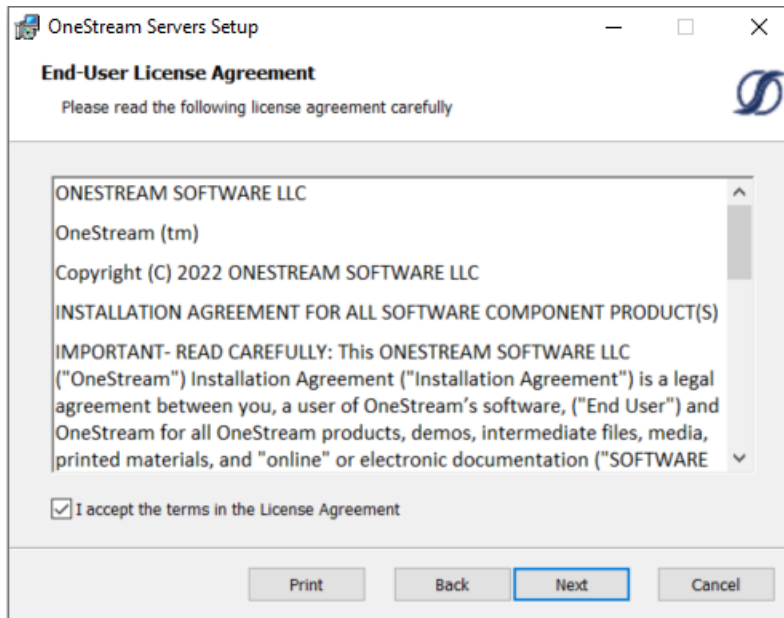
4. Select **Custom** setup and click **Next**.
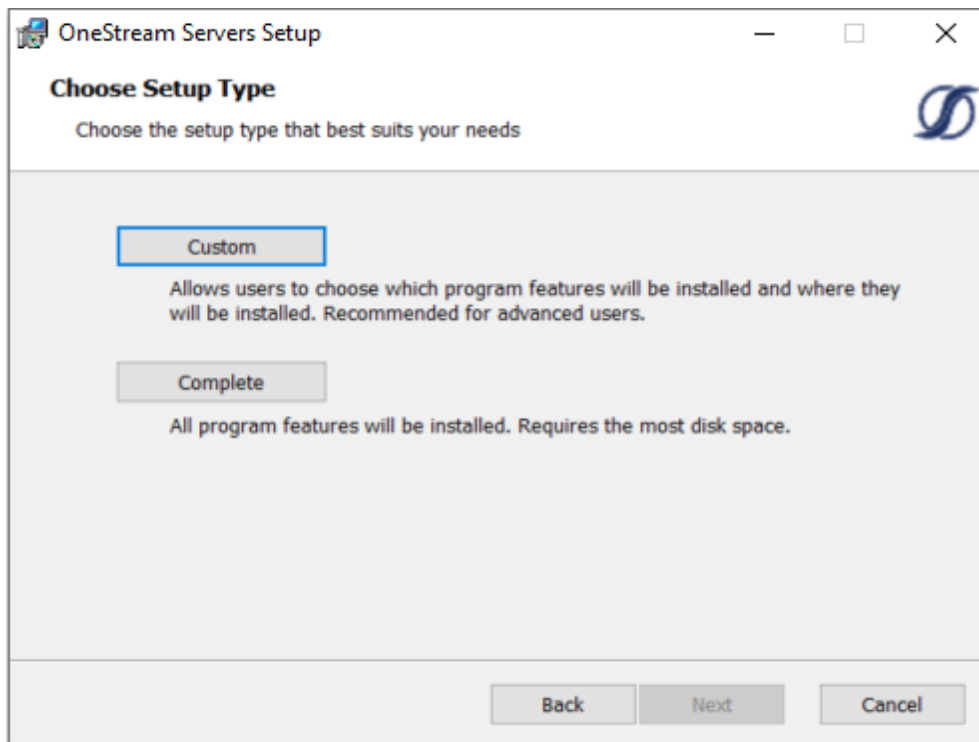
5. To install the Mobile Server only, click **Mobile Server** and choose **Will be installed on local hard drive**. Repeat for **Server Configuration Utility**. Click **Next.**
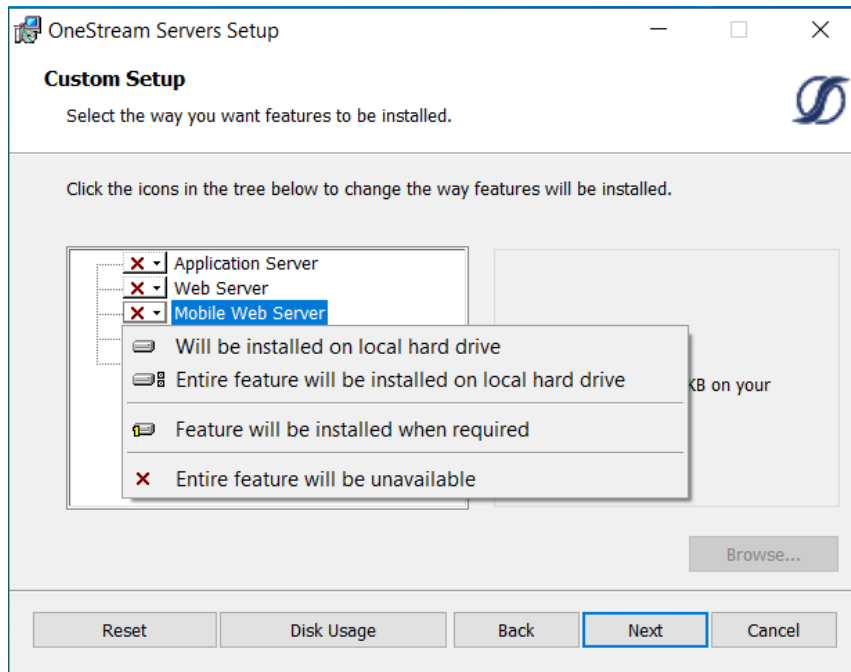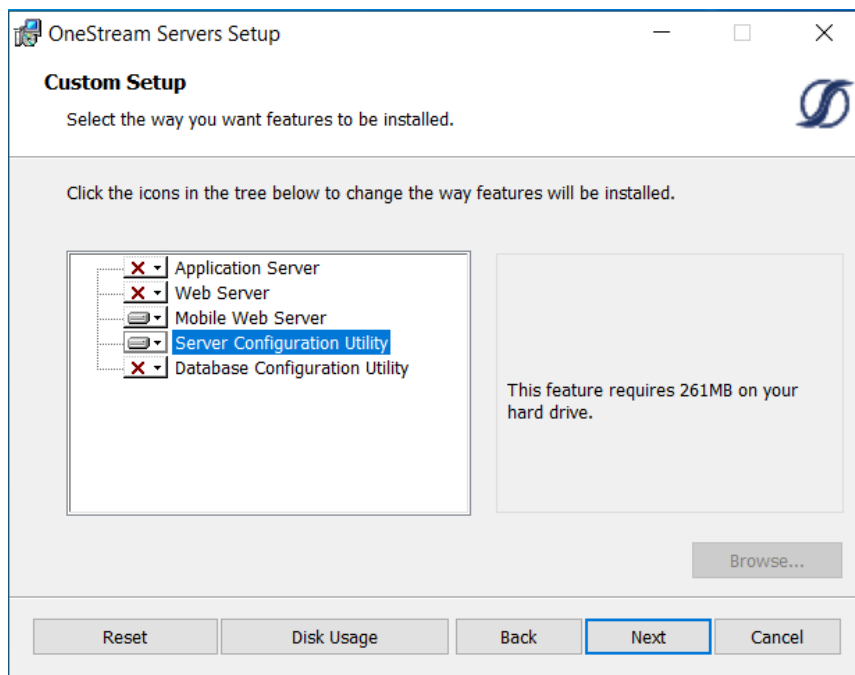
6. Change the folder path, if needed, and click **Next**.

> **NOTE:** The default installation path is C:\Program Files\OneStream Software. Select Change to change the drive for the installation. For example, D:\Program Files\OneStream Software.
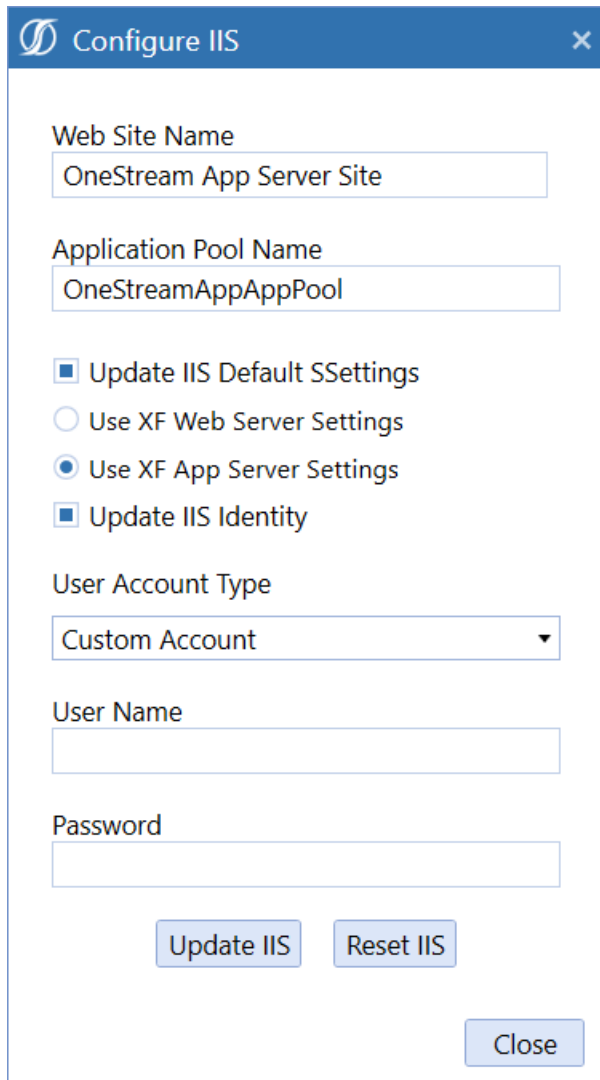
7. Click **Install**.

8. Click **Finish** to complete the installation.

Configuring the OneStream Mobile Server

Update the Mobile Server IIS settings using the Configure IIS tool.

1. Select **Tools** > **Configure IIS**.

2. Enter the following values:

   a. Web Site Name: OneStream Mobile Server Site

   b. Application Pool Name: OneStreamMobileAppPool

3. Check **Update IIS Default Settings**.

4. Select **Use App Server Settings**.

5. Check **Update IIS Identity**.

6. Set the User Account Type to the proper value from the drop down list. (It should be "Custom Account" if using a domain service account.)

    a. UserName: Enter the OneStream Service Account as (Domain\UserName).

    b. Password: Enter the Password.

7. Click **Update IIS Settings** to set the IIS Application Pool settings and click **OK**.

8. Click **Reset IIS** to recycle IIS.

> **NOTE:** You can also recycle IIS by stopping and restarting the web server in IIS, or by using an IISRESET Command via an administrator command prompt.

Installing Client Components

Refer to the *System Requirements and Architecture Guide* for Hardware and Software Requirements related to Client Workstations and versions supported of relevant Microsoft Office products.

Considerations for Excel Add-in and OneStream Studio

Some considerations when installing the OneStream Excel Add-in and Studio:

- In order to properly install the OneStream Excel Add-in or Studio, changes are required to the Windows registry. Ensure that the user has rights to update their own registry while doing installations. Note that this limitation does not apply to the use of OneStream Windows App.

- Changes made to the Windows registry for the OneStream Excel Add-in or Studio during the installation are made only for the user who is doing the installation. For example, if someone in Information Technology is logged into the user's machine to do the installation, the business user will not be able to access the OneStream Excel Add-in.

- The registration process of the OneStream Excel Add-in or Studio requires certain Microsoft .NET Framework rights to execute a program stored in that folder. The folder is C:\Windows\Microsoft.NET\Framework (or Framework64 if the user is running the 64-bit version of Excel).

- If the user had Excel 2003 or another prior version installed before Excel 2010 or greater was installed on the machine and subsequently uninstalled the older version of Excel, the newer version of Excel and the OneStream Excel Add-in will need to be completely uninstalled and reinstalled.

Other Office Add-ins may conflict with the OneStream Excel Add-in, so discuss other registered Add-ins when discussing installation issues with OneStream Support.

Installing the OneStream Excel Add-In Client Package

The Excel Add-In client should be installed for any user who will enter or analyze data in Excel. The Excel Add-In is an Excel COM add-in that connects to OneStream servers through the OneStream web services.

An administrator or user with administrator privileges must before the initial installation.

Installing Using the Installation Wizard

1. Double-click **Setup.msi** then click **Next**.
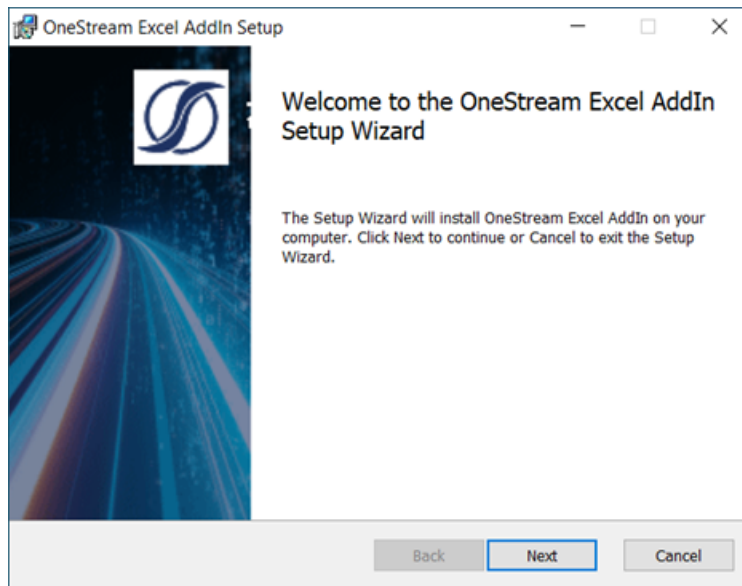
2. Accept the terms of the license agreement and click **Next**.
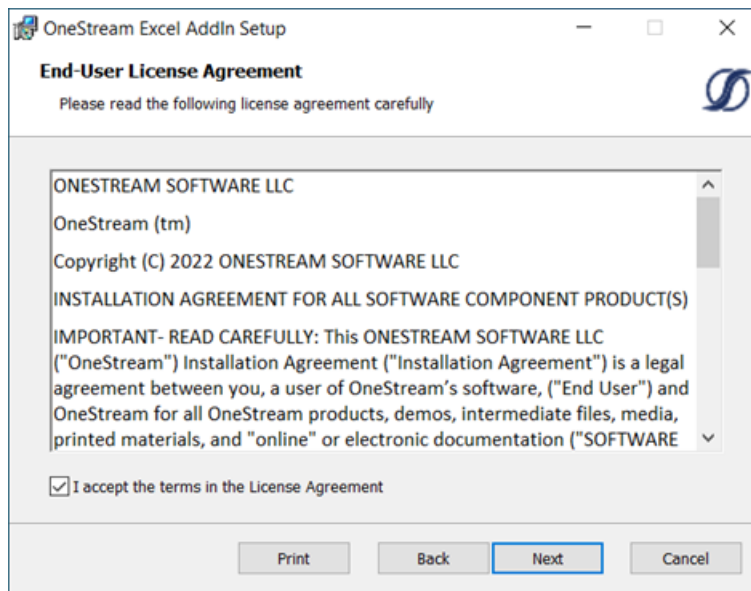
3. Change the folder path, if needed, and click **Next**.
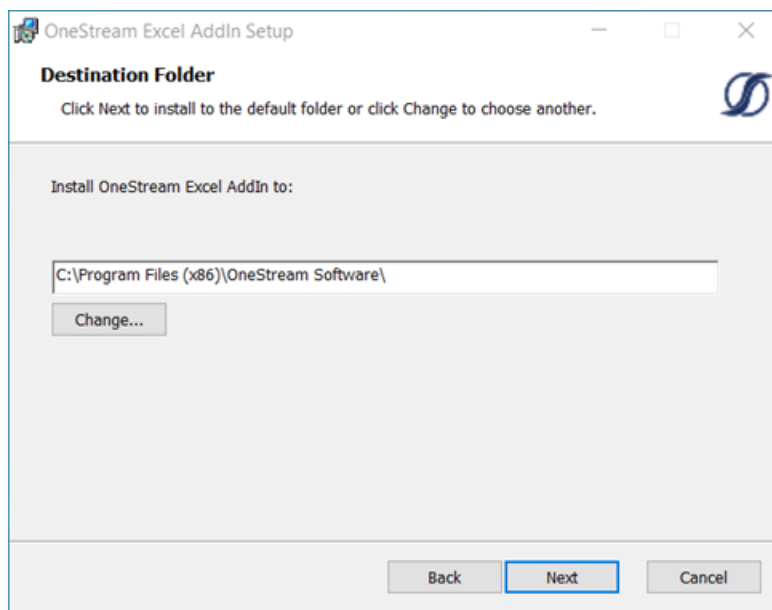
> **NOTE:** The default installation path is C:\Program Files\OneStream Software. Select Change to change the drive for the installation. For example, D:\Program Files\OneStream Software.

4. Click **Install** to start.
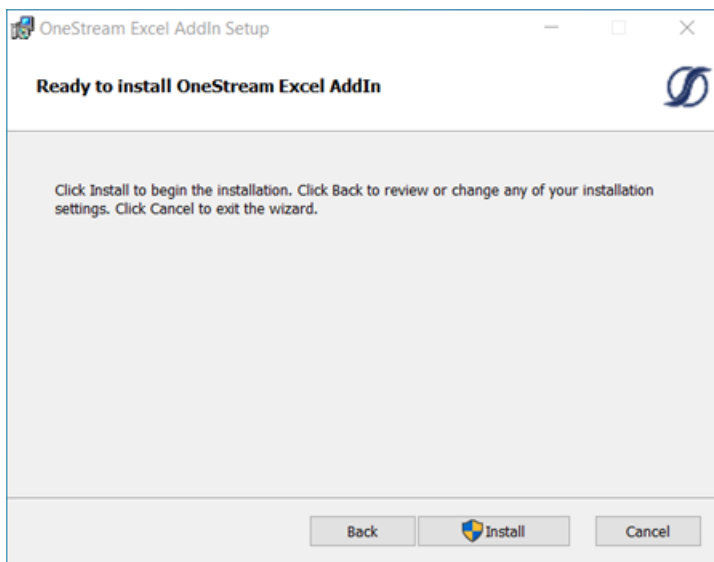
5. Click **Finish** to complete the installation.
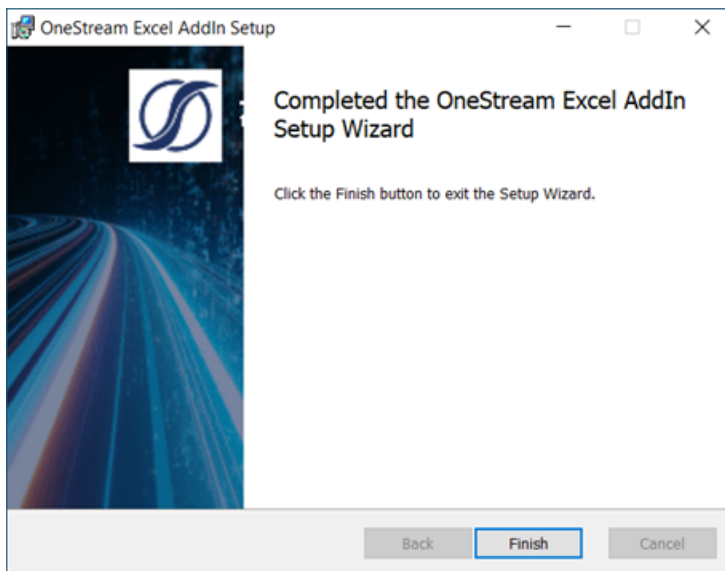
Deploying the OneStream Windows App with ClickOnce

About ClickOnce

The recommended deployment method for OneStream Windows App is Microsoft ClickOnce. Microsoft ClickOnce is provided for true web-based deployment and is supported on all Windows versions under mainstream Microsoft support.

ClickOnce is an integrated component of the Windows .NET Framework and it is a prerequisite to have the PC updated to OneStream's minimum version. See the *System and Architecture Guide*.

Deployments target a system-managed, user-specific folders. Changes are not made to C:\Program Files and registry settings, and no scripts are run.
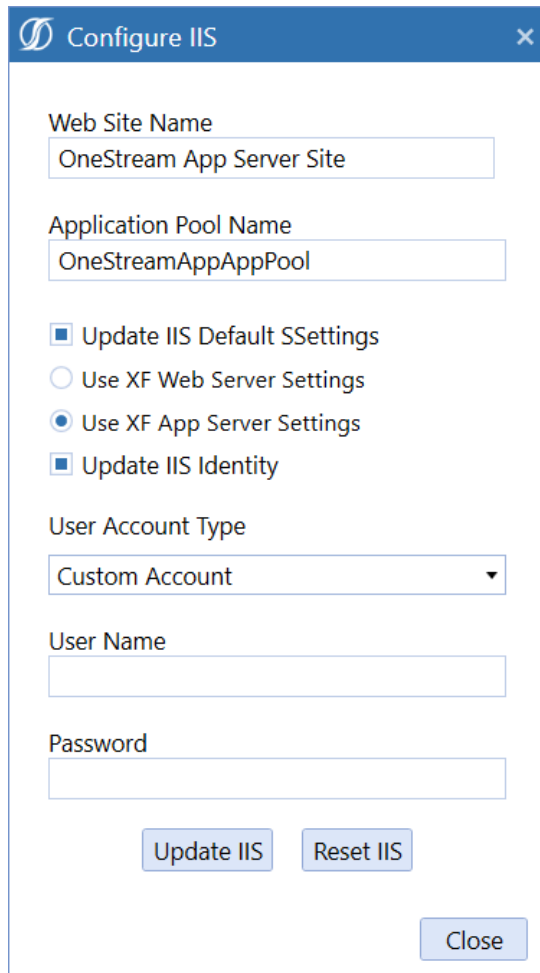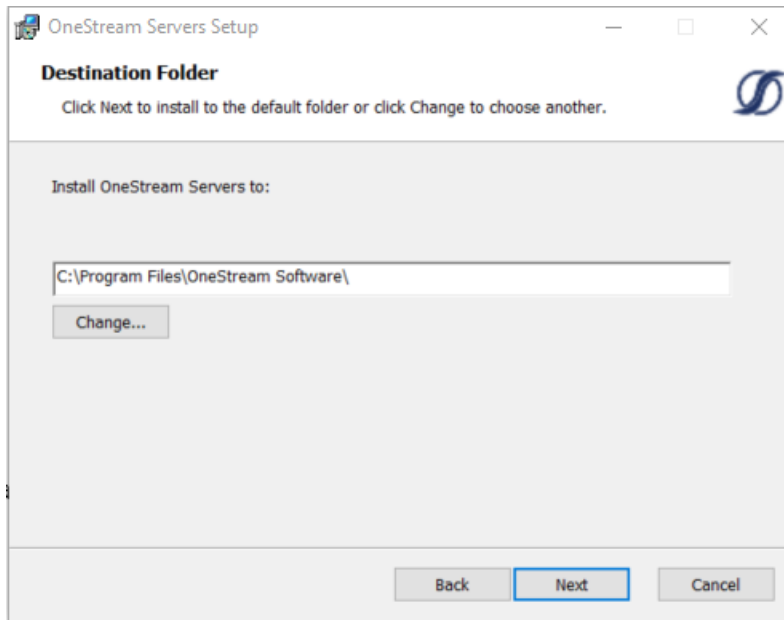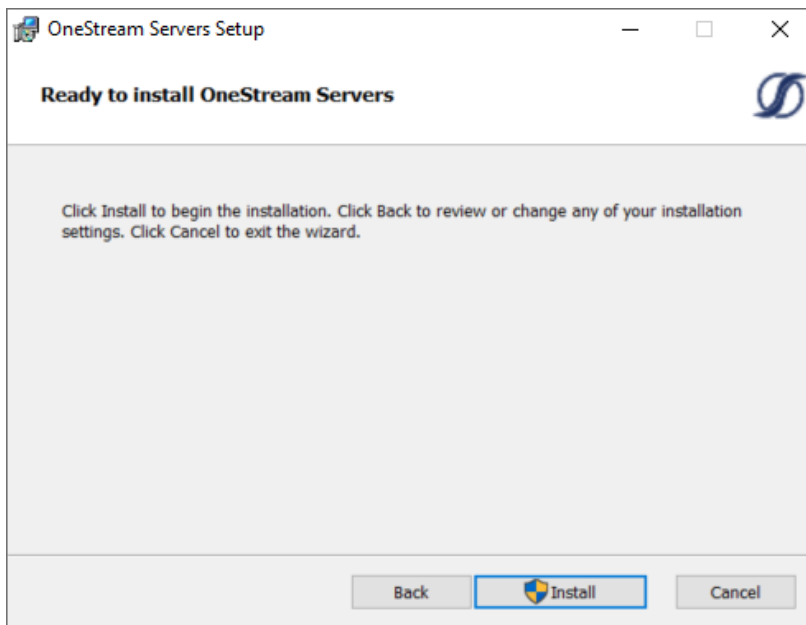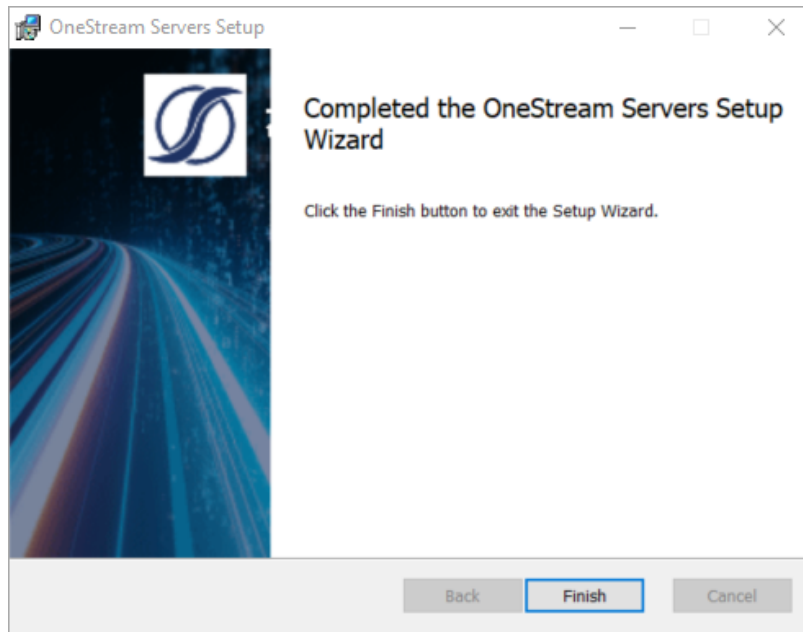
The deployment package ClickOnce generates cannot be modified. If deployed with ClickOnce, the OneStream app will update.

Web-Based Deployment

To deploy, users go to a standard website deployed via OneStream. The URL would be similar to: http://<servername>:50001/OneStreamWeb/Onestreamwindowsapp.aspx

After clicking a Run button from this page, the application will be downloaded (package size ~220 MB with ~325 files) and deployed in a trusted local user directory that does not require administrative rights (location controlled by Windows and .NET Framework and is located under C:/Users/<username>/AppData/Local/Apps/2.0/<Windows Assigned GUID>)

OneStream Windows App is then launched with the URL of the launching server pre-configured for the user to make that connection. A shortcut icon can then be saved on the user's desktop

Installing the Windows App

OneStream Desktop can be installed for the current user (per user) or for all users on the machine (per machine). Installing per user can be done by any user, and will not affect any other user of the machine. Installing per machine requires administrative privileges and makes the application available to all users on the machine. With this type of installation:

- The Logon screen is slightly different.

- When OneStream servers are upgraded, applications do not automatically upgrade as they do with a standard deployment.

- A new executable file with the latest version must be downloaded from MarketPlace and reinstalled.

- Users must enter the URL of the OneStream application server to connect.

To install:

1. Download `Setup.msi` from the MarketPlace.

2. Double-click `Setup.msi` to launch the Installation wizard.

3. Follow the prompts such as accepting the license agreement and selecting the setup type.

4. Click **Install**.

To launch the application, see "Launching the Windows App" in the *Design and Reference Guide.*

Install Multiple Desktop Applications

With OneStream Desktop 7.0.1, you can install the application alongside previous versions of the software. This is called a side-by-side install.

There are several circumstances when you may need to use more than one version of the application. The most common scenario is that your company is planning to upgrade to a newer version, so a test environment is created with the new version to support customer testing. To test this new version and maintain access to the current version, you need to install another instance of the application.

The desktop application can only communicate with the same server version. This ensures that all the features of the application have been properly tested, and that unsupported scenarios cannot be introduced by upgrading only a portion of the system.

Named Instances

You can install up to eight versions of the desktop application on a single machine. Each installation instance has a specific name embedded in the installer: 0, I2, I3, I4, I5, I6, I7, I8.

Each instance also has a corresponding name in the installation folder and shortcut. The default instance is named 0 and does not have a corresponding change to the installation folder or shortcut. For example, I2 has a shortcut name of OneStream Desktop (2).

NOTE: More than one user can have the same named instance installed.

NOTE: With a per machine installation, you must use a named instance that is unique across all users on a machine. It is possible that all eight instances may already be claimed and the installation attempt fails.

Installation

To install an additional Desktop Application version, use the provided PowerShell script (setup.ps1).

Per User Installation

When you install an additional instance of the Desktop Application per user it is a Per User Installation.

- To install, open a non-elevated PowerShell and run the provided script.

- The script provides a series of prompts to guide you through the installation process based on other existing installations.

NOTE: If there are no existing installations when the script is run, no message displays, and the installer launches with the default named instance.

Per Machine Installation

When you install an additional version of the Desktop Application for multiple users on a machine it is a Per Machine Installation.

- To install, open an elevated (admin) PowerShell and run the provided script.

- The script provides a series of prompts to guide you through the installation process based on other existing installations.

NOTE: If there are no existing installations when the script is run, no message displays, and the installer launches with the default named instance.
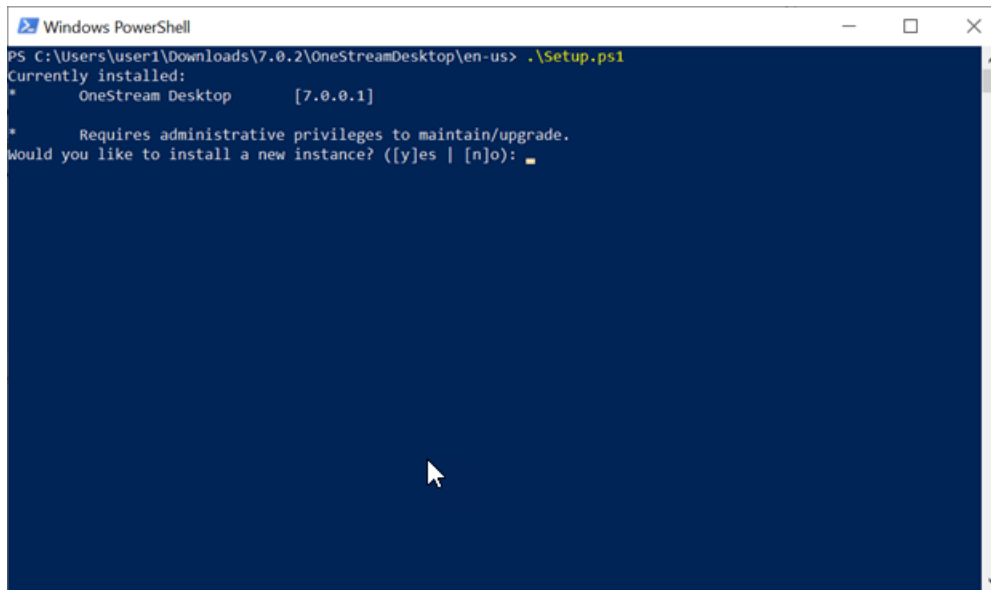
Script Prompt Examples

The following sections provide information for these scenarios:

- Per user installation if another per machine installation exists

- Per user installation if a combination of per machine and per user installation exists

- Per machine installation if another per machine installation exists

- Per machine installation and per user installations already on the machine

Per user installation if another per machine installation exists. The script is being run in a non-elevated PowerShell (per user). There is one existing per machine installation.
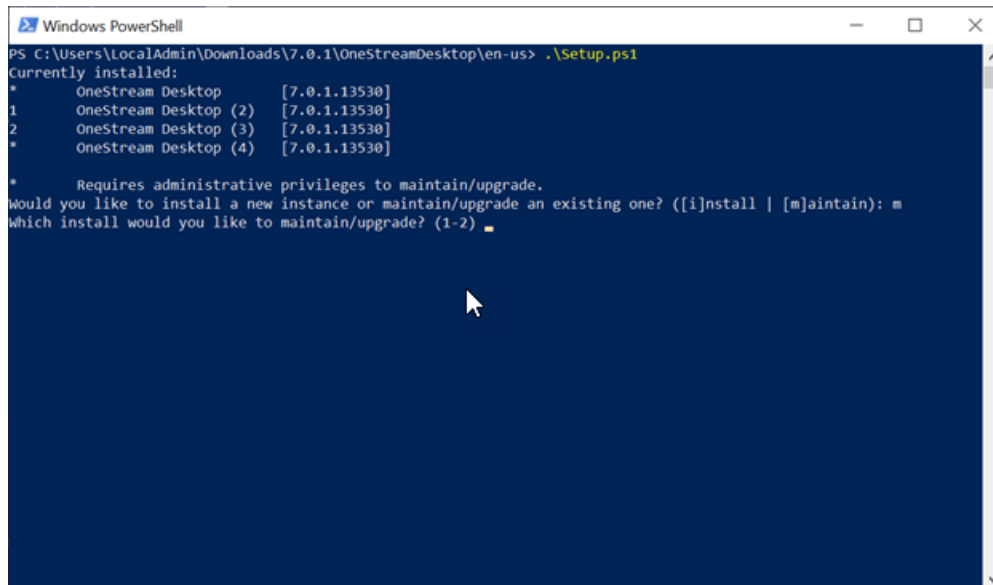
All known existing instances are listed. The list of known installations is the combination of all per machine installs, plus all per user installs for this user. The asterisk (*) signifies that the installation is a per machine install, and that it cannot be modified because that would require running in an elevated PowerShell.

The only operation that you can perform is to install a new instance of OneStream Desktop.

Per user installation if a combination of per machine and per user installation exists. The script is being run in a non-elevated PowerShell. There are a combination of both per machine and per user installs already on the machine.
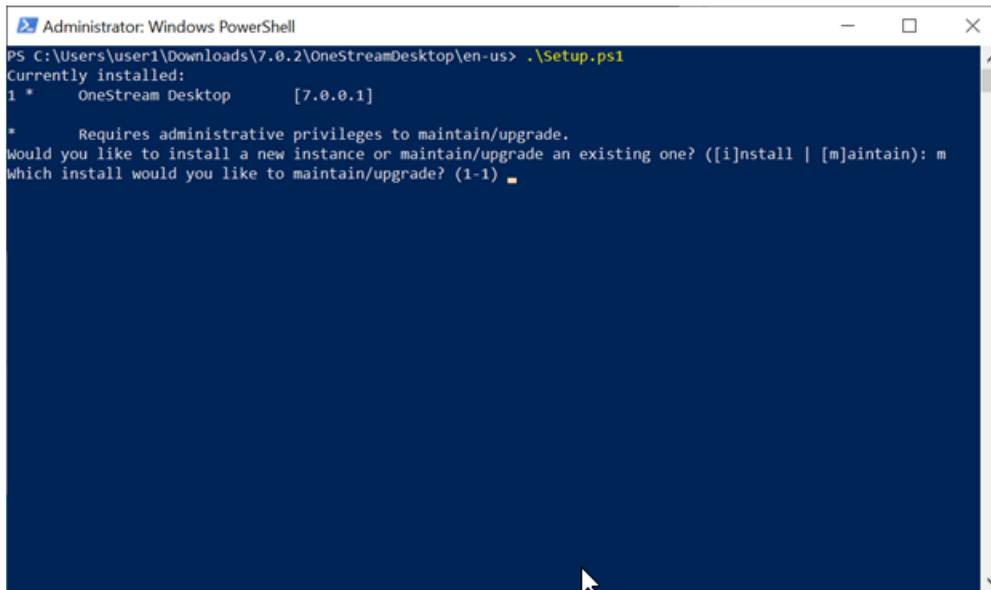
There are four existing installations of OneStream Desktop. The first and fourth installs are per machine, and cannot be modified in a non-elevated PowerShell. The second (I2) and third (I3) instances are per user, and can be modified in the current PowerShell. Type 'm' and press <Enter> to maintain an instance. You will be prompted to select which installations to modify. Only the instances that can be modified have a number at the beginning of the list to select. Type '1' and press <Enter> to launch the installer for the selected instance.

If you want to install a new instance, type 'i' at the first prompt and press <Enter> to launch a new per user installation.

*Per machine installation if another per machine installation exists.*This script is being run from an elevated PowerShell (admin). There is one existing per machine installation.
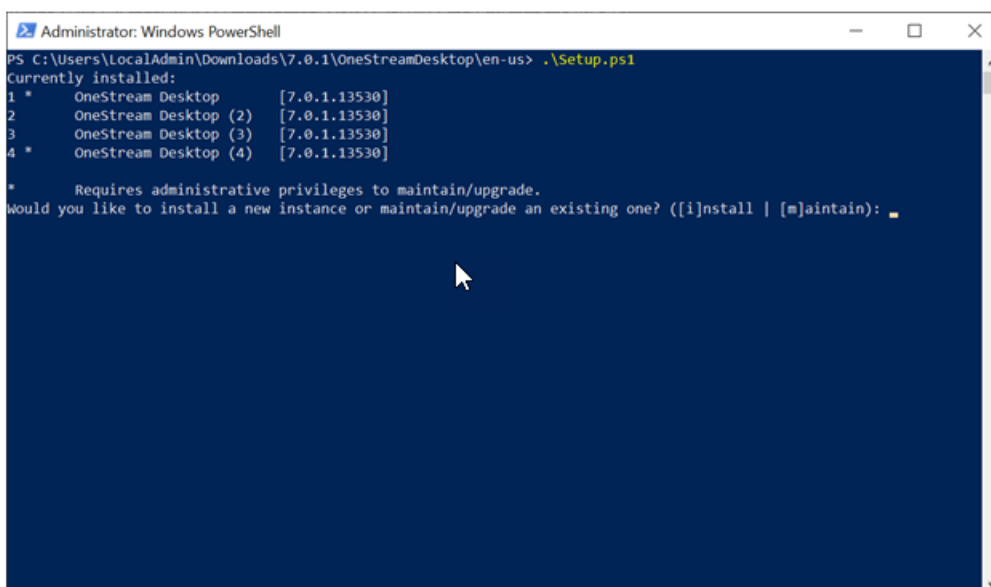
Because the PowerShell is elevated, the existing install can be maintained. If you select 'i' from the first prompt, it will install a new per machine instance. If you select 'm' you will be prompted to select the instance to maintain and then launch the installer for that instance.

*Per machine installation if another per machine installation and a per user install exists.*This script is being run from an elevated PowerShell (admin). There is a combination of both per machine and per user installs already on the machine.

> **NOTE:** All instances are numbered, meaning in this elevated PowerShell, any of the installs can be modified. The per machine installs are marked with an asterisk (*).

Advanced Installation

The OneStream Desktop installer is a standard MSI file, and common command-line attributes are available. You can perform advanced installations using the msiexec.exe tool that is deployed with Windows operating systems.

To perform a silent installation, use the `/qn` command-line parameter. This triggers an install that does not show a user interface, and installs using defaults.

To install a specific named instance you must set two properties. `MSINEWINSTANCE=1` is always required, regardless of which named instance is installed. The second property is `TRANSFORMS=":<instance name>"`, where <instance name> is I2-I8.

There are a number of command parameters that can be used, followed by the name of the MSI file. The `/i` command is to install, and the `/x` command is to remove.

The `/l` parameter, followed by a log file name, enables logging. There are various levels of verbosity, which you can find in the msiexec documentation. The most verbose option would be `/l*v <file.log>`.

The following example would launch the installer with the full user experience, and log everything to a file:

```
msiexec.exe /i setup.msi /l*v setup.log
```

This example performs a silent install of the 3rd named instance (or maintains it if already installed):

```
msiexec.exe /i setup.msi MSINEWINSTANCE=1 TRANSFORMS=":I3" /qn
```

> **NOTE:** Don't forget the colon in the TRANSFORMS property.

> **NOTE:** If you are using PowerShell and press the **Tab** key to auto complete the name of the MSI, it will insert '`.\`' in front of your MSI file name. This will not work and those characters must be removed.

> **NOTE:** Installation scope is still relevant when performing an advanced installation. To install per machine, the advanced command-line must be performed in an elevated context.m,kl

Silent Install

To silently install from the command prompt:
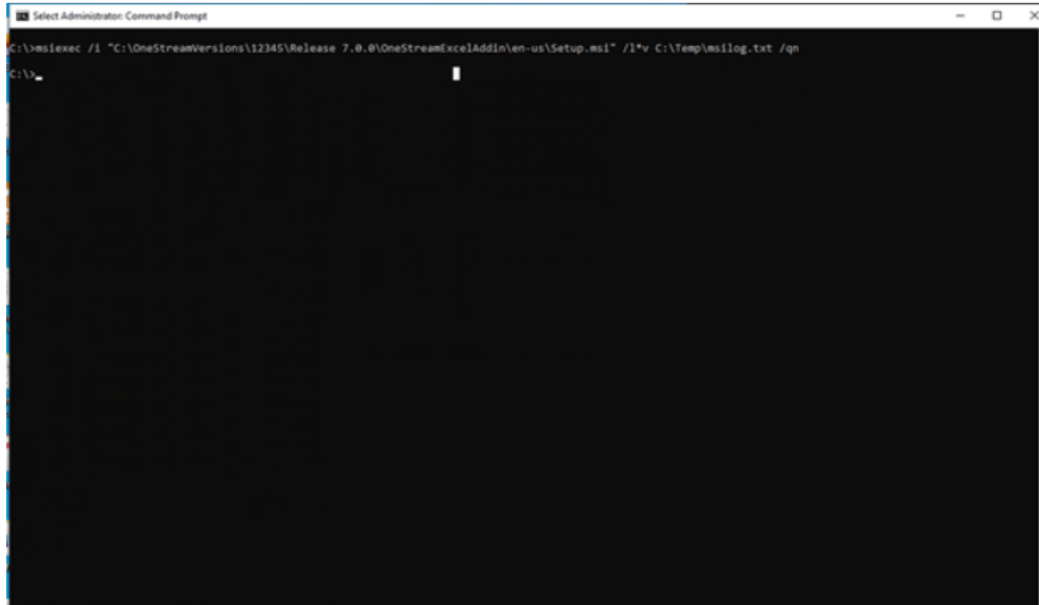
For Excel Add-In enter:

```
C:\>msiexec /i "C:\OneStreamVersions\12345\Release
7.0.0\OneStreamExcelAddin\en-us\Setup.msi" /qn
```

For Excel Add In with a log file:

```
C:\>msiexec /i "C:\OneStreamVersions\12345\Release
7.0.0\OneStreamExcelAddin\en-us\Setup.msi" /l*v C:\Temp\msilog.txt /qn
```

For Desktop Application:

```
C:\>msiexec /i "C:\OneStreamVersions\12345\Release
7.0.0\OneStreamDesktop\en-us\Setup.msi" /qn
```

For Desktop Application with a log file:

```
C:\>msiexec /i "C:\OneStreamVersions\12345\Release
7.0.0\OneStreamDesktop\en-us\Setup.msi" /l*v C:\Temp\msilog.txt /qn
```

For more than one Desktop Application or side-by-side up to eight instances:

```
C:\>msiexec /i "C:\OneStreamVersions\12345\Release
7.0.0\OneStreamDesktop\en-us\Setup.msi" MSINEWINSTANCE=1 TRANSFORMS=":I2"
/l*v C:\Temp\msilog.txt /qn

C:\>msiexec /i "C:\OneStreamVersions\12345\Release
7.0.0\OneStreamDesktop\en-us\Setup.msi" MSINEWINSTANCE=1 TRANSFORMS=":I3"
/l*v C:\Temp\msilog.txt /qn

C:\>msiexec /i "C:\OneStreamVersions\12345\Release
7.0.0\OneStreamDesktop\en-us\Setup.msi" MSINEWINSTANCE=1 TRANSFORMS=":I4"
/l*v C:\Temp\msilog.txt /qn

C:\>msiexec /i "C:\OneStreamVersions\12345\Release
7.0.0\OneStreamDesktop\en-us\Setup.msi" MSINEWINSTANCE=1 TRANSFORMS=":I5"
/l*v C:\Temp\msilog.txt /qn

C:\>msiexec /i "C:\OneStreamVersions\12345\Release
7.0.0\OneStreamDesktop\en-us\Setup.msi" MSINEWINSTANCE=1 TRANSFORMS=":I6"
/l*v C:\Temp\msilog.txt /qn

C:\>msiexec /i "C:\OneStreamVersions\12345\Release
7.0.0\OneStreamDesktop\en-us\Setup.msi" MSINEWINSTANCE=1 TRANSFORMS=":I7"
/l*v C:\Temp\msilog.txt /qn

C:\>msiexec /i "C:\OneStreamVersions\12345\Release
7.0.0\OneStreamDesktop\en-us\Setup.msi" MSINEWINSTANCE=1 TRANSFORMS=":I8"
/l*v C:\Temp\msilog.txt /qn
```

> **NOTE:** The default instance is 0, and the subsequent installs are I2, I3, I4, I5, I6, I7, I8

Silent Uninstall

To silently uninstall from the command prompt:
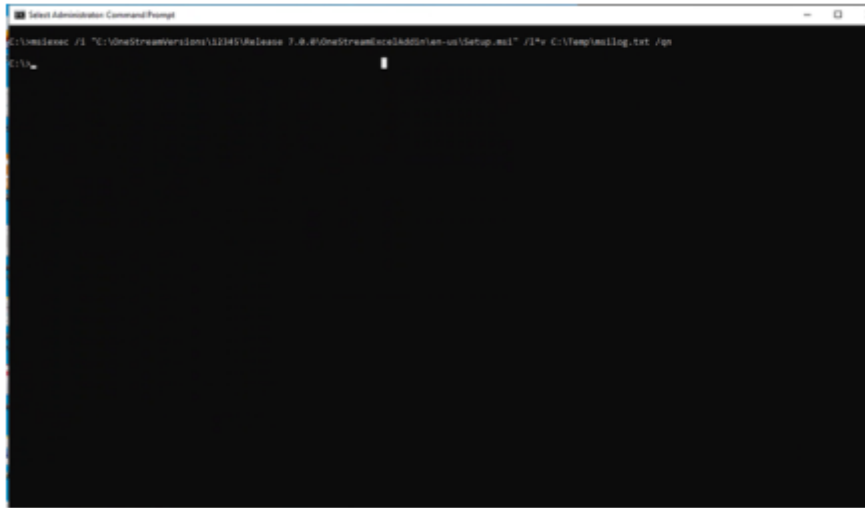
`C:\>msiexec /x "C:\OneStreamVersions\12345\Release 7.0.0\OneStreamExcelAddin\en-us\Setup.msi" /qn`

Installing the Studio Client Package

The OneStream Studio client is recommended for users that need to create report layouts. OneStream Studio is a Windows client that connects to the OneStream servers through the OneStream web services. See Excel Add-In Installation in the previous section as the steps are similar to installing and updating the Studio Client.

Upgrading or Downgrading the Excel Add-In or Studio

Upgrade or Downgrade Client via OneStream in the Browser

Users receive a compatibility error when they connect to a OneStream web server with the Excel Add-in or OneStream Studio that has a different version from the local version. Given the change in the .NET Framework version and other changes starting in 4.2, you cannot downgrade a release prior to 4.2.

Users can upgrade the client in two ways:

1. Use the Client Updater functionality in the OneStream Windows App (see next section).

2. Uninstall and re-install the Excel Add-in or OneStream Studio.

Upgrade or Downgrade the Client via OneStream Windows App

The Client Updater is used to retrieve updated software from the OneStream server for the Excel Add-In and Studio client programs when the version being used does not match the version of OneStream found on the server being connected. Note that to perform the update, the user needs to be able to write to the installation folder. From this page, the user can first save all OneStream work and then restart OneStream using elevated Windows privileges using the Restart as Windows Administrator button.

Information regarding the current version is displayed at the top of the window, as well as the module being reviewed, the location of the installation folder and the version status all appear in this window.

Example of versions that match:
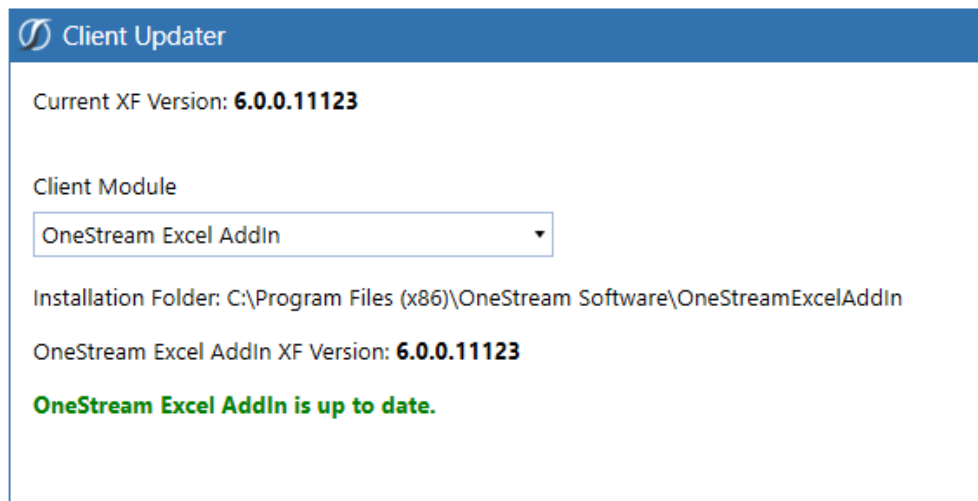
Example of versions that do not match:
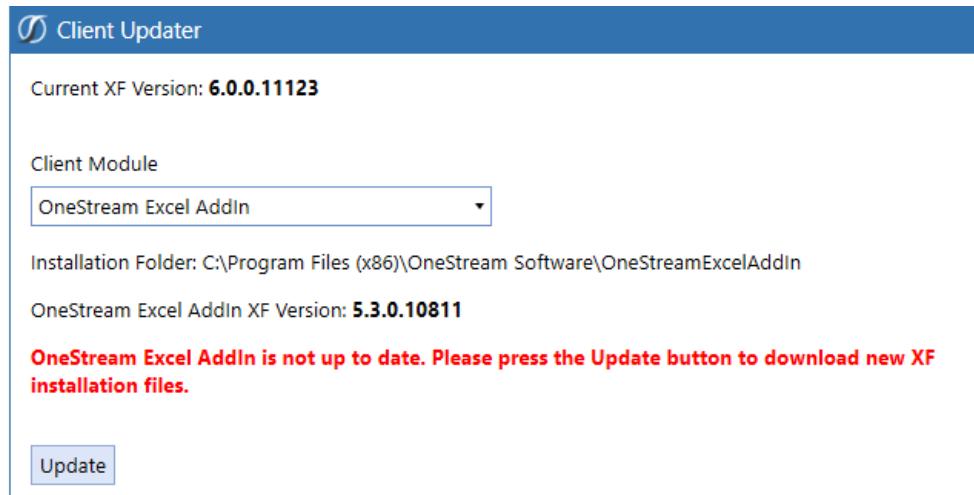
Client Module

This field is used to select which application (OneStream Excel Add-In; OneStream Studio) needs to be compared to the current version of each that are currently installed on the user's desktop. Click the selection arrow and select the appropriate application.

If the versions do not match, click the **Update** button, then click **OK**.

> **NOTE:** If any versions of Excel or Studio are open, they must be closed before clicking OK to proceed.

> **NOTE:**
> A backup folder with files for the outdated version is automatically created and saved as part of the update process.   It can be found in the same location as the newly updated version folder.

If this functionality has been disabled, the following message appears when trying to update those applications. "The Client Updater has been disabled by your System Administrator. Please use OneStream's full client installation program, or see your System Administrator."

Configuring System Components

The following sections describe how to configure OneStream to run in a specific network infrastructure. The topics are presented in a logical order that ensures the prerequisites of each step have been accomplished.

1. OneStream's Configuration Files and Tools

2. Creating the Application Server File Share Root Folder

3. Creating Service Accounts and Permissions

4. Creating Database Connections and Schemas

5. Configuring Application Server(s)

6. Configuring Web Server(s)

OneStream's Configuration Files and Tools

OneStream utilizes three types of XML files to store its configuration information. Two of the configuration files are OneStream specific configuration files (XFWebServerConfig.xml & XFAppServerConfig.xml) and one of the files is a generic Microsoft ASP.net configuration file (Web.Config).

The generic Web.Config file is optionally used to store an alternate path to the OneStream specific configuration files. For example, if multiple web servers want to share a common XFWebServerConfig.xml file, the common path should be set in the Web.config file of the OneStream web site on each web server. The same process applies to application servers that need to share a common XFAppServerConfig.xml file.
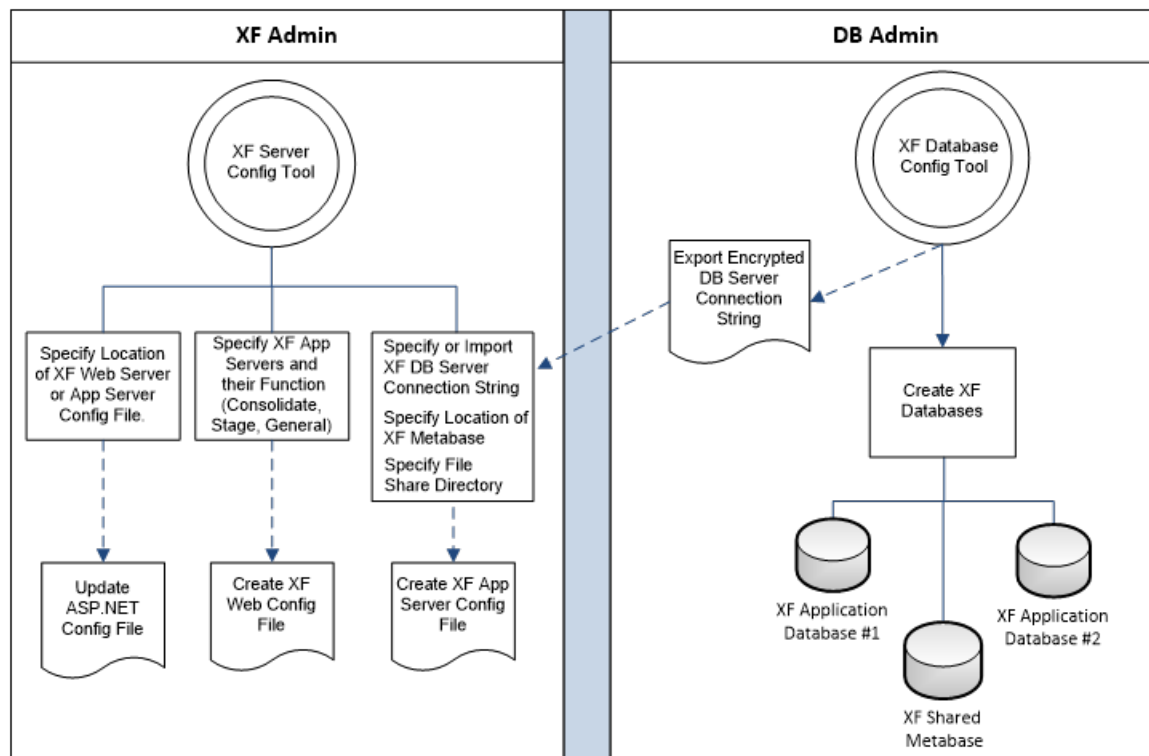
XFWebServerConfig.xml

This file is a proprietary OneStream xml file that contains configuration information specific to OneStream web servers. It should also be noted, that the web server configuration file defines the application server pool that is used by one or more web servers. This file can be accessed and maintained using the Server Configuration Tool.

File Contents

- Specifies application server pool

- Defines application server usage

Stage
Performs data loading and transformation services only

Consolidation
Performs consolidations and calculation services only

General
Performs all services

Data Management
Performs data-related functions that can be very hardware-intensive.

Default Location

During the web server installation, the OneStream server installation package creates an empty version of the XFWebServerConfig.xml file in the App Data folder in the virtual directory created for the OneStream web server application.

Default Web Server XFWebServerConfig.xml Path

C:\Program Files\OneStream Software\OneStreamWebRoot\OneStreamWeb\App_
Data\XFWebServerConfig.xml

XFAppServerConfig.xml

This file is a proprietary OneStream xml file that contains configuration information specific to OneStream application servers. This file can be accessed and maintained using the OneStream Server Configuration Tool.

File Contents

- Database server connection information (Optionally Encrypted)

- File share directory location

- Application server threshold/limit settings

- Application server threading model settings

- System culture settings

- Environment settings

- Monitoring settings

- Task Load Balancing settings

- Azure subscription settings

- Azure EDU level settings

- Server Sets settings

Default Location

During the application server installation, an empty version of the XFAppServerConfig.xml file is created in the App Data folder in the virtual directory created for the OneStream application server application.

Default Application Server XFAppServerConfig.xml Path

C:\Program Files\OneStream Software\OneStreamAppRoot\OneStreamApp\App_
Data\XFAppServerConfig.xml

Web.Config

This file is a standard Microsoft ASP.Net configuration file that exists on the web and application server. This file specifies settings that control how a Microsoft ASP.Net web site behaves. However, application specific information can be stored in this file to allow products running in an ASP.Net/IIS web site to store configuration information.

File Contents

OneStream optionally uses this file, on both the web and application servers, to store an alternative folder path for the OneStream specific configuration files. This setting allows web and/or application servers to create a shared folder for configuration files that can be used to share configuration files between servers. This file can be accessed and maintained using the Server Configuration Tool.

Default Location

During the web server and Application server installation process, the OneStream server installation package will create OneStreamWeb or OneStreamApp ASP.net virtual directory. The Web.Config file exists at the root of this virtual directory.

Default web server Web.Config path

C:\Program Files\OneStream Software\OneStreamWebRoot\OneStreamWeb\Web.Config

Web Server Security Hardening

The Web.config file is typically reviewed during server security scans by IT security teams and auditors. Numerous security scanning tools may be used during these audits. OneStream recommends making the changes identified in "Appendix 7: Web.config Hardening Process Overview" on page 170 for optimal security and consistent scanning results.

[Encrypted Database Connections].xml

OneStream allows for optional separation of duties between application server administrators and database server administrators. Database connection information can be protected limiting the use of the OneStream Database Configuration Utility to database administrators.

This utility enables database administrators to create and maintain connections and schemas in the relational database system and simply provide an xml file containing encrypted database connection information to application server administrator. The application server administrator can simply import the contents of the encrypted database connection xml file into the XFAppServerConfig.xml file using the OneStream Server Configuration Tool.

See Appendix 6: Upgrade Process Overview for more details.

Creating the Application Server Share Root Folder

Application servers require a shared folder that they can use as a file system workspace. Application servers need this workspace to provide a place for users to upload and download documents, store batch processing files, and write system logs. This information is temporary in nature and not critical to application function. The application servers rely on its presence, but from a backup and recovery perspective this information is not critical. All critical information is stored in the OneStream relational databases.

This file folder should be created on a file share that all application servers should reference. The path to this folder will be referenced during the application server configuration process.

Creating Service Accounts and Permissions

OneStream uses a minimum of three server processes that require service accounts for system communication. The following information defines the required account permissions by server and serves as a guide to help configure OneStream's service accounts in accordance with a company's network and data center policies. By default, the IIS application pools created for the OneStream web and application servers run under the NT AUTHORITY\NETWORK SERVICE account. For information on creating and managing service accounts see this Microsoft TechNet article: http://technet.microsoft.com/en-us/library/dd548356(WS.10).aspx.

OneStream server components communicate using the Windows Communication Foundation (WCF). This makes inter server communications simple and flexible. Configuring the product to work within firewall constraints is more straight forward than legacy DCOM based applications.

Web Server Account

The service account used to run the OneStream web server IIS App Pool requires minimal privileges. The default NT AUTHORITY\NETWORK SERVICE has sufficient permissions to run the OneStream web server. If you cannot use NT AUTHORITY\NETWORK SERVICE, use a limited permission domain account, other managed service accounts and virtual accounts such as IIS AppPool\OneStreamWeb instead. This account should be created in the same domain as that of the OneStream Web Server.

Application Server Account

The service account used to run the OneStream application server IIS App Pool requires database access privileges and file share privileges. OneStream recommends that a dedicated service account be created to run the OneStream application server IIS application pool. This dedicated service account should be created in the same domain as that of the OneStream Application Server. Adding privileges to the default NT AUTHORITY\NETWORK SERVICE account may create a security risk because other services using this account will also gain these privileges.

Database Logon Permissions

The account that OneStream uses to access SQL Server should be granted the Public and Sysadmin privileges. These privileges are required to allow the OneStream server process to create and maintain application database schemas. Each OneStream application is contained in its own database schema.

Depending on how SQL Server security has been configured these privileges will either need to be assigned to the service account being used to run the OneStream application server, (If database uses Windows integrated security) or to a standalone SQL Server account if (If database does NOT use Windows integrated security).

Database Access Permission Note
SQL Server privileges may be reduced to a more restrictive level based on the organizations database security policies. At a minimum the account used to access SQL Server must be able to Insert, Update, Delete, Create / Drop tables, and execute a bulk insert via ADO.Net bulk copy libraries.

OneStream can prevent application databases being created in the product, ensuring control by a corporate database management team.

Minimum SQL Server Account Permissions

App Databases
DBOwner and Public

Framework
DBOwner and Public

Master
DBOwner and Public

File Share Permissions

The OneStream application server IIS App Pool account must have Full Control privileges to the application server file share root folder. The application server uses location to create and delete files and folders for common task such as uploads, downloads, batch processing, and logging.

Creating Database Connections and Schemas

OneStream uses three database schema types, Framework, Application and State. The sections below describe the two schema types in detail and provide instructions on how to use the OneStream Database Configuration Tool to create connections and schemas.

Note: If upgrading from a 4.x.x version to a 6.0 version, Database Schema updates will need to be implemented as part of the upgrade. Full Database Backups are required for any databases you will be updating (Framework and any Application databases). See Appendix 6 Upgrade Process Overview.

SQL Database Considerations

To create an Azure ready database, the Use File Groups property must be unchecked when creating the Application Database Tables. Doing this will create a log file and a database file and ensure that Azure can be supported at any time. This option applies to SQL 2014 and higher. A new application database will need to be created to utilize Azure in the future.
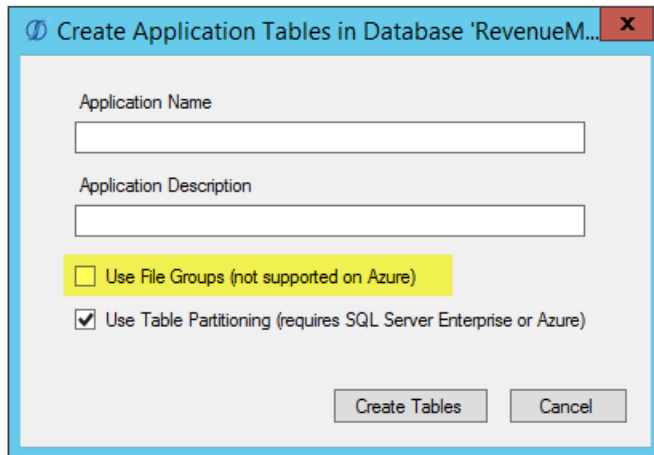
Microsoft does offer a utility to migrate from a classic SQL Enterprise database using File Groups to a non-File Group. The information can be found here: https://azure.microsoft.com/en-us/blog/migration-cookbook-now-available-for-the-latest-azure-sql-database-update-v12/

Framework Database

Framework can be thought of as the system database. This database contains system level information and servers as a controlling database or gateway to accessing application data. OneStream always connects to just one Framework database which is specified during the application server configuration process. The Framework database maintains the list of application databases that are associated with the OneStream instance.

The Framework Database contains the following data elements:
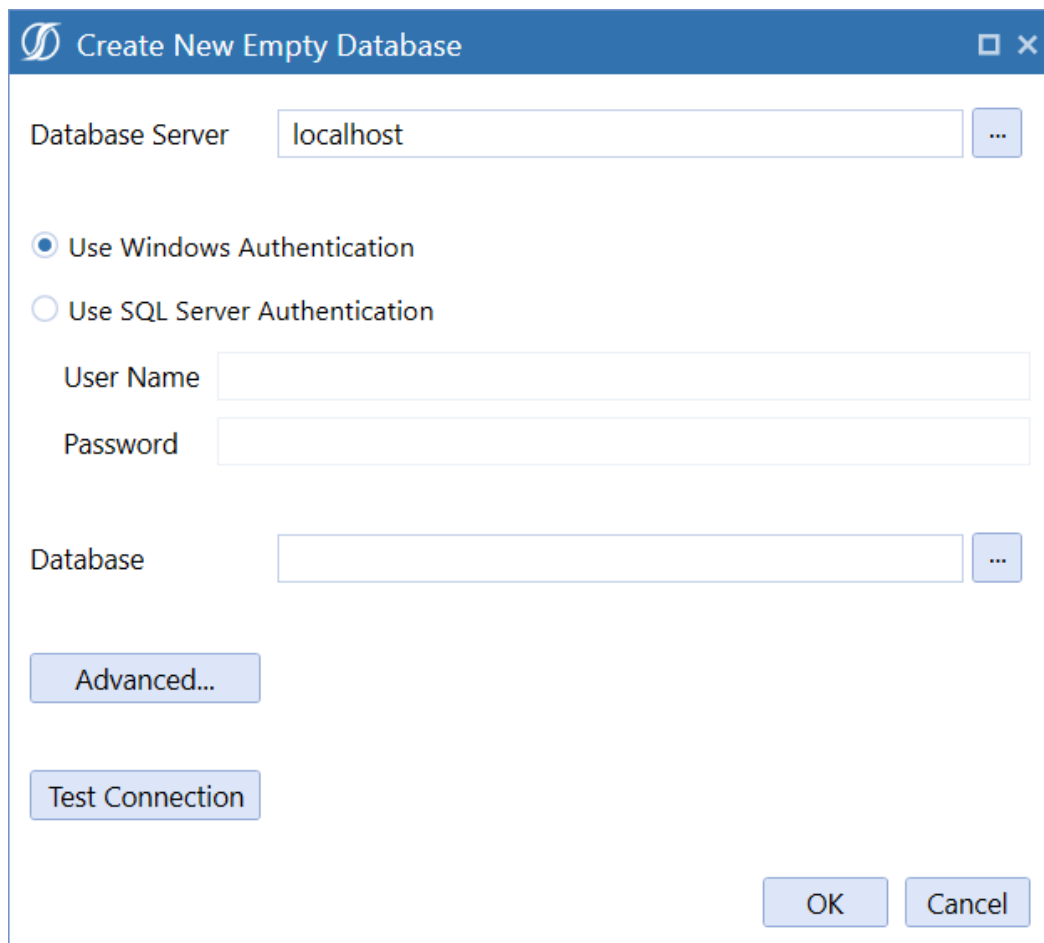
- Users

- Groups

- System Roles

- Environment Metrics

- Task Activity Log

- Error Log

- Application Definitions

- System Level Report and Dashboard Definitions

Creating a Framework Database

To create a new Framework database schema, launch the OneStream Database Configuration Tool on a machine that can connect to the SQL Server instance that will host the database. Next, follow the steps listed below:

1. Select the top item titled Databases in the left tree control.

2. Right-click the node to show the context menu.

3. Select **Create New Empty Database** from the menu.

4. Use the standard database creation dialog to specify the Database Server Name, Authentication Method, and the Database Name.

5. Click **Advanced** and set the Connect Timeout to 60 and the Max Pool Size to 5000.

6. Click **OK**.

7. Select the newly created database in the left tree control.

8. Right-click the selected item to show the context menu.

9. Select **Create Framework Database Tables** from the menu.

10. Choose Use File Groups and/or Use Table Partitioning and click **Create Tables**. These are selected by default.

11. Select this database in the list under Databases.

12. Right-click and select **Apply OneStream License**.

13. Paste your OneStream license key that was sent to your company in this field and click **Save**.

14. Restart IIS to accept the license key.

    The new Framework database is ready for use.

Application Database(s)

Application can be thought of as a data content database. Each OneStream instance can access many Application databases. An Application database contains information specific to a set of Financial Models and the Workflows used to manage the models. The system can be configured so that users can create new Application databases in the product or this task can be restricted, so that a database administrator must use the OneStream Database Configuration Tool to create new Application database schemas which are then attached to the Framework database.

An Application database contains the following data elements:

- Application Roles

- Cube and Dimension Definitions

- Workflow Definitions

- Data Transformation Definitions

- Data Quality and Certification Definitions

- Staged Data

- Cube Data

- Certification and Sign-Off Data

Creating an Application Database
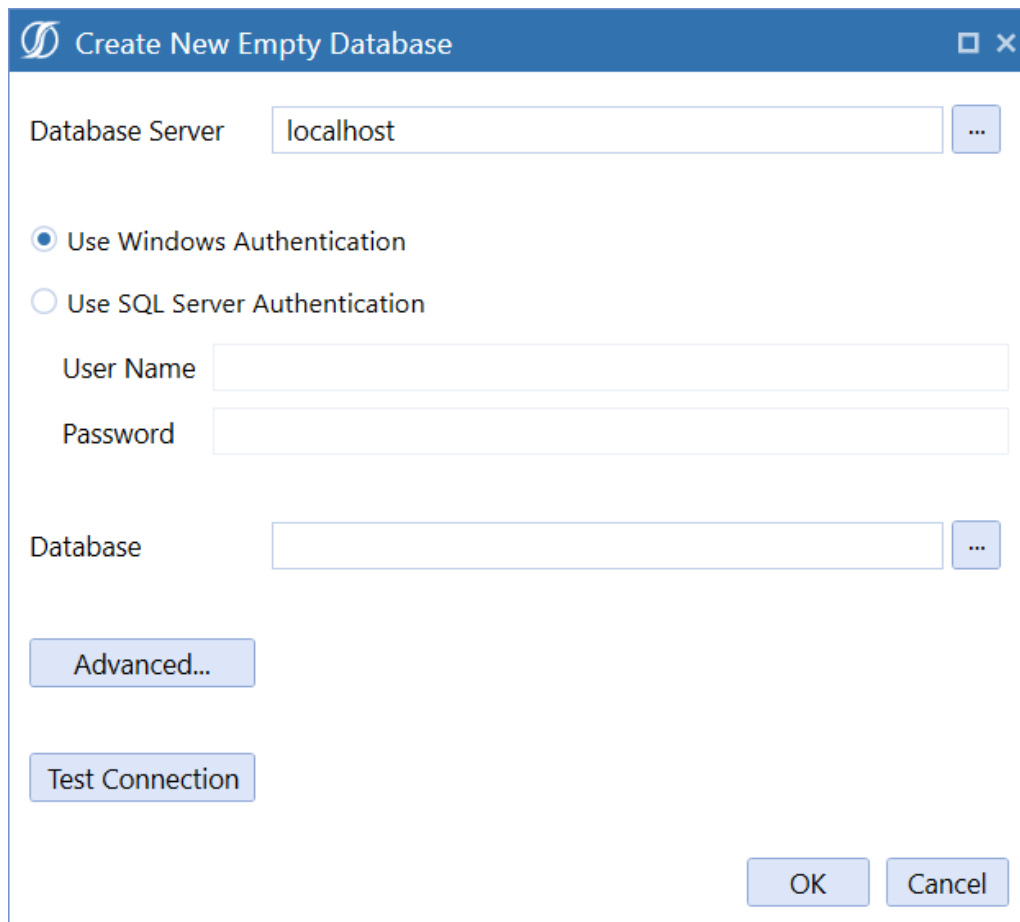
Application databases can be created directly in OneStream or by using the OneStream Database Configuration tool.

First, launch the Database Configuration Tool on a machine that can connect to the SQL Server instance that will host the database.

Next, follow the steps listed below:

1. Select the top item titled Databases in the left tree control.

2. Right-click the node to show the context menu.

3. Select **Create New Empty Database** from the menu.

4. Use the standard database creation dialog to specify the Database Server Name, Authentication Method, and the Database Name.

5. Select the newly created database in the left tree control.

6. Right-click the selected item to show the context menu.

7. Select **Create Application Database Tables** from the menu.

8. This launches the following dialog.

9. Specify the **Application Name** and **Description**.

10. Select Use File Groups and/or Use Table Partitioning. These are selected by default.

11. Click the ellipsis to select a pre-configured Time Dimension xml file
    If a Time Dimension Definition is not specified, the application database will default to the
    Standard Time Dimension Type. This creates a Monthly Time Dimension and stores the
    data by month in the data tables. All applications created prior to Version 4.1.0 are using
    this Time Dimension Type. Refer to Time Dimensions in the Design and Reference Guide
    for more details.

12. Click **Create Tables**.

13. (Optional) Rename Application: If the new Database Connection is a copy of an existing
    OneStream Application database accessed in the same OneStream environment, you
    must give this application a new name. Right-click this database and select Rename
    Application. Type a unique Application Name. The Application Description field is optional
    and leave Create New Application Unique ID as checked. Click **Rename**.

State Database

State can be thought of as a temporary database. OneStream uses the State database to store temporary report state information. This schema can be deleted and recreated at any point because the information contained in the database is only relevant to a user's current session.

Creating a State Database

To create a new State database schema, launch the OneStream Database Configuration Tool on a machine that can connect to the SQL Server instance that will host the database. Next, follow the steps listed below:

1. Select the top item titled Databases in the left tree control.

2. Right-click the node to show the context menu.

3. Select **Create New Empty Database** optio from the menu.

4. Use the standard database creation dialog to specify the Database Server Name, Authentication Method, and the Database Name.

5. The tables for this database are created on demand by OneStream.

    The new State database is ready for use.

Exporting Database Connections

OneStream uses the Database Configuration Tool to manage and create database connections in isolation from the server configuration process.   Once a database or database connection has been created in the OneStream Database Configuration Tool, a file can be exported from the tool containing a list of encrypted database connections.

Each OneStream application server configuration file stores a database connection string pointing to the host SQL Server. This connection is then used in combination with Framework and Application database names to establish connection to databases.

To make process of assigning a database connection to an application server, the exported database connection file can be imported into the OneStream Server Configuration Tool.

Managing Data Record Storage

OneStream applications can have a monthly or weekly Time Dimension. The pre-fixed Time Dimension Types determine if a particular application is monthly or weekly and the type of calendar used (e.g., 445, 454, etc.) A Custom Time Dimension Type allows users to specify the number of months in a quarter and the number of weeks in a month and can only be applied to new applications. All applications created prior to Version 4.1.0 are using the Standard Time Dimension Type which creates a Monthly Time Dimension and stores the data by month in the data tables. Standard monthly applications can be converted into a weekly application by copying the application's data into binary data storage.

The implications of this action are very serious because of the effect this can have on existing reports, objects using specific time logic and how data is processed going forward. OneStream requires an application review before any existing application converts from standard to binary. Contact OneStream Support to further discuss this process.

Configuring Application Server(s)

Configuring application servers requires that the following be stored in XFAppServerConfig.xml:

- The path to the file share root folder that will serve as a workspace for the application server.

- The pool of database server connections, schema names and server connections for the Framework and State databases.

All other configuration values can use standard default values.

Using the Server Configuration Tool

1. Open the OneStream Server Configuration Tool.

2. Select **File** > **Open Application Server Configuration File** and browse to the default XFAppServerConfig.xml in the application server's virtual directory: C:\Program Files\OneStream Software\OneStreamAppRoot\OneStreamApp\ App_Data\XFAppServerConfig.xml.

3. When the file is open, define the following settings to configure the application server:

Application Server Configuration Settings

OneStream Environment

Define the following settings to customize your environment.

- **Environment Name and Color:** Enter the name to be displayed (in white) for the environment. You can enter up to 150 characters. Specify a provided environment color or enter a hex value to display the name on a colored background. For example:

- **Can Use Client Updater: True** enables the Client updater to upgrade a user's version of Excel Add-In and Studio. **False** disables upgrades to Excel and Studio using the Client update. If disabled, users get a message indicating functionality was disabled by their System Administrator.

- **Can Use Administrator User: True** activates the generic Administrator user account. **False** disables the Administrator logon. If the other Admin accounts were deleted, set this to **True** to support logon**.**

- **Use Detailed Logging: False** omits internal error language and information from the error log.

- **Enable Help:** Select **True** to display a help icon that launches the official documentation set. Select **False** to not display a help button.

- **Enable File Share Uploads: True** enables authorized users to upload or edit files or folders in the File Share using the OneStream File Explorer. When **False**, users and Administrators can only browse, and not upload or edit files and folders. Users get a security error when writing or editing files or folders using API or another method.

Logon Agreement Type and Message: To display a specific message after a user logs on, select **Custom** and enter the message text.

Environment Monitoring
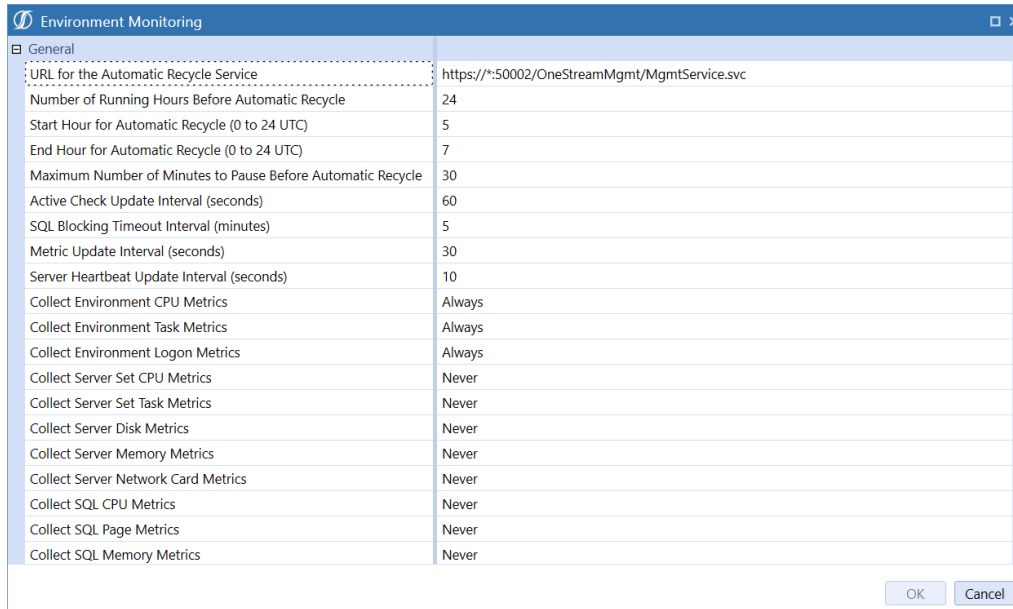
Use Environment Monitoring settings to define the real time update frequency, and the KPIs and metrics to monitor. These metric categories help you manage and optimize applications and the application environment:

- Environment

- Application server

- Database server

- Server set

Us the Environment page to evaluate and monitor the environment, isolate bottlenecks, look at properties and configuration changes, and scale in/out application servers and database resources.

This section is used to specify how often metrics are collected and what metric types are collected:

URL for the Automatic Recycle Service

Used to specify the address of the recycle management service. The protocol for the address should be set to however the service is deployed (https or http) and the port (default is 50002). The asterisk will force the service to use the fully qualified domain name of the executing server.

Number of Running Hours Before Automatic Recycle

Default is 24, which means once a day, the server will recycle. Automatic Recycling allows Application Servers a chance to recycle, which is a recommended practice. These first four settings control this behavior.

Start Hour for Automatic Recycle (0 to 24 UTC)

Default is 5, which means 05:00 UTC time. This is the earliest time in a day when a server can automatically recycle. It is best to set this and the End Hour to be a range of time with the lowest amount of Application Server activity.

End Hour for Automatic Recycle (0 to 24 UTC)

Default is 7, which means 07:00 UTC time. This is the latest time in a day when a server can automatically recycle.

Maximum Number of Minutes to Pause Before Automatic Recycle

Default is 30. This means that when it is time to recycle a server automatically, it will first pause from accepting more server tasks, but allow for existing assigned tasks to complete processing for 30 minutes before recycling. If there are no active tasks for this server, it will recycle when the time comes.

Active Check Update Interval (seconds)
The system is designed to be pro-active and to check for any internal issues. This setting determines how often the system will check for table fragmentation and database deadlocks.

Metric Update Interval (seconds)
The Metrics are collected on a timer using this setting. To minimize database access and to maximize performance, some metrics are collected on every iteration and some will skip one or more iteration based on the metric collection iteration count settings that have been assigned to each metric. For example, if this property is set to 30 (seconds) and the "Collect Environment CPU Metrics" is set to "Every2Iterations" then the system will collect metrics every 60 seconds. If the property is set to "Always" the system will collect at every iteration, "Never" will never collect, "Once" will only collect upon server initiation. Also, the user can minimize database writes by using the global settings in the Application Server in the OneStream Server Configuration Utility.

Server Heartbeat Update Interval (seconds)
Used to specify how often each server updates its record that it is alive and responding to user input.

Collect Environment CPU Metrics
How often to collect environment CPU metrics.

Collect Environment Task Metrics
How often to collect environment task metrics (i.e; running tasks, Queued Tasks …).

Collect Environment Login Metrics
How often to collect environment user login metrics.

Collect Server Set CPU Metrics
How often to collect Server Set CPU metrics.

Collect Server Set Task Metrics
How often to collect Server Set task metrics (i.e; running tasks, Queued Tasks …).

Collect Server Disk Metrics
How often to collect server disk metrics (i.e; Average Disk read/write per sec…).

Collect Server Memory Metrics
How often to collect server memory metrics (i.e; Available mbytes…).

Collect Server Network Card Metrics
How often to collect server network card metrics.

Collect SQL CPU Metrics
How often to collect SQL Server CPU metrics.

Collect SQL Page Metrics
How often to collect SQL Server Page caching metrics (i.e; Page Life Expectancy…).

Collect SQL Memory Metrics
How often to collect SQL Server CPU metrics.

Collect SQL Connection Metrics
How often to collect SQL Server connection metrics (i.e; Number of connections…).

Collect SQL Query Metrics
How often to collect SQL Server Query metrics (i.e; Number of Deletes/Inserts…).

Collect SQL File Metrics
How often to collect SQL Server File growth metrics.

Collect SQL Elastic Pool CPU Metrics - Azure SQL
How often to collect SQL Server Elastic Pool CPU metrics (i.e; Number of connections…).

Collect SQL Elastic Pool DTU Metrics - Azure SQL
How often to collect SQL Server Elastic Pool DTU metrics (i.e; Number of connections…).

Collect SQL Elastic Pool Storage Metrics – Azure SQL
How often to collect SQL Server Elastic Pool Storage metrics (i.e; disk storage usage…).

Collect SQL Elastic Pool Workload Metrics - Azure SQL
How often to collect SQL Server Elastic Pool Workload metrics.

SQL Blocking Timeout Interval (Minutes)
Checks the SQL Blocked Items Timestamp. If the Timestamp is greater than "SQL Blocking Timeout Interval (minutes), " a warning is logged.

Fragmentation Iteration Count
Default is 600 (minutes). Used for fragmentation check, every 10 hours if this field is set to 600. This is used to determine how often the database tables are fragmented.

Fragmentation Percent Threshold
Default is 90. Used for fragmentation threshold check in percent.

Detailed Logging
If true, then log whenever we enter and exit the metric collection and the Active System check.

Number Hours to Retain Offline Servers
Default is 1. Remove offline servers from the heartbeat table after certain number of hours.

Task Load Balancing

Task Load Balancing	
□ General	
Maximum Queue Processing Interval (seconds)	10
Maximum Average CPU Utilization	70
Maximum Queued Time (minutes)	30
Number of Past Metric Readings for Average CPU	2
Task Logging	False
Detailed Logging	False

Default settings for Task Load Balancing for larger jobs like consolidation and data management. Application servers utilize queuing and smart load balancing to run that task on the appropriate application server. Task queueing and smart load balancing will prevent more than one processor-intensive task from running on the same server at the same time. When an asynchronous task is started (i.e., a task that uses the progress bar), it can be initialized in a Queued state before it starts its work in its Running state. The queued state takes very little application server resources. The algorithm keeps the task in the Queued state until all other queueable tasks have completed on that application server or until the CPU is low enough to run the task. There are also settings that can be configured to cause queued tasks to be automatically run if too much time elapses.

Maximum Queue Processing Interval (seconds)
Default is 10. Used to specify how often the queue will look for new jobs to execute.

Maximum Average CPU Utilization
Default is 70. Used to specify the maximum CPU utilization before a task is queued to a server but not executed till the CPU drops below that maximum.

Maximum Queue Time (minutes)
Default is 30. Used to specify the max queued time before a job is executed.

Number of Past Metric Readings for Average CPU
Default is 2. Used to specify the number of past metric reading used to calculate the average CPU utilization.

File Share Root Folder
To set the file share root folder property, type in the path to the file share folder, or copy and paste the path from Windows Explorer.

File Share Root Folder	C:\OneStream\FileShare

> **NOTE:** File Share Root Folder might have a default directory defined as C:\OneStream\File Share\. This directory is not created. Please define the File Share Root Folder directory. Grant the system user NT AUTHORITY\NETWORK SERVICE full access to this folder.

File Share Batch Harvest Root Folder
Used to define a separate folder path for the Batch Harvest folder. Azure users who do not have access to the Azure folder can use this field to identify a folder where they can place their files for batch harvest.

> **NOTE:** If you specify a separate path, the default folder will not be used.

Business Rules Assembly Folder
Used by Application Servers to reference the location of DLL files stored in a common Network Share Folder.

User Inactivity Timeout in Minutes
Used to identify the number of minutes a user has before their OneStream session times out while they are on a page in XF, or in Excel, without taking action.

Task Inactivity Timeout in Minutes
Used to identify the number of minutes a task in user cancelled status, or sitting in the queue waiting to be processed, has before being timed out.

Specify Log Retention Information
To set the log retention properties in days, type the number of days that the organization wishes to ensure logs are retained. If the delete activity or error logs buttons are used in OneStream, the logs will be cleared with the exception of the current number of days specified.

Number of Days to Retain Logon Activity	90
Number of Days to Retain Task Activity	90
Number of Days to Retain Error Log	90

Log Retry Errors

If an application server has a transient issue trying to access the database, it will retry the database query. If LogRetryErrors is set to True (the default), an item will be added to the error log to indicate that a retry was needed. If an implementation is seeing a lot of database retries in the error log, they should review their database installation to make sure it is operating correctly.

PDF Embedded Fonts to Remove

Embedding fonts in a PDF Report Book significantly increases the size of the PDF file. Use this property to specify the fonts to not embed to reduce the size of PDF files and control the resolution during Report Book PDF generation. For multiple fonts, use a semicolon separated list.
The default setting is: Arial; Calibri; Segoe UI; Tahoma; Times New Roman; Verdana.

> **NOTE:** This is for Report Books only.

PDF Embedded Fonts to Remove	Arial; Calibri; Segoe UI; Tahoma; Times New Roman; Verdana

Authentication

External (Optional)

You can configure LDAP authentication if required. To configure, click to the right of the (Collection) and a new ellipsis will appear.

> **NOTE:** When upgrading OneStream from a version prior to version 5.1.x, please note that the Windows Section of the external authentication provider has been enhanced to support SSL enabled MSAD. As part of this change any prior MSAD authentication providers will need to be transferred to the updated Windows Section of the external authentication provider entry. The Domain Name should be verified and entered into the "Name of Account Store" field and the appropriate binding options and type of account store should be specified.

External Authentication Providers	(Collection)

Click on the new ellipsis for the editor window.

Create a name for the provider and fill in the information above based on the installation. The final configuration is then performed in the security section of under the individual ID's profile.

> **NOTE:** Comma separated values (group names) can be entered in the Security Group Names field. A user must belong to any of the groups listed before OneStream will authenticate the user with MSAD. Security groups are also still needed in OneStream.

> **NOTE:** Windows Authentication mechanism has been enhanced to support SSL enabled MSAD. See below a typical example of configuration values for Combined Multiple Server Binding Options, Server Binding Options, Security Group Names, Account Store Container Name fields.

Name of Account Store
The name of the domain or server for Domain context types, the machine name for Machine context types, or the name of the server and port hosting the ApplicationDirectory instance.

Name of Account Store Container
The container on the store to use as the root of the context. All queries are performed under this root.

Type of Account Store
Specifies the type of store to which the principal (user) belongs. Possible values are ApplicationDirectory (represents the AD LDS store), Domain (represents AD DS store) and Machine (represents SAM store). Defaults value is Domain.

Server Binding Options
A combination of one or more ContextOptions values specifying the options used to bind to the server. Use one of the options from the dropdown or, if UseCombinedContextOptions is specified, enter multiple comma separated values in the Combined Multiple Server Binding Options field. See Appendix 12 for a detailed description of these options.

> **NOTE:** OneStream's implementation of MSAD/LDAP authentication allows for the use of a single username and password when prompted to log into OneStream.

Under Systems | Tools | Security, then the individual profile.

Authentication	
External Authentication Provider	Test

Native Authentication Settings

Native Authentication	
General	
OK to Use Same Password after Reset	True
Password Expiration (days)	0
Password Minimum Length	1
Password Maximum Length	100
Password Requires Letters	False
Password Requires Numbers	False
Password Requires Mixed Case	False
Maximum Invalid Logon Attempts	-1

Control the standards for creating and updating passwords by defining the minimum/maximum character length, variety of characters, expiration dates and maximum invalid logon attempts. The maximum Invalid Logon Attempts default is -1 (no maximum). If a positive number is provided in this field, and a user exceeds the provided maximum invalid logon attempts, their account will be locked/disabled. After an Administrator re-enables their account, the user will be required to change their password.

> **NOTE:** The built in Administrator's account can also be disabled if the maximum invalid login attempts value is exceeded, like other users working in the user interface.

Azure Configurations (Azure-only or if Elastic Pool Being Used)

Azure Subscription Settings

The Azure subscription settings must be filled in, as they are used for login and retrieve Azure settings and data. In version 5.0 this is used to retrieve Elastic Pool metrics.

⌀ Subscription	
⊟ General	
Azure Subscription ID	
Azure Tenant ID	
Azure Client ID	
Azure Secret Key	
Azure App Insights Instrumentation Key	

Cache

⊟ Cache	
Derive Maximum Number of Data Records in RAM	True
Reserved Memory	25%
Number of Bytes Per Data Record	3200
Maximum Number of Data Records in RAM	10000000
Maximum Number of Expanded Cube View Rows	1000000
Maximum Number of Unsuppressed Rows Per Cube View Page.	2000
Maximum Number of Seconds To Process a Cube View.	20

Reserved Memory (GB)
Enter either a number of gigabytes (e.g., 8) or a percentage of total RAM (e.g., 40%). This property is a string that contains a number for GB or a percentage of total RAM. If a percentage is specified, and after it is converted, the minimum is 4GB and the maximum is 64GB. If a number is specified, the minimum is 0GB and the maximum is 256GB.

⊟ Databases	
Database Server Connections	(Collection)

Databases
Specify Database Information For the application server to connect to a database server, define one or more database connections to which the server can connect. Database connections can be imported from an xml file containing encrypted connection strings created using OneStream Database Configuration Tool or connection strings can be created directly in the OneStream Server Configuration Tool.

Importing Encrypted Database Connections
To import connections produced by the OneStream Database Configuration Tool, open the OneStream Server Configuration Tool and click the Tools menu and select the Import Database Server Connections option. Now browse for the encrypted database connections xml file produced using the OneStream Database Configuration Tool. After the connections have been imported, be sure to name each connection so that it can be referenced in the configuration process.

See Appendix 6: Upgrade Process Overview for more details.

Database Server Connection Settings
Changes need to be made to the Database Server Connections for users to create and change data in additional database tables. These settings are used by some of the OneStream MarketPlace Solutions such as Specialty Planning and Reconciliation Control Manager.

Select the Database Server Connections field and click the ellipsis to launch these settings.

Azure Database Connection Settings

Azure Database Connection Settings	
Azure Elastic Pool Max DTU Setting	1600
Azure Elastic Pool Min DTU Setting	200
Azure Elastic Pool Name	
Azure Resource Group	
Azure Service Level Objective	
Azure SQL Edition	
Azure SQL Scaling Type	NotUsed
Azure SQL Server Name	
Azure SQL Storage Max Size	0
Azure SQL System Business Rule Name	

Azure Elastic Pool Max DTU Setting
This is a fail-safe setting that the user can't set the DTU setting above this point.

Azure Elastic Pool Min DTU Setting
This is a fail-safe setting that the user can't set the DTU setting below this point.

Elastic Pool Name
The name of the elastic pool used with this database connection.

Azure Resource Group
The resource group name that the elastic pool is in.

Azure Service Level Objective
The service level used. This setting is used to create application on Azure.

Azure SQL Edition
The Azure SQL Server edition used.

Azure SQL Scaling Type
This feature will be available in a future release.

Manual, Business Rule, and ManualAndBusinessRule. The type of scaling that is used to scale in/out the SQL Server eDTUs.

Azure SQL Server Name
The name of the SQL Server database. This setting is used to create application on Azure.

Azure SQL Storage Max Size
This is used to specify the database storage size when creating a database on Azure.

Azure SQL System Business Rule Name
This feature will be available in a future release.

If SQL Scaling Type is set to Business Rule, this setting must be set to a Business Rule that is used to Scale Out and Scale In. The Environment metrics and the database metrics are passed to this rule to properly determine the eDTU scaling. See System Extender Business Rules in the Design and Reference Guide.

General	
Name	OneStream Database Server
Access Group for Ancillary Tables	Administrators
Allow Database Creation via UI	True
Can Create Ancillary Tables	True
Can Edit Ancillary Table Data	True
Database Provider Type	SqlServer
Is External Database	False
Maintenance Group for Ancillary Tables	Administrators
Table Creation Group for Ancillary Tables	Administrators
Use File Groups when Creating Databases	True
Use Table Partitioning when Creating Databases	True

Access Group for Ancillary Tables
This should be set to a group who will edit records.

Maintenance Group for Ancillary Tables
This should be set to a group who will create the tables.

Other settings highlighted need to be set to True in order to execute table creation via the MarketPlace Dashboards.

Defining Database Connections Manually
Database server connections are defined and named using the Database Server Connection Collection Editor which is opened by clicking the button in the right column of the Database Server Connections property.
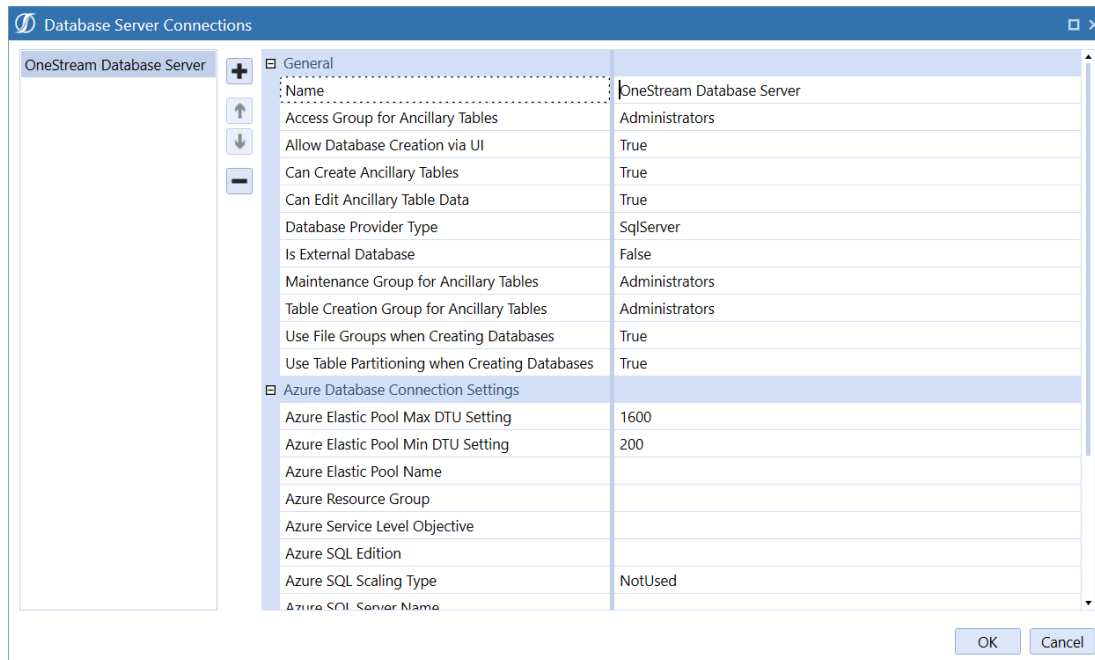
The Database Server Connection Collection Editor allows database connections to be added or removed and the creation of meaningful names for the connections. The name of the connection is important because it acts as an abstraction layer to which the application server can interact. This allows the database administrator to change the connection string for a name database server connection without affecting the application server or any other component that may rely on the named connection.

In addition, name database connections can help organize development, test, and production environments because relevant names can be assigned to a database connection based on its environment. Named connections appear as a list in the OneStream user interface during the process of creating new applications. This feature allows OneStream administrators to simply pick database connections to use for the new application database. If the Allow Database Creation Via UI property in the Application Server Configuration is set to False, this means none of the attached database servers are setup to allow new application creation which will then disable the Create New Application Database icon.

> **NOTE:** The Allow Database Creation Via UI property in the Application Server Configuration is always set to False for Cloud customers.

Specify Database Connections and Schemas

Once database connections have been imported or defined manually, they can be used in conjunction with the schema name to define full connection information for the Framework and State databases.

Define Framework Database Connection

In order to define the Framework database connection, specify the database server connection name that contains the Framework database, and then specify the name of the Framework database schema on the server.

Framework Database Server Connection	OneStream Database Server
Framework Database Schema Name	OneStream_Framework

Define State Database Connection

In order to define the State database connection, specify the database server connection name that contains the State database, and then specify the name of the State database schema on the server.

State Database Server Connection	OneStream Database Server
State Database Schema Name	OneStream_State

Server Sets

> **NOTE:** A server should only be in one server set.

A server set should contain all servers, which will perform that set's unique combination of behaviours.

Used to create Server Sets for server grouping.

Server Sets	
Server Sets	(Collection)

Azure - This feature will be available in a future release.

These settings apply only when running in OneStream Cloud.

Azure Resource Group Name
This feature will be available in a future release.

The Azure resource group name for the Server Set.

Azure Scale Set Name
This feature will be available in a future release.

The Azure scale set name in the resource group.

Can Stop or Start Servers
This feature will be available in a future release.

If true, then the user can stop and start the server from the Environment page.

Maximum Capacity
This feature will be available in a future release.

A failsafe setting specifying the maximum number of servers that can be scaled up to.

Minimum Capacity
This feature will be available in a future release.

A failsafe setting specifying the minimum number of servers that can be scaled down to.

Scaling Type
This feature will be available in a future release.

Specify whether this Scale Set is scaling at all or doing so manually, using a Business Rule, Automatically or both manually and a Business Rule. If Business Rule-based, see next property.

System Business Rule Name
This feature will be available in a future release.

If Scale Set is scaling using a Business Rule, then the Business Rule name needs to be specified. See System Extender Business Rules in the Design and Reference Guide.

Behaviors

Process Queued Consolidation Tasks
If set to true, then this server can process Consolidation tasks.

Process Queued Data Management Tasks
If set to true, then this server can process Data Management tasks.

Process Queued Stage Tasks
If set to true, then this server can process Stage tasks.

Queued Tasks Require Named Application Server
If set to true, then this server will only run tasks that are assigned to it.

General

Name
Server Set name.

Server Name for Standard Server Sets (Supports *? Wildcards)
Specify the Server names if we are using the Standard Server Set Provider type. See next property.

Sever Set Provider
Specify whether we are using the "Standard", "Azure", or "External" provider type.

Processing

Can Change Queueing Options on Servers
Specify whether the queueing options of a specific server can be changed. If the Value is set to True, it will allow the administrator to change the queueing behavior of a specific server as it relates to queueing Stage, Consolidation and Data Management tasks.

Can Pause or Resume Servers
If set to true, then the user can pause and/or resume the server from the Environment page.

Can Recycle App Pool on Severs
If set to true, then the user can recycle the app pool from the Environment page via the Reset IIS button.

Task Activity

Task Activity is used to capture log information for Books, Cube Views and Quick Views to analyze data analysis performance.

⊟ Task Activity	
Log Books	True
Log Cube Views	True
Log Quick Views	False
Threshold for Logging Get Data Cells (count)	2000

Log Books
When set to True (default), a log is created in Task Manager when the items are included as Task Activity steps for that specific book. The intention of this feature is to verify entries in the Task Activity grid and the settings in the configuration file work as expected.

Log Cube Views
When set to True, a log is created in Task Manager when a Cube View is opened, a report is run or an export to Excel is completed in the data explorer. The intention of this feature is to analyze data analysis performance.

Log Quick Views
When set to True, a log is created in task manager when a new Quick View is created or when rows/columns are shifted/moved around. The intention of this feature is to analyze data analysis performance.

Threshold for Logging Get Data Cells (count)
This logs the calls to GetDataCells and GetDataCellsUsingScript. It includes context information such as the Excel file name or the Cube View name. It only creates logs if the number of Data Cells being requested is equal to or greater than the value provided in this field.

Sharing One Configuration File for All Application Servers

All OneStream application servers can share a single configuration file if desired. This makes controlling server behavior more centralized and reduces configuration time.

Sharing a configuration file is a simple process. Follow the standard application server configuration process on one server and then copy the configuration file to a file share that all application servers can read. This shared folder will then need to be referenced by each application server in its ASP.Net Web.Config file.

Setting a Reference for the Configuration File Share Folder

Open the OneStream Server Configuration Tool on each application server, click the File menu, and select Open ASP.Net Configuration File. Next, open to the Web.Config for the application server. This file is located in application server's virtual directory root folder (C:\Program Files\OneStream Software\OneStreamAppRoot\OneStreamApp\Web.Config). Once the file is open, set the Configuration Folder property value equal to the configuration file share folder, save the configuration file and restart IIS.

Configuring Web Server(s)

Configuring OneStream web servers requires one or more application server(s) to be defined for the web to connect to it. The web server configuration file not only defines the URL connection information required for a web server to communicate with an application server, but it also defines the processing capabilities of the application server.

Using the Server Configuration Tool

1. To begin the application server configuration process, open the OneStream Server Configuration Tool.

2. Go to **File > Open Web Server Configuration File**.

3. Browse for the default XFWebServerConfig.xml located in web server's virtual directory (C:\Program Files\OneStream Software\OneStreamWebRoot\OneStreamWeb\App_ Data\XFWebServerConfig.xml).

4. Once it is open, begin the configuration process.

Enable SSO Logging

When you enable SSO logging, a log file is created that records log on activity for attempts, successes and failures. If deactivated (set to False), the log file must be manually deleted from the server.

Each log file is limited to approximately one megabyte. When a file reaches it's maximum size, it is rolled (renamed) and a new file generated. You can have ten rolled logs. When the eleventh is generated, the oldest logs is deleted.

To enable SSO logging:

1. In the Web Server Configuration Settings dialog, click the elipsis in **Single Sign On Identity Provider**.

Web Server Configuration File - C:\Program Files\OneStream Software\OneStreamWebRoot\OneStreamWeb\App_Data\XFWebServerConfig.xml	
⊟ Web Server Configuration Settings	
Application Servers	(Collection)
Single Sign On Identity Provider	(Detail)
Enable Silverlight	True
Use Native App Detailed Logging	False
Use Detailed Error Logging	True
Server Heartbeat Update Interval (seconds)	10
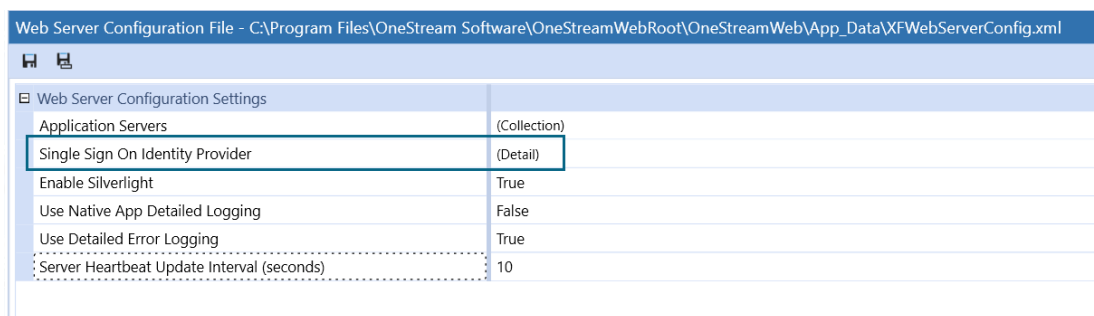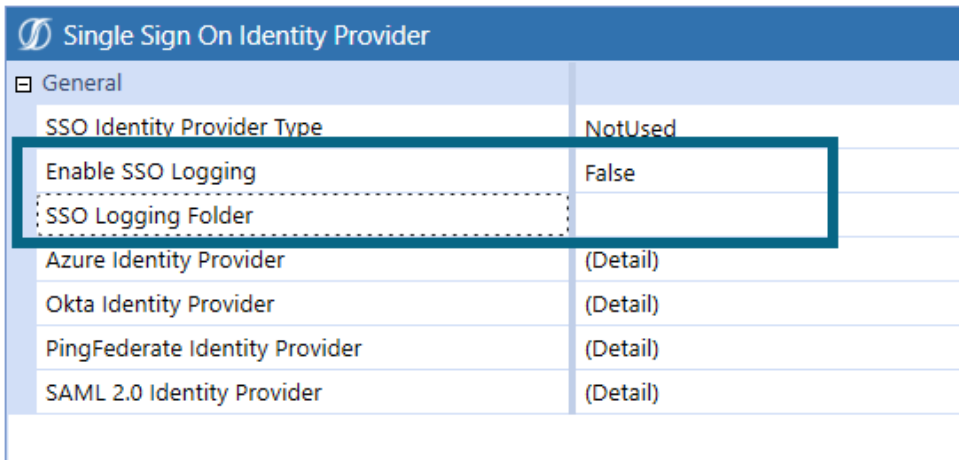
The Single Sign On Identity Provider dialog box opens.

2. In **Enable SSO Logging**, select **True**.

3. In **SSO Logging Folder**, enter the directory path to store the file. We recommend a sub-folder in the config files directory.

> **IMPORTANT:** The OneStream Web process owner must have create/write permission to the folder where the logs are written.

Creating Application Server Definitions

Application server connections are defined and named using the Web Server App Server Collection Editor which is opened by clicking the button in the right column of the Application Servers property.

Application Server Properties

Pause or Resume setting will determine if the Pause and Resume button will be displayed in the Web To App Connections. If set to True the buttons will be displayed and will allow the user to pause and then resume a specific connection to an application server. If this option is set to False the buttons will not be displayed.

The Web Server App Server Collection Editor allows application server definitions to be added or removed and the creation of meaningful names for each server, the processing capabilities of the server, and the WCF Address (URL of the application server).

Application Server Processing Capabilities

Application servers can be designated to perform specific processing tasks or all processing tasks.

Consolidation	Consolidation application servers are limited to performing consolidation related functions which can be very hardware-intensive.
Data Management	Data Management application servers are limited to performing data related functions which can be very hardware-intensive.
General Access	General Access application servers can perform all processing tasks, including consolidations and stage loads.
Stage Load	Stage Load application servers are limited to performing Staging (Load & Transform) related functions which can be very hardware-intensive.

Application Server WCF Address

OneStream application servers uses the Windows Communication Foundation (WCF) for inter server communications. In order for a web server to locate and communicate with an application server, specify the application server's URL.

OneStream uses standard ports for its web and application servers. Sample URL's with the standard ports numbers are listed in the table below.

Web Server Sample URL:	http://<Servername>:50001/OneStreamWeb/OneStreamXF.aspx
Application Sever Sample URL:	http://<Servername>:50002/OneStreamApp

Sharing One Configuration File for All Web Servers

All OneStream web servers can share a single configuration file if desired. This makes controlling server behavior more centralized and reduces configuration time.

Sharing a configuration file is a simple process. Follow the standard web server configuration process on one server and then copy the configuration file to a file share that all web servers can read. This shared folder will then need to be referenced by each web server in its ASP.Net Web.Config file.

Setting a Reference for the Configuration File Share Folder

Open the OneStream Server Configuration Tool on each application server, click the File menu, and select Open ASP.Net Configuration File. Next, open to the Web.Config for the web server. This file is located in web server's virtual directory root folder (C:\Program Files\OneStream Software\OneStreamWebRoot\OneStreamWeb\Web.Config). Once the file is open, set the Configuration Folder property value equal to the configuration file share folder, save the configuration file and restart IIS.

Application Validation

After the installation and configuration is complete, test the application. Once the login page loads, make sure it paints correctly. For the very first login use the following username and password:

User Name	Administrator
Password	OneStream

> **NOTE:** This will require a password reset after the first login.

Configuring Secure Sockets Layer (SSL)

The following steps outline how to configure a OneStream Web Server for Secure Sockets Layer (SSL) encryption, so https is used instead of http. These steps were performed in IIS running on a Windows Server 2012 operating system.

Pre-Configuration

Configure the application servers and web servers for normal unencrypted http access.

Create a Server Certificate

1. Start IIS Manager and select the desired server in the left-hand Connections tree. Then double-click the **Server Certificates** icon.

2. Select the appropriate option in the right-hand pane to create a test certificate, or to request an official certificate from an authority.

3. To create a test certificate, select **Create Self-Signed Certificate**. Then specify a name and select the **Personal** certificate store.

4. Click **OK**.

Create Web Server HTTPS Binding

> **NOTE:** Configure web server(s), not application server(s) for SSL.

Configuring the web servers will cause the internet traffic between end-users and the web server to be encrypted. Since the web servers and application servers are physically co-located in the same server room, it is not usually necessary to pay the performance penalty for encrypting that link.

1. In IIS, select the **OneStream Web Server Site**, right-click, and select **Edit Bindings**. By default, the web server will initially have only one binding (for http), add a binding for https.

2. In the Site Bindings dialog, select **Add**. Then select **https**, **All Unassigned**, and use the default SSL port (443) if appropriate.

3. Leave Hostname empty and leave Require server name indication unchecked.

4. In the SSL certificate combo box, select the certificate that was created above.

5. Click **OK**.

Configuring SSL On the Application Server Tier

Configuring SSL on the Application Server Tier in the environment will require the use of a local account for the IIS Application Pools (OneStreamAppAppPool) rather than a domain level service account. This can be achieved using either:

1. A Microsoft Azure Storage Account in a Azure deployment of OneStream.

   Configuring the application servers will cause the internet traffic between the web server and application server to be encrypted.

2. In IIS, select the **Application Pools** node and right-click on **OneStreamAppAppPool**. Click **Advanced Settings**.

3. Click on the Identity field and click the … icon to update the identify for the application pool.

4. Choose the **Custom Account** radio button and click **Set**.

5. Enter the local account username and corresponding password and confirm the password and click **OK** to save and **OK** to confirm.

6. Click **OK** to close the application pool Advanced Settings Dialog.

7. Recycle IIS for the changes to take effect.

Perform this process on each OneStream Application Server in the environment.

Test SSL Address

1. Restart IIS.

2. Open a browser and navigate to the OneStream site using https instead of http.

   For example, the URL for the encrypted (SSL) connection could be:

   ```
   https://[SeverName]:[SSLPortNumber]/
   OneStreamWeb/OneStreamXF.aspx
   ```

3. If a test certificate was used instead of an officially signed certificate, the browser displays a warning indicating a problem with the security certificate. **Click Continue to this website**.

4. Log onto OneStream to ensure that the client can communicate with the web server.

Disable Unencrypted HTTP Access

Even if an SSL connection is fully configured, users can access the web server using the URL for the unencrypted HTTP connection. Disable that access as follows to ensure users only use SSL (HTTPS).

1. In IIS, select the **OneStream Web Server Site** and then double-click the **SSL Settings**.

2. Select **Require SSL** and click **OK**.

3. Restart IIS.

4. Test both URLs in a browser to confirm that:

    - The HTTPS URL works.

    - The HTTP URL displays an error.

External Security Providers and Single Sign-On (SSO)

This section provides an overview for configuring OneStream Application access with a single sign-on provider.

When users sign in to a OneStream application, they go through an authentication process where they are required to prove that they are who they say they are. This is typically done by entering a user ID and password in the OneStream application. The user ID and password could be through external authentication that are Cloud-based where OneStream is not storing a user's password or via OneStream native users, where this password is stored in the OneStream Framework.

How Does Single Sign-on Work?

- The OneStream administrator must provision a user to OneStream in the System tab. The properties of this user denote whether this is a native (internal) OneStream user or to which external authentication provider they belong, in which case no password is required to be stored. The provisioning of this user record occurs manually or via the OneStream BRAPI (i.e. Client API). OneStream Security identifies the user's roles and data authorization inside each Application.

- Federated Single Sign-On enables applications to redirect to Azure AD, Okta, PingFederate or your SAML 2.0 provider for user authentication.

- The user experience during the authentication process is dependent on the configuration of your identity provider. The default experience is the user is presented with a dialog where they would enter the single user ID and password previously configured in an external provider that they may use to sign in to other corporate applications, hence a single sign-on.

- Most external authentication providers allow for control over the user experience using techniques such as Integrated Windows Authentication (IWA) or a variant which can eliminate or reduce the need to enter a user name and password multiple times throughout the day. Consult with your identity provider for information on how to configure this experience.

  - For Azure AD this is referred to as "Pass-Through Authentication."

  - For Okta, this is referred to as "Okta IWA Web App."

  - For PingFederate, this is referred to as "PingFederate IWA Integration Kit."

> **NOTE:** After a user has been authenticated through SSO and logs off or the user does not match an existing user in OneStream Security, authentication can occur using native credentials.

OneStream supports Azure AD, Okta and PingFederate identity providers using OpenID Connect (OIDC) as well the SAML 2.0 authentication protocol. The steps in this subsequent section provide information on configuring OneStream for external authentication with Transport Layer Security (TLS) encryption (i.e., access to the Web Server using https versus http). For a point of reference, these steps were performed in IIS running on a Windows Server 2016 operating system.

Microsoft Azure AD Configuration

Azure AD Configuration: Adding an Application

Any application that is to be configured to use the capabilities of Azure AD must first be registered in an Azure AD tenant. This registration process involves giving Azure AD details about the application, such as the URL where it is located, the URL to send replies after a user is authenticated, the URI that identifies the app, etc.

First, create the three application registrations as shown below. During the process, you'll be asked to take note of Application IDs, URIs, SSO keys and other necessary information required to configure OneStream.

Web Application Setup

This section will describe the setup of the Azure AD application used by OneStream to authenticate clients accessing the Microsoft Silverlight application using a PC and Internet Explorer.

Complete the following steps.

1. Create a new Application Registration with the following properties:

    - **Name**: This should identify the type of application being created for example "OneStream Web"

    - **Supported Account Types**: "Accounts in this organizational directory only"

    - **Redirect URI**: Web

2. Enter the sign-on URL of the application in the text field to the right of the dropdown under Redirect URI. This is the URL that Azure will point to. For example, http(s)://<servername>:<port>/onestreamweb/onestreamxf.aspx

Home > OneStream Cloud Dev/Test AD - App registrations > Register an application

Register an application

* Name

The user-facing display name for this application (this can be changed later).

OneStream Web ✓

Supported account types

Who can use this application or access this API?

⦿ Accounts in this organizational directory only (OneStream Cloud Dev/Test AD)

◯ Accounts in any organizational directory

◯ Accounts in any organizational directory and personal Microsoft accounts (e.g. Skype, Xbox, Outlook.com)

Help me choose...

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web ⌄ | https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx ✓

3. Once registered take note of the Application ID value in the Overview pane and copy this to a text editor for later use.

4. Complete the following sections and information in the Overview Page:

    •   **Authentication**

    •   **Redirect URI**: Enter the URL for the Windows app of your OneStream environment in the Redirect URI field in addition to the existing entry. For example, http (s)://<servername>:<port>/onestreamweb/onestreamwindowsapp.aspx

        > **NOTE:** This value must be entered exactly and saved

    •   **Implicit Grant**

    •   **ID Tokens**: Check the selection box

    •   **Expose an API**

    •   **Application ID URI**: Click the "set" button this and change this App ID URI value to your site URL of http(s)://<servername>:<port>/onestreamweb/onestreamxf.aspx and click **Save**.

        > **NOTE:** This value must be entered exactly.

5. Click **Add a scope** and fill in the following parameters:

    •   **Scope name**: user_impersonation

    •   **Who can consent?**: Admins and users

    •   **Admin consent display name**: Access OneStream Web

    •   **Admin consent description**: Allow the application to access OneStream Web on behalf of the signed-in user.

    •   **User consent display name**: Access OneStream Web

- **User consent description**: Allow the application to access OneStream Web on your behalf.

- **State**: Enabled

Branding

Home Page URL: Enter the site URL of http(s)://<servername>:<port>/onestreamweb/onestreamxf.aspx.

> **NOTE:** This value must be entered exactly.

Certificates and Secrets

1. Click Certificates & Secrets in the navigation bar to generate a key.

    a. Click **New client secret** and give it a Description such as OneStream Web.

    b. Select the appropriate expiration time limit in the Expires field.

    > **NOTE:** If a key rotation policy has been implemented, select the appropriate time length.

    c. Click **Add**. The key is generated.

    > **NOTE:** The key can never be accessed again, so it is very important to take note of the key value for use and copy to the last page.

    > **NOTE:** If the key needed to be checked at a later time, a new one will need to be generated.

    d. Click **Add**.

2. Creating the Web registration is now complete.

Mobile Application Setup

This section will describe the setup of the Microsoft Azure AD application used by OneStream to authenticate clients accessing the application using the HTML 5 / Mobile interface.

1. Create a new Application Registration with the following properties:

   - **Name**: This should identify the type of application being created for example "OneStream Mobile"
     Supported Account Types: "Accounts in this organizational directory only"
     Redirect URI: Web

   - Enter the mobile application url in the text field to the right of the dropdown under Redirect URI. For example, http(s)://<servername>:<port>/onestreammvc.aspx

   Register an application

   ---

   * Name

   The user-facing display name for this application (this can be changed later).

   | OneStream Mobile | ✓ |

   Supported account types

   Who can use this application or access this API?

   ⦿ Accounts in this organizational directory only (OneStream Cloud Dev/Test AD)

   ◯ Accounts in any organizational directory

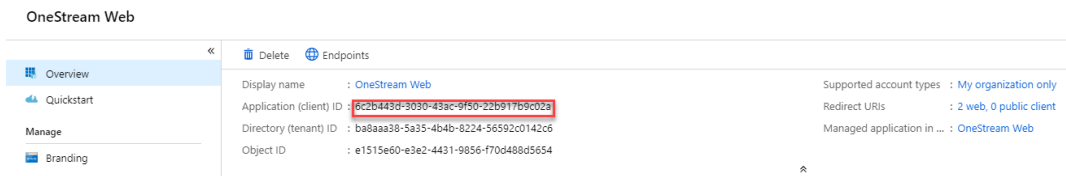   ◯ Accounts in any organizational directory and personal Microsoft accounts (e.g. Skype, Xbox, Outlook.com)

   Help me choose...

   Redirect URI (optional)

   We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

   | Web ⌄ | https://sitename.onestreamcloud.com:50004/onestreammvc | ✓ |

2. Once registered take note of the Application ID value in the OverView pane and copy this to a text editor for later use:

3. Click on **Authentication** and navigate to implicit grant in the same pane.

4. Click the ID tokens checkbox and click **Save**.

5. Expose an API:

   - Application ID URI: Click the "set" button and change this App ID URI value to your site URL of http(s)://<servername>:<port>/onestreammvc and click Save.
     Note: This value must be entered exactly.

6. Click **Add a scope** and fill in the following parameters:

   - **Scope name**: user_impersonation

   - **Who can consent?**: Admins and users

   - **Admin consent display name**: Access OneStream Web

   - **Admin consent description**: Allow the application to access OneStream Web on behalf of the signed-in user.

   - **User consent display name**: Access OneStream Web

   - **User consent description**: Allow the application to access OneStream Web on your behalf.

   - **State**: Enabled

7. Click **Add**.

Branding

Home Page URL: Enter the site URL of http(s)://<servername>:<port>/onestreammvc and click Save

> **NOTE:** This value must be entered exactly.

Certificates and Secrets

1. Click **Certificates & Secrets** in the navigation bar to generate a key.

2. Click **API permissions** in the left navigation bar.

3. Click **Add a permission**.

4. Click **APIs my organization uses** under Select an API.

5. Click in the Search bar and type in the name of the Web registration that was just created. For example, OneStream Web.

6. Once located, click the name of the Web registration.

7. Click the **Delegated permissions** option and the checkbox for user_impersonation.

8. Click **Add permissions**.

   The newly added entry now appears in the API permissions section.

   Creating the Mobile registration is now complete.

Native Application Setup

This section describes the setup of the Azure application used by OneStream to authenticate clients accessing the Application using the OneStream App for Windows, the Excel Add-In, and Studio.

1. Create a new Application Registration with the following properties.

   - **Name**: This should identify the type of application being created for example "OneStream Native"
     Supported Account Types: "Accounts in this organizational directory only"
     Redirect URI: Public Client (Mobile and Desktop)
     Enter the Native application url in the text field to the right of the dropdown under

Redirect URI. For example, http(s)://onestreamclient

Register an application

* Name

The user-facing display name for this application (this can be changed later).

OneStream Native

Supported account types

Who can use this application or access this API?

(•) Accounts in this organizational directory only (OneStream Cloud Dev/Test AD)

( ) Accounts in any organizational directory

( ) Accounts in any organizational directory and personal Microsoft accounts (e.g. Skype, Xbox, Outlook.com)

Help me choose...

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.
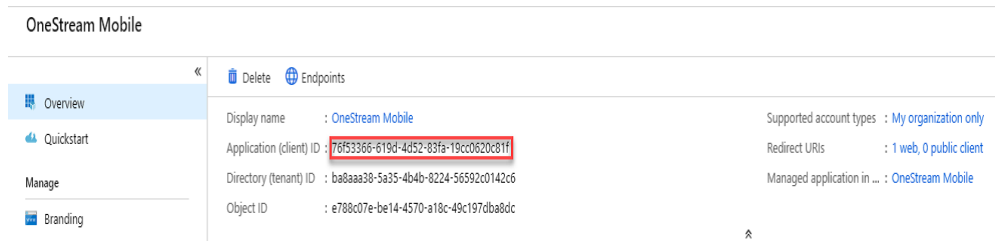
| Public client (mobile & desktop) | https://onestreamclient |

2. Once registered, take note of the Application ID value in the Overview pane and copy this to a text editor for later use.

Home > OneStream Cloud Dev/Test AD - App registrations > OneStream Native

OneStream Native

Overview
Quickstart

Manage

Branding

Delete   Endpoints

Display name	: OneStream Native	Supported account types	: My organization only
Application (client) ID	: e5c2e3e1-f814-4e6e-a1ec-eceac26f64ef	Redirect URIs	: 0 web, 1 public client
Directory (tenant) ID	: ba8aaa38-5a35-4b4b-8224-56592c0142c6	Managed application in ...	: OneStream Native
Object ID	: da729711-4039-43fa-b9c3-0524f2cda5e0		

3. Complete the following sections and information in the Overview Page.

- Click on **Authentication** and navigate to implicit grant the same pane.

- Click the **ID tokens** checkbox and click **Save**.

- Branding:

Home Page URL: Enter the site URL of http(s)://onestreamclient and click **Save**.

> **NOTE:** This value must be entered exactly.

4. Click API permissions in the left navigation bar.

5. Click **Add a permission**.

6. Click **APIs my organization uses** under Select an API.

7. Click in the Search bar and type in the name of the Web registration that was just created. For example, OneStream Native.

8. Once located, click the name of the Web registration.

9. Click **Delegated permissions option** and the checkbox for user_impersonation.

10. Click **Add permissions**.

11. The newly added entry now appears in the API permissions section.

    The Native registration is now complete.

OneStream Web Server Configuration

This section describes the configuration of the OneStream Web Server Configuration file to support Microsoft Azure SSO.

> **NOTE:** See the following example of the result of these steps. Several values, including Client ID and Secret Key will be different than the examples shown.

1. Launch the OneStream Server Configuration Utility application.

2. Select **File > Open Web Server Configuration File**.

3. In the WebServer Configuration File window, expand the Single Sign On Identity Provider section.

4. Select **Azure** in the SSO Identity Provider Type field.

5. Expand the Azure Identity Provider section.

6. Enter the domain of your Azure tenant in the Azure Domain field. Any domain name registered in this tenant can be used.

7. Enter the Home Page URL value previously noted during the Web Application setup into the Azure Web Reply URL field. For example, http(s)://<servername>:<port>/OneStreamWeb/OneStreamXF.aspx

8. Enter the Windows App Launch Page value previously noted during the Web Application setup into the Azure Web Reply URL field. For example, http(s)://<servername>:<port>/OneStreamWeb/OneStreamWindowsApp.aspx

9. Enter the Application ID value previously noted during the Web Application setup into the Azure Web Client ID field.

10. Enter the App ID URI value previously noted during the Web Application setup into the Azure Web App ID URI field.

11. Enter the Key value previously noted during the Web Application setup into the Azure Web SSO Secret Key field

12. Enter the Home Page URL value previously noted during the Mobile Application setup into the Azure MVC Mobile Reply URL field.

13. Enter the Application ID value previously noted during the Mobile Application setup into the Azure MVC Mobile Client ID field.

14. Enter the App ID URI value previously noted during the Mobile Application setup into the Azure MVC Mobile App ID URI field.

15. Enter the Key value previously noted during the Mobile Application setup into the Azure MVC Mobile SSO Secret Key field.

16. Enter the Application ID previously noted during the Native Application setup into the Azure Client ID field.

17. Enter the Redirect URIs previously noted during the Native Application setup into the Azure Client Redirect URI. This should always be http(s)://OneStreamClient .

18. Click **File > Save** to save the changes to the Web Server Configuration file.
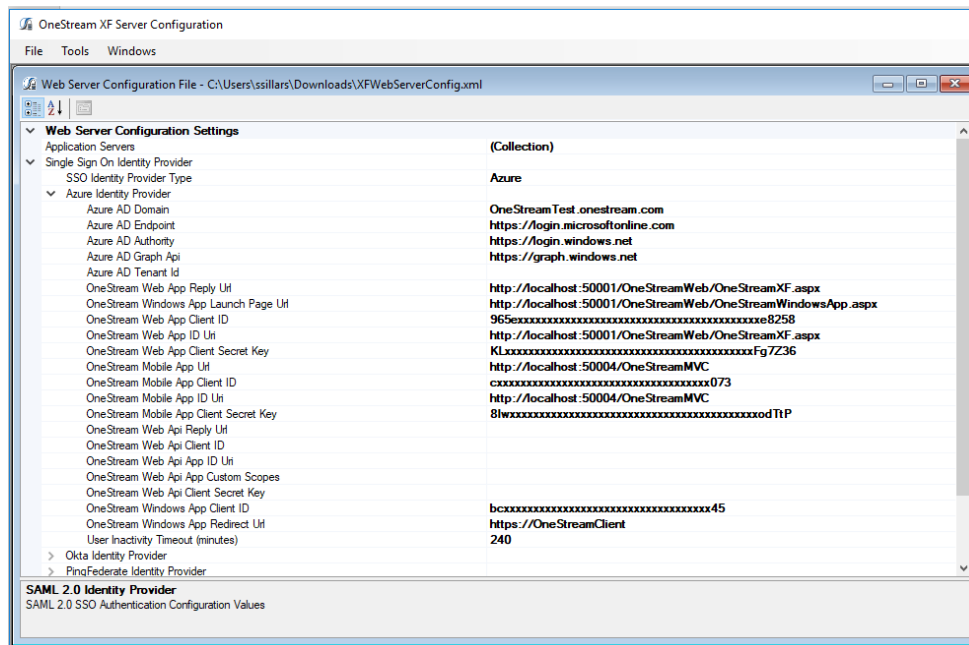
Azure AD Government Tenant

Azure AD Government uses different endpoints for authentication from Commercial tenants.

The following values will need to be updated:

Azure AD Endpoint = https://login.microsoftonline.us

Azure AD Authority = https://login.microsoftonline.us

OneStream Application Server Configuration

This section describes the configuration of the OneStream Application Server Configuration file to support Azure SSO.

1. Launch the OneStream Server Configuration Utility application.

2. Open the Web Server Configuration File and select the value in the Azure Web SSO Secret Key field.

3. Copy the Azure Web SSO Secret Key value and then close the window.

4. Select File | Open Application Server Configuration File.

5. Navigate to the location where the XFAppServerConfig.xml file is stored.

6. Select the file, then click **Open** .

7. In the Authentication section, click inside the External Authentication Providers field. An ellipsis appears.

8. Click the Ellipsis.

9. In the External Authentication Providers window click **Add**.

10. Paste the value copied in Step 3 into the External Provider Web SSO Secret Key field.

11. Click the selection arrow in the Authentication Provider Type field and select AzureSSO.

12. Click in the Name field and enter AzureSSO.

13. Click **OK** to save the changes.

14. Click **File > Save** to save the changes to the Application Server Configuration file.

15. Close the window.

16. Reset IIS on all OneStream Servers.
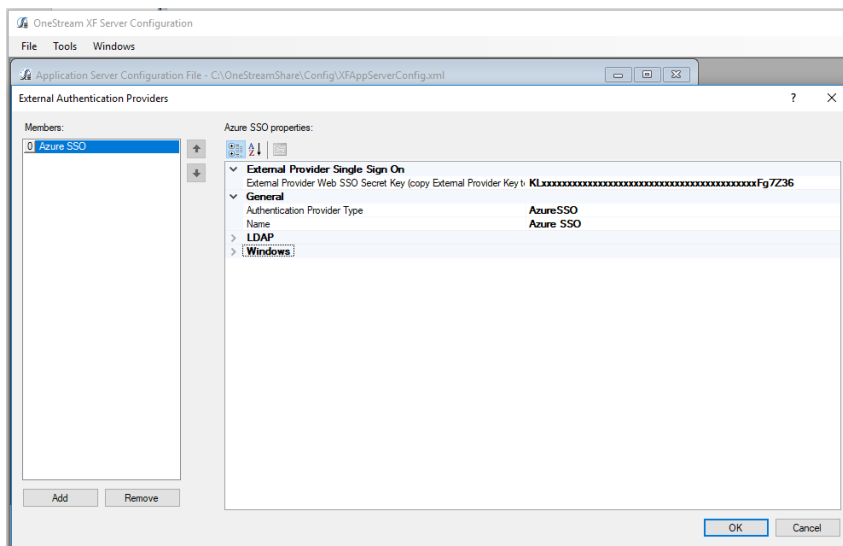
User Application Security Configuration

Add any users that will use the AzureSSO authentication method. Under the authentication section, select AzureSSO for the External Authentication Provider and the External Provider User Name should be the same as the user ID/email address listed in the Azure AD.

General	
Name	Azure User
Description	
Is Enabled	True
Authentication	
External Authentication Provider	AzureSSO
External Provider User Name	azureuser@onestreamsoftware.com
Internal Provider Password	
Preferences	

If the Azure SSO works properly, after logging into the Azure portal, it should redirect to the OneStream application URL and pass the Azure SSO token to OneStream. This will populate the user field on the login screen and allow for application selection without re-entering user credentials.

Okta Configuration

The section provides an overview for configuring OneStream application access and Okta.

OneStream supports application-initiated login only, a chicklet will not be displayed in Okta to your end users to access the application from. Once the OneStream configuration has been completed you can add a bookmark application to Okta pointing to your OneStream URL to provide access to OneStream from the Okta "My Applications" list.

The steps in this section provide information on configuring Okta for web, mobile and native applications (Excel, Studio and OneStream App for Windows) as well as authorizing the server for OneStream API access.

Create the three application registrations as shown below. During the process, you'll be asked to take note of Application IDs, URIs, SSO keys and other necessary information required to configure OneStream. Use the form on the last page of this guide to store the information. When completed, securely send the form to OneStream Cloud Services.

Mobile Application Setup

This section will describe the setup of the Okta application used by OneStream to authenticate clients accessing the application using the HTML 5 / Mobile interface.

> **NOTE:** The Client ID and Secret values will be different than the examples shown. Also, the servername placeholder should be updated to reflect your environment's server name.

1. Click the **Back to Applications** button to return to the Applications screen.

2. Click the **Add Application** button.

3. Click the **Create New App** button.

4. In the Create a New Application Integration window, verify that Web is selected in the Platform field and select the OpenId Connect option in the Sign on method field.

5. Click the **Create** button.

6. In the Create OpenID Connect Integration window, enter "OneStream Mobile" in the Application name field.

7. Enter the OneStream mobile URL in the Login redirect URIs field. For example, http (s)://<servername>:<port>/onestreammvc/authentication/logon

8. Click the Add URI button and enter the OneStream mobile URL in the Logout redirect URIs field. For example, http(s)://<servername>:<port>/onestreammvc/authentication/logon

9. Copy this URL to the last page.

> **NOTE:** This value must be entered exactly.

10. Click the **Save** button. The General Settings widow appears.

11. Click the **Edit** button.

12. Verify that the Authorization Code option is selected.

13. Click to select the Implicit (Hybrid) option. Two additional fields are displayed.

14. Click to select the Allow ID Token with implicit grant type option.

15. Verify the Logon Initiated by field is set to App Only.

> **NOTE:** OneStream does not support Okta initiated authentication.

16. Click the **Save** button.

17. In Client Credentials, click the Show button to display the Client secret key.

18. Take note of the Client ID and Client secret key and please copy this information to the last page.

19. Select the Assignment menu and assign the application to any users who will use OneStream.

Native Application Setup
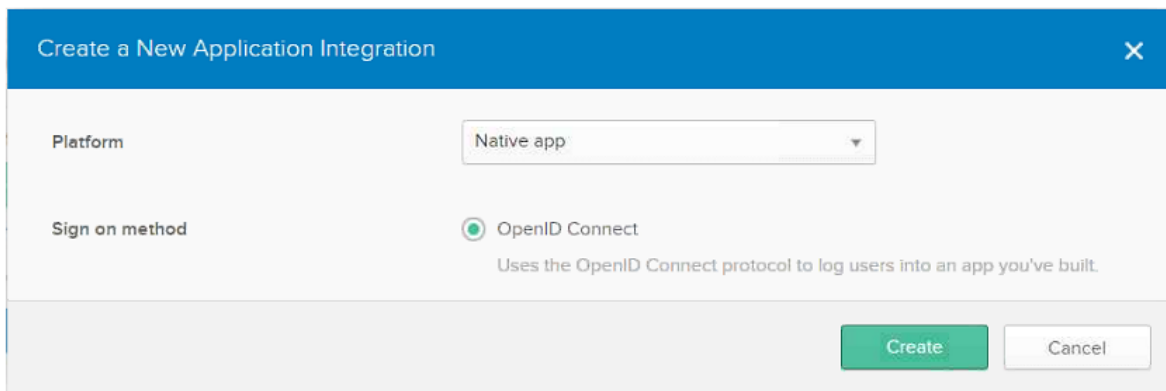
This section will describe the setup of the Okta application used by OneStream to authenticate clients accessing the Application using the OneStream App for Windows, the Excel Add-In, and Studio.

> **NOTE:** The Client ID will be different than the examples shown.

1. Click the **Back to Applications** button to return to the Applications screen.

2. Click the **Add Application** button.

3. Click the **Create New App** button.
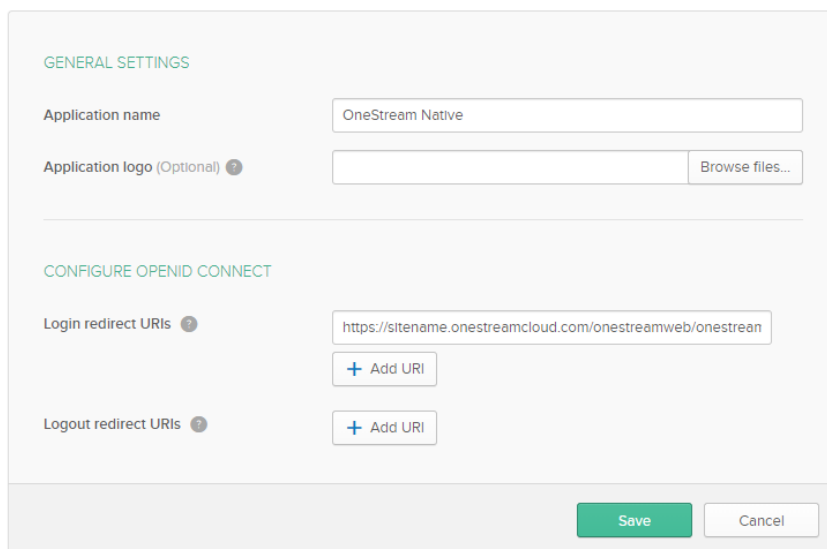
4. In the Create a New Application Integration window, click the selection arrow in the Platform field and select the Native app option. The Sign on method value automatically defaults to OpenID Connect.
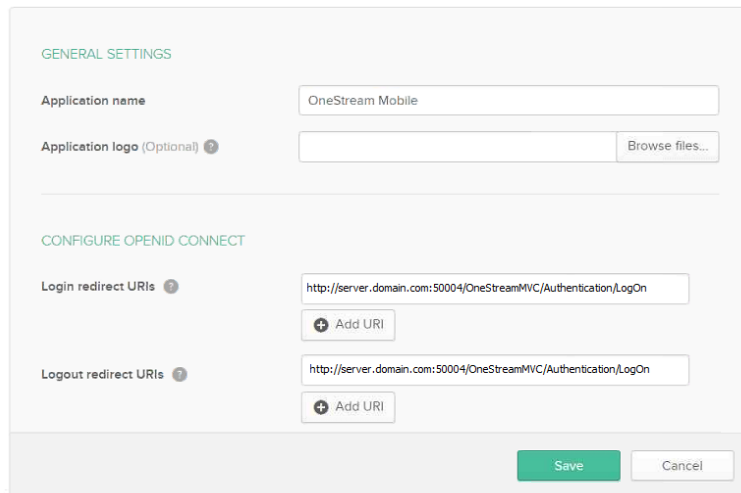
5. Click the **Create** button.

6. In the Create OpenID Connect Integration window, enter "OneStream Native" in the Application label field.

7. Enter the following into the Login redirect URIs field: http (s)://<servername>:<port>/onestreamweb/onestreamlogoncallback.aspx/

8. Please copy this URL to the last page Note: This value must be entered exactly.

9. Click the **Save** button. The General Settings widow appears.

10. Click the **Edit** button.

11. Verify that the Authorization Code option is selected.

12. Click to select the Resource Owner Password option. This is used for backend API authentication.
    Note: Verify that the option for Implicit(Hybrid) Grant Type is NOT selected.

13. Take note of the Client ID and please copy this information to the last page.

14. Verify the Use PKCE (for public clients) option is selected.

15. Click **Save**.

16. On the Assignments tab assign the application to any users who will use OneStream. Except for Client ID, you should see something very similar to the screenshots below:
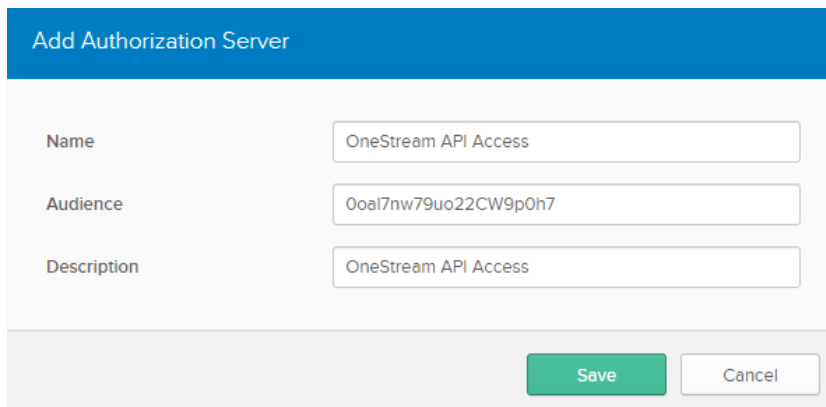
Authorization Server for OneStream API Access

If you intend to use the OneStream API from PowerShell, the following steps must be completed. Note: There may be additional Okta licensing cost associated with this option.

1. Click **Security > API**.

2. Click the **Add Authorization Server** button.

3. In the Add Authorization Server window, enter "OneStream API Access" in the Name field.

4. Refer to the Client ID value that was created/stored for the Native App and enter it in the Audience field.

5. In the Description field, enter an appropriate value.

6. Click the **Save** button.

7. Click the newly added server, then click the Settings menu.

8. Review the value in the Issuer field.

9. Select the last component of the uri stored in the Issuer field and please copy this value to the last page. For example: assuming the Issuer value is 'https://example.okta.com/oauth2/ausazrxxxxxxxxx750h7'

    copy: ausazrxxxxxxxxx750h7

Audience	0oal7nw79uo22CW9p0h7
Description	OneStream API Access
Issuer	Custom URL (https://sso-dev.onestreamsoftware.com/oauth2/ausl7vlp44O4lhp7j0h7)
Metadata URI	https://sso-dev.onestreamsoftware.com/oauth2/ausl7vlp44O4lhp7j0h7/.well-known/oauth-authorization-server
Signing Key Rotation ⓘ	Automatic
Last Rotation	31 May 2019

10. Leave Scope and Claims menus unchanged.

11. Click **Access Policies** and then click the **Add New Access Policy** button.

12. Click the Name field enter Access to OneStream API.

13. Enter a description as appropriate.

14. Verify that the following clients option is selected for the Assign to field.

15. Refer to the Client ID value that was created/stored for the Native App and enter it in the field that appears

16. Click the **Create Policy** button to save. The new policy is added and displays in the Access Policies screen.

17. Click the **Add Rule** button.

18. In the Edit Rule window, enter an appropriate value in the Name field.

19. Verify that the Resource Owner Password option is selected in the IF Grant type | Client acting on behalf of a user field.

20. Click the **Create Rule** button to save.

OneStream Web Server Configuration

This section describes the configuration of the OneStream Web Server Configuration file to support Okta SSO.

> **NOTE:** See the following examples for the results of these steps. Several values, including Client ID and Secret Key will be different than the examples shown.

1. Launch the OneStream Server Configuration Utility application.

2. Select **File > Open Web Server Configuration File**.

3. In the WebServer Configuration File window, expand the Single Sign On Identity Provider section.

4. Select Okta as the SSO Identity Provider type.

5. Expand the Okta Identity Provider section.

6. Enter your Okta organization URL into the Okta Domain field.

7. Enter the Logon URI value previously noted during the Web Application setup into the Okta Web Reply URL.

8. Enter https://[ServerName]:[PortNumber]/OneStream/OneStreamLogonCallback.aspx/ into the Okta Redirect Uri field.

9. Enter the Client ID value previously noted during the Native Application setup into the Okta Native Client ID field.

10. Enter the Client ID value previously noted during the Web Application setup into the Okta Web Client ID field.

11. Enter the Client secret value previously noted during the Web Application setup into the Okta Client Secret Key field.

12. Enter the Logon URI value previously noted during the Mobile Application setup into the Okta MVC Mobile Reply URL.

13. Enter the Client ID value previously noted during the Mobile Application setup into the Okta MVC Mobile Client ID field.

14. Enter the Client secret value previously noted during the Mobile Application setup into the Okta MVC Mobile SSO Secret Key field.

15. Click in the Okta Response Type field and enter the following into the field: "code id_token".

16. Leave Okta API Endpoint URI blank.

17. Click in the Okta Scopes field and enter the following into the field: "openid email phone address profile".

18. Enter the Authorization Server ID value previously noted during the Authorization Server setup into the Okta Authorization Server ID field.

19. Set the SSO Identity Provider Type field to Okta.

20. Click File | Save to save the changes to the Web Server Configuration file.

OneStream Application Server Configuration

This section describes the configuration of the OneStream Application Server Configuration file to support Okta SSO.

1. Launch the OneStream Server Configuration Utility application.

2. Open the Web Server Configuration File and select the value in the Okta Client Secret Key field.

3. Copy the Okta Client Secret Key value and then close the window.

4. Select File | Open Application Server Configuration File.

5. Navigate to the location where the XFAppServerConfig.xml file is stored.

6. Select the file, then click the Open button.

7. In the Authentication section, click inside the External Authentication Providers field. An ellipsis appears.

8. Click the Ellipsis.

9. In the External Authentication Providers window click Add.

10. Paste the value copied in Step 3 into the External Provider Web SSO Secret Key field.

11. Click the selection arrow in the Authentication Provider Type field and select OktaSSO.

12. Click in the Name field and enter Okta.

13. Click OK to save the changes.

14. Click File | Save to save the changes to the Application Server Configuration file.

15. Reset IIS on all OneStream servers.
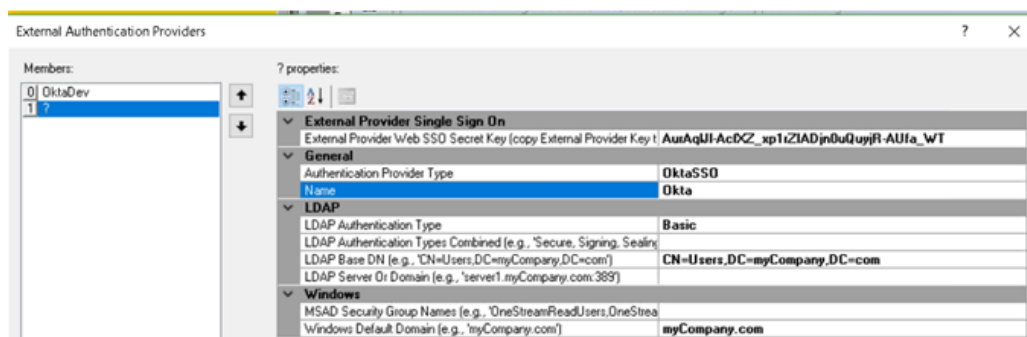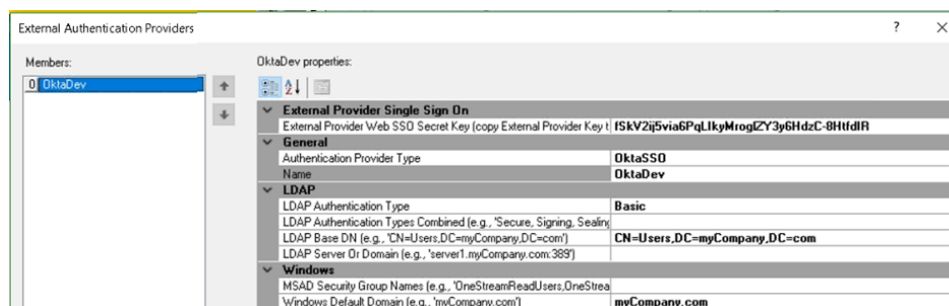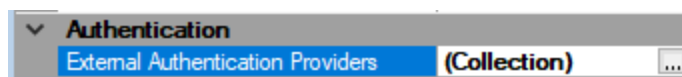
User Application Security Configuration

Add any users that will use the Okta SSO authentication method. Under the authentication section, select Okta SSO for the External Authentication Provider and the External Provider User Name should be the same as the user ID/email address listed in the Okta.

If the Okta SSO works properly, after logging into Okta, it should redirect to the OneStream application URL and pass the Okta SSO token to OneStream. This will populate the user field on the login screen and allow for application selection without re-entering user credentials.

PingFederate Configuration

The section provides an overview for configuring access to PingFederate. Before performing these steps, ensure you installed and configured PingFederate as described in "Appendix 9: Installing and Configuring PingFederate " on page 177.

The steps in this section provide information on configuring PingFederate OAuth web, mobile, and native clients (Excel, Studio and the OneStream Windows App). The OneStream server configuration is also provided.

Web Application Setup

This section will describe the setup of the OAuth client used by OneStream to authenticate users accessing the Microsoft Silverlight application.

> **NOTE:** See the following examples of these steps. The Client ID and Secret values will be different than the examples shown.

1. Logon to PingFederate Admin console.

2. Navigate to OAuth Server, then to Clients.

3. Click **Create New**.

4. Enter a value for Client ID (ex. OneStreamWeb), take note of the value as it will be needed for OneStream configuration.

5. Enter a value for Name (ex. OneStreamWeb).

6. Enter a value for Description (Optional).

7. On Client Authentication types, click the Client Secret radio button. Click Change Secret and then click Generate Secret. Take note of the value as it will be needed for OneStream configuration.

8. Redirection URIs: Enter the respective Uris for OneStream web application. Take note of these values for use in configuring OneStream:

    a. http://ServerName:PortNumber/OneStreamWeb/OneStreamXF.aspx

    b. http://ServerName:PortNumber/OneStreamWeb/OneStreamWindowsApp.aspx

9. Select the Bypass Authorization Approval box.

10. Select the following Grant Types:

    - Authorization Code

    - Implicit

11. Click **Save**.

Mobile Application Setup

This section will describe the setup of the OAuth client used to authenticate users accessing the OneStream Mobile application.

> **NOTE:** See the following pages for example screenshots of these steps. The Client ID and Secret values will be different than the examples shown.

1. Logon to PingFederate Admin console.

2. Navigate to OAuth Server, then to Clients.

3. Click **Create New**.

4. Enter a value for Client ID (ex. OneStreamMobile), take note of the value as it will be needed for OneStream configuration.

5. Enter a value for Name (ex. OneStreamMobile).

6. Enter a value for Description (Optional).

7. On Client Authentication types click the Client Secret radio button. Click Change Secret and then click Generate Secret. Take note of the value as it will be needed for OneStream configuration.

8. Redirection URIs: Enter the respective Uri for OneStream Mobile application. Take note of this value for use in configuring OneStream:

    http://ServerName:PortNumber/OneStreamMvc/Authentication/Logon

9. Select the Bypass Authorization Approval box.

10. Select the following Grant Types:

    * Authorization Code

    * Implicit

11. Click **Save**.

Native Application Setup

This section will describe the setup of the OAuth client used to authenticate users accessing the OneStream native applications such as OneStream Windows App, the Excel Add-In, and Studio.

> **NOTE:** See the following pages for example screenshots of these steps.

1. Logon to PingFederate Admin console.

2. Navigate to OAuth Server, then to Clients.

3. Click **Create New**.

4. Enter a value for Client ID (ex. OneStreamClient), take note of the value as it will be needed for the OneStream configuration.

5. Enter a value for Name (ex. OneStreamClient).

6. Enter a value for Description (Optional).

7. On Client Authentication types click on None.

8. Redirection URIs: Enter the redirect Uri http(s)://[ServerName]:[PortNumber]/onestreamweb/OneStreamLogonCallback.aspx/ needed to receive the authentication response for the native applications. Take note of this value for use in configuring OneStream.

9. Select the Bypass Authorization Approval box.

10. Select the following Grant Types:

    - Authorization Code

    - Refresh Token

    - Resource Owner Password Credentials

    - Access Token Validation (Client is a Resource Server)

11. Click **Save**.

Access Token Management in Ping Admin console

When creating (or modifying) an instance of Access Token Manager in Ping Admin console, it is important to specify a path in PingFederate server to publish a JSON Web Key Set with the key/certificates that will be used for signature verification. For full configuration refer to Appendix 8. Take note of the path as it will used for OneStream configuration.

OneStream Web Server Configuration

This section describes how to set up the OneStream Web Server Configuration file to support PingFederate SSO.

> **NOTE:** See the following example for the result of these steps. Several values, including Client ID and Secret Key will be different than the examples shown.

1. Launch the OneStream Server Configuration Utility application.

2. Select **File** > **Ope**n Web Server Configuration File.

3. In the WebServer Configuration File window, expand the Single Sign On Identity Provider section.

4. Select PingFederate as the SSO Identity Provider type.

5. Expand the PingFederate Identity Provider section.

6. Enter your PingFederate server URL into the PingFederate Domain field.

7. Enter the Logon URI values previously noted during the Web Application setup into the PingFederate Web Reply URL and PingFederate Web Reply URL for Windows App Launch Page.

8. Enter https://[ServerName]:[PortNumber]/OneStream/OneStreamLogonCallback.aspx/ into the PingFederate Native Redirect Uri field.

9. Enter the Client ID value previously noted during the Native Application setup into the PingFederate Native Client ID field.

10. Enter the Client ID value previously noted during the Web Application setup into the PingFederate Web Client ID field.

11. Enter the Client secret value previously noted during the Web Application setup into the PingFederate Web Client Secret Key field.

12. Enter the Logon URI value previously noted during the Mobile Application setup into the PingFederate Mobile Reply URL.

13. Enter the Client ID value previously noted during the Mobile Application setup into the PingFederate Mobile Client ID field.

14. Enter the Client secret value previously noted during the Mobile Application setup into the PingFederate Mobile SSO Secret Key field.

15. Click in the PingFederate Response Type field and enter the following into the field: "code id_token".

16. Click in the PingFederate Scopes field and enter the following into the field: "openid email phone address profile".

17. Click in the Path on the PingFederate server to publish a JSON Web Key Set with the keys/certificates that can be used for signature verification then enter "/ext" followed by the path previously noted (ex. "/ext/oauth/jwks").

18. Click **File > Save** to save the changes to the Web Server Configuration file.

OneStream Application Server Configuration

This section describes the configuration of the OneStream Application Server Configuration file to support PingFederate SSO.

1. Launch the OneStream Server Configuration Utility application.

2. Open the Web Server Configuration File and select the value in the PingFederate Client Secret Key field.

3. Copy the PingFederate Client Secret Key value and then close the window.

4. Select **File > Open Application Server Configuration File**.

5. Navigate to the location where the XFAppServerConfig.xml file is stored.

6. Select the file, then click **Open**.

7. In the Authentication section, click in the External Authentication Providers field. .

8. Click the ellipsis.

9. In the External Authentication Providers window click **Add**.

10. Paste the value copied in Step 3 into the External Provider Web SSO Secret Key field.

11. Click the selection arrow in the Authentication Provider Type field and select PingFederateSSO.

12. Click in the Name field and enter PingFederate.

13. Click **OK** to save the changes.

14. Click **File** > **Save** to save the changes to the Application Server Configuration file.

15. Reset IIS on all OneStream servers.

16. If SSL Settings > RequireSSL setting is turned on in IIS, ensure **Accept Client Certificates** option is enabled.
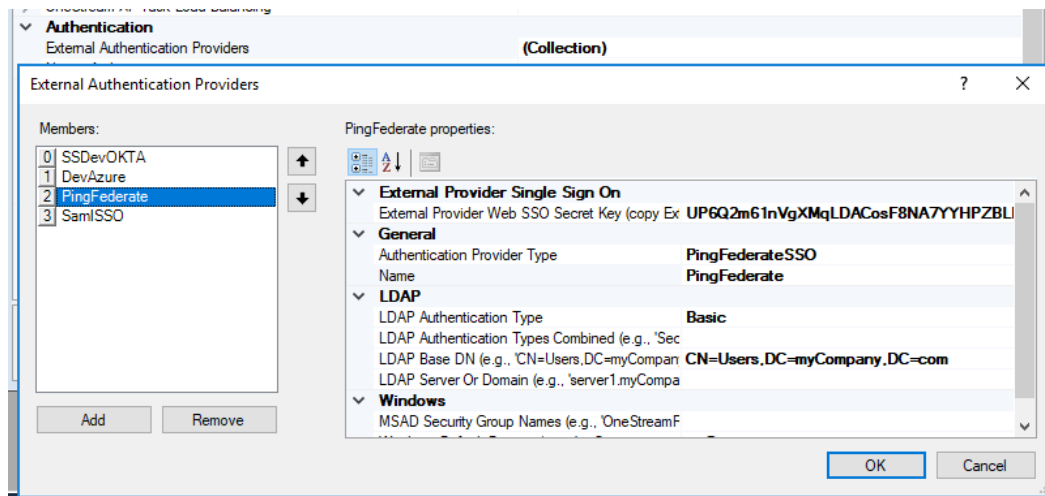
User Application Security Configuration

Add any users that will use the PingFederateSSO authentication method. Under the authentication section, select PingFederateSSO for the External Authentication Provider and the External Provider User Name should be the same as the user ID/email address returned by the adapter in PingFederate.
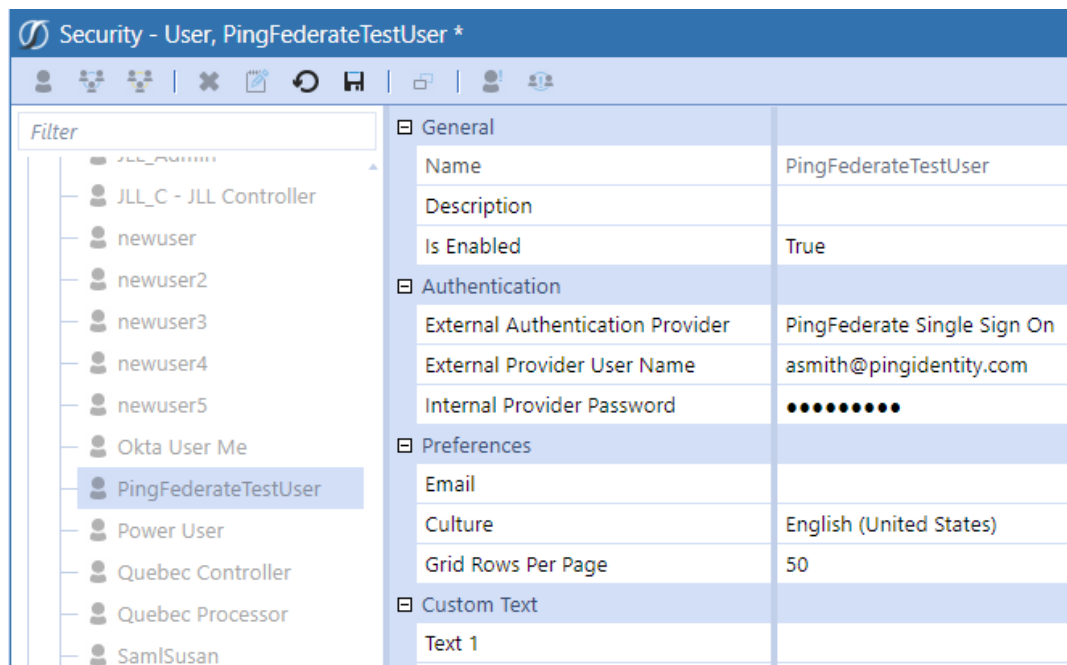
If PingFederateSSO works properly, after logging into PingFederate logon screen (if no IWA adapter is in use), it should redirect to the OneStream application URL and pass the PingFederateSSO token to OneStream. This will populate the user field on the login screen and allow for application selection without re-entering user credentials.

SAML 2.0 SSO Configuration

OneStream supports single sign-on (SSO) logins through SAML 2.0. A SAML 2.0 identity provider (IdP) can take many forms, an example of which is a self-hosted Active Directory Federation Services (ADFS) server. ADFS is a Microsoft service that allows the secure sharing of identity information between trusted entities.

Another supported SAML IdP is Okta. Sections that follow describe steps to configure SAML 2.0 authentication of applications with both ADFS and Okta as identity providers. Configurations with other identity providers will vary.

> **NOTE:** SAML 2.0 does not initiate an authentication flow with a username and password from the service provider therefore using the Client API (PowerShell,PRM, etc) with SAML 2.0 is not supported. This may limit the automation capabilities available.

Configure SAML 2.0 SSO with ADFS and OneStream as Service Provider

OneStream supports single sign-on (SSO) logins through SAML 2.0 with ADFS. The following instructions cover of both ADFS as Identity Provider and OneStream Applications as Service Provider.

Requirements

To use ADFS to log in to your OneStream instance, the following components are needed:

1. An Active Directory instance where all users have an email address attribute.

2. A server running Active Directory Federation Services (ADFS).

3. A server running OneStream .

4. The ADFS signing SSL certificate and its public fingerprint.

5. A private certificate installed in the web server to sign the logout requests to ADFS and its public fingerprint.

6. If using OneStream Windows App:

   • A client Windows machine to run the application.

   • A trusted certificate issued to localhost installed on all client machines for hosted SSL and its thumbprint.

ADFS Endpoints

After a full installation of ADFS is complete, take note of the endpoint values for Federation Service Identifier, SAML 2.0/WS-Federation and Federation Metadata. Typically, if the defaults were in place these values will be:

• Federation Service Identifier: https://adfs_sitename/adfs/services/trust

• Federation Metadata URL : https://adfs_sitename/FederationMetadata/2007-06/FederationMetadata.xml

• SAML 2.0/W-Federation URL : https://adfs_sitename/adfs/ls/

Download ADFS metadata and save as xml, ex: AdfsMetadataWeb.xml. The file contains information about both SP and IDP. It also contains few tags which are not supported by OneStream.

Remove the following tags from federation AdfsMetadataWeb.xml:

1. <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#"> ... </X509Data></KeyInfo></ds:Signature>

2. <RoleDescriptor xsi:type="fed:ApplicationServiceType" ... </EndpointReference></fed:PassiveRequestorEndpoint></RoleDescriptor>

3. <RoleDescriptor xsi:type="fed:SecurityTokenServiceType" .... </EndpointReference></fed:PassiveRequestorEndpoint></RoleDescriptor>

4. <SPSSODescriptor WantAssertionsSigned="true" protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"> .... </SPSSODescriptor>

Adding Relying Party Trust

In ADFS Management Console add a Relying Party Trust for OneStream Web application:

1. Right-click on Relying Party Trusts.

2. Add Relying Party Trust.

3. Leave Claims Aware checked, click **Start**.

4. In Select Data Source check Enter data about the relying party manually option, click Next.

5. Enter a name for the party trust in the Display name field, ex OneStreamWeb.

6. Optionally enter descriptive notes about the party trust in the Notes field, click Next.

7. Click **Next** in the Configure Certificate tab.

8. Check Enable support for the SAML 2.0 SSO service URL and enter.

    https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx in Relying party trust SAML 2.0 SSO service URL, click **Next**.

9. In Configure Identifiers tab, Relying party trust identifier add:

    https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx, click Next

10. In Choose Access Control Policy tab, click **Next**.

11. Review the summary tabs, click **Next**, then **Close wizard**.

Configuring Relying Party Trust

1. Right click on the relying party just created, then click Properties.

2. Click on Endpoints tab and enter these additional endpoints:

    * Endpoint type: SAML Assertion Consumer

    * Binding: Post

    * Index: 0

    * Trusted URL:

        https://<web server host name>/onestreamweb/onestreamxf.aspx

    * Endpoint type: SAML Assertion Consumer

- Binding: Post

- Index: 1

- Trusted URL: https
  https://sitename.onestreamcloud.com/onestreamweb/onestreamwindowsapp.aspx

- Endpoint type: SAML Assertion Consumer

- Binding: Post

- Index: 2

- Trusted URL:

  https://<web server host name>/OneStreamWeb/OneStreamLogonCallback.aspx/

- Endpoint type: SAML Assertion Consumer

- Binding: Redirect

- Index: 3

- Trusted URL:

  https://<web server host name>/OneStreamWeb/OneStreamLogonCallback.aspx/

- Endpoint type: SAML Logout

- Binding: Post

- Index: 3

- Trusted URL: add
  https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx

3. Click on the Signature tab and add the public portion of the web server's certificate.

4. Close the Properties window.

Configure Claims Issuance Policy

1. Right-click the relying party, then click **Edit Claim Issuance Policy**.

2. Add these transform rules:

   Claim Rule Name: Attributes

   Attribute store: Active Directory

   Rule Template: Send LDAP attributes as Claims

   Add mappings:

   - Given-name to Given Name

   - Surname to Surname

   - E-Mail-Addresses to E-Mail Address

- Click **OK** to close.

  Claim Rule Name: Windows Account Name

  Attribute store: Active Directory

  Rule Template: Send LDAP attributes as Claims

  Add mappings:

- SAM-Account-Name to Name ID

  Click OK to close

  Claim Rule Name: UpnToAppId

  Attribute store: Active Directory

  Rule Template: Send LDAP attributes as Claims

  Add mappings:

- User-Principal-Name to Application Identifier

  Click OK to close.

Configure Adfs Relying Party Trust to return both the message and the assertion signed

Open a PowerShell console and run a Set-AdfsRelyingPartyTrust with the following arguments:

Set-ADFSRelyingPartyTrust -TargetName {XFApplicationName} -SamlResponseSignature "MessageAndAssertion"

Where {XFApplicationName} refers to #5 in Adding Relying Party Trust above.

Configure OneStream for SAML 2.0 with ADFS

1. Place AdfsMetadata.xml in the same shared folder as OneStream's XFWebServerConfig.xml and XFAppServerConfig.Xml files.

2. Launch OneStream Server Config utility.

3. Open XFWebServerConfig.xml file.

4. Create a SAML Identity Provider section with the below attributes/values.

5. Enter SAML specific values as below:

SAML 2.0 Identity Provider	
Multiple ACS URLs Supported.	True
ACS URL for Web Application.	https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx
ACS URL for Windows Application.	https://sitename.onestreamcloud.com/onestreamweb/onestreamwindowsapp.aspx
ACS URL for Mobile Application.	https://sitename.onestreamcloud.com:50004/onestreammvc/authentication/logon
Unique ID for Windows Application.	https://sitename.onestreamcloud.com/onestreamweb/onestreamlogoncallback.aspx
Single Sign-On URL for Web Application.	https://xxxxxxxxxxx.oktapreview.com/app/ossdev4xxxxxxxxxxx0_onestreamwebsaml_1/exkeg5fzm3bIQ1AXc0h7/sso/saml
Single Sign-On URL for Native Applications.	https://xxxxxxxxxxx.oktapreview.com/app/ossdev47xxxxxxxxxx0_onestreamwebsaml_1/exkeg5fzm3bIQ1AXc0h7/sso/saml
Single Sign-On URL for Mobile Application.	https://xxxxxxxxxxx.oktapreview.com/app/ossdev47xxxxxxxxxx0_onestreammvcsaml_1/exkersovteU9GOnfW0h7/sso/saml
Entity ID for Web Application.	http://www.okta.com/xxxxxxxxxxxxxx1AXc0h7
Entity ID for Native Applications.	http://www.okta.com/xxxxxxxxxxxxxx1AXc0h7
Entity ID for Mobile Application.	http://www.okta.com/xxxxxxxxxxxxxxOnfW0h7
Metadata File Name for Web Application	OktaIdpMetadataWeb.xml
Metadata File Name for Mobile Application	OktaIdpMetadataMvc.xml
Metadata Content For Web Application.	<?xml version="1.0" encoding="UTF-8"?><md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" entityID
Metadata Content For Native Applications.	
Metadata Content For Mobile Application.	
Signing Certificate Store Name	Root
Signing Certificate Store Location	LocalMachine
Signing Certificate Find Mode	FindBySerialNumber
Signing Certificate Find Value	015cxxxxxxxxxxxxxxad46
Logout Certificate Store Name	Root
Logout Certificate Store Location	LocalMachine
Logout Certificate Find Mode	FindBySerialNumber
Logout Certificate Find Value	30831ffdfxxxxxxxxxxxx3ba2bb6afdf
User Inactivity Timeout (minutes)	12

Multiple ACS URLs Supported:

True (This is IdP specific. Okta supports multiple ACS values)

ACS URL for Web Application:

https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx

ACS URL for Windows Application:

https://sitename.onestreamcloud.com/onestreamweb/onestreamwindowsapp.aspx

ACS URL for Mobile Application:

https://sitename.onestreamcloud.com:50004/onestreammvc/authentication/logon

Unique ID for Windows Application:

https://sitename.onestreamcloud.com/onestreamweb/onestreamlogoncallback.aspx/

Single Sign-On URL for Web Application:

https://adfs_sitename adfs/ls/

Single Sign-On URL for Mobile Application:

https://adfs_sitename/adfs/ls/

Identity Provider Entity ID for Web Application:

https://adfs_sitename/adfs/services/trust

Identity Provider Entity ID for Mobile Application:

https://adfs_sitename/adfs/services/trust

Identity Provider's Metadata File Name for Web Application:

AdfsIdpMetadata.xml

Identity Provider's Metadata File Name for Mobile Application:

AdfsIdpMetadata.xml

After installing the public key of ADFS Signing certificate on the web server enter values indicating how to programmatically extract the installed certificate for Identity Provider's Certificate.

Example:

Identity Provider's Certificate Store Name: Root

Identity Provider's Certificate Store Location: LocalMachine

Identity Provider's Certificate Find Mode: FindBySerialNumber

Identity Provider's Certificate Find Value: xxxxxxxxxxxxxxxxxxx

Create and install in the web server, a private certificate, install the public key in the Adfs server and enter the respective values for the Single Logout certificate section:

Logout Certificate Store Name: My (Personal)

Logout Certificate Store Location: LocalMachine

Logout Certificate Find Mode: FindBySerialNumber

Logout Certificate Find Value: xxxxxxxxxxxxxxxxxxx

Open AdfsIdpMetadata.xml, copy the content and paste it into Identity Provider's Metadata Content For Web Application

Save and close.

Open XFAppServerConfig.xml file

Create a SAML authentication SSO provider section

Enter https://adfs_sitename/adfs/services/trust for External Provider SSO Key field

Save and close.

Configure Windows client machine for OneStream Windows App SAML 2.0 Authentication with ADFS

> **NOTE:** The recommended installation method is to use Click Once deployment.

If installing versions prior to 5.0, see Appendix 10 on how to configure OneStream native application authentication with SAML 2.0 and ADFS

For OneStream version 5.0 and above, no client configuration is required.

Appendix 2: Configuration Checklist

Prepare the Service Accounts

1. Create the IIS Application Pool Service Account used for inter-server communication. Ensure full access to the application server file share.

2. Create the SQL Server Native User Account (Preferred), enabling the Public and DBOwner roles required for databases.

3. Enable the Public and DBOwner server roles required for the SQL Server Master Database.

OneStream uses partitioning with other advanced SQL Server features which require SQL Server to make updates to the Master Database during the schema creation process.

SQL Server Database Connection String

Use Pooling
True

Connection Pool Limit
3000

Connection Timeout
60

Application Server(s)

1. IIS Application Pool Advanced Settings (OneStreamAppAppPool)

2. Identity = Service Account

3. Configuration File

4. Specify file share path.

5. Update ASP.Net Configuration File

6. Specify Shared Application Server Configuration File Path

> **NOTE:** Configuration file must be named XFAppServerConfig.xml

Web Server(s)

1. IIS Application Pool Advanced Settings (OneStreamWebAppPool)

2. Identity = Service Account

3. Configuration File

4. Application Server Cluster

5. Update ASP.Net Configuration File

6. Specify Shared Web Server Configuration File Path
    Note: Configuration file must be named XFWebServerConfig.xml

Appendix 3: Performance Optimization Checklist

Database Server Memory

Get better performance by providing a significant amount of RAM to the database server to allow the SQL Server to cache large portions of a database. Memory requirements depend on each client's application specifications.

Database File IO

This spreads My Company Name, LLC database files across multiple disks.

Database Authentication

For better performance, consider using SQL Server authentication rather than Windows integrated authentication, which showed a 15-25% performance decline in testing. This is due to the high degree of multi-threading used by OneStream causing many database connections to be created simultaneously and requiring many database connection authentication calls.

Database Properties

Enable **Autogrowth** for all database files and use ten Megabytes. See
http://support.microsoft.c om/kb/315512
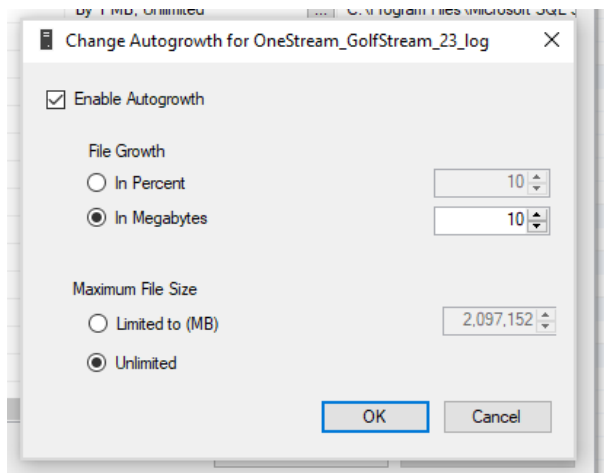
Database Instance Tuning

SQL Server Memory Parameters

Max Server Memory (In MB) (Recommend Value = [Server RAM – 2GB])

Database Server

Performance testing shows significant improvements with SQL Server over base 2016, so we recommend SQL server 2012, 2014,2016, 2017, 2019 Enterprise Edition.

Application Server

Create separate application servers for each server type.

General Server:

- High Server Demands (Concurrency)

- User interface request, queries, and reports

Stage Server:

- High Server Demands (Mapping)

- Data loading and transformation

- May use significant amounts of CPU time and RAM.

Consolidation Server:

- High Server Demands (Calculations)

- Analytic Model Calculations and Consolidations

- May use significant amounts of CPU time and RAM.

Appendix 4: Troubleshooting

Client Web Connection Terminates Before Web Service Returns Content

If a user has connection issues when trying to log in or during long running web service calls, check the error log for remote server error entries.

Example:
Description: [HttpWebRequest_WebException_RemoteServer]

Arguments: NotFound

Possible solution:

1. Determine if the user's virus scan software is applying network filters to the connection.

2. Check the registry setting below:

   HKeyCurrentUser\Software\Microsoft\Windows\CurrentVersion\Internet Settings

   ReceiveTimeout should be **36000000**.

   This value may be decreased by a virus or by any anti-virus program, causing client connections to time-out and "Server Not Found" errors.

3. Ensure configuration files are available.

   Make sure that the application server and web server files are in the Configuration folder on the file share and use proper naming conventions.

Long Running Server Process Hangs or Stops With Logging Errors

If there is a hung process that does not appear to be completing, (e.g. consolidation) then check the Event Viewer, in the summary window, then the information section, look for WAS in the sources field. If there are errors regarding WAS (Windows Activation Service) try the following.

1. Open IIS Manager, Click on Application Pools, Click on the OneStream Application Pool, then Advanced settings, and scroll down to Idle Time-out (minutes) parameter. (The standard setting is 20 minutes. Follow up with OneStream Support for further options. TBD)

2. Check Firewall settings to see that they allow two-way traffic. A rule to open the port number to allow traffic through may need to be created. Port numbers for OneStream are 50001 & 50002.

Web Server Not Communicating With Application Server

If a Web service error is received when first pulling up XF, confirm that the application server is properly configured. To check this put in the following URL to a browser:

http://<Servername>:50002/OneStreamApp/svc/Data.svc

If the application server is properly configured, a page that looks something like this will display:
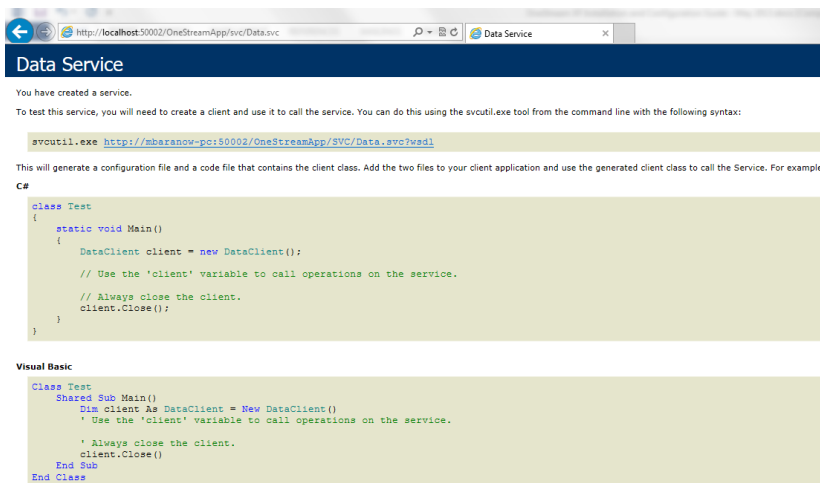
Difficulties Registering the OneStream Excel Add-In in Excel

1. To properly install the OneStream Excel Add-in, changes are required to the Windows registry. Ensure that the user has rights to update their own registry while doing installations, i.e. that they are an administrator of their own machine or have similar privileges. Also, it is best of the end user is logged in while installing the Add-in and not an IT representative.

2. Changes made to the Windows registry for the OneStream Excel Add-in during the installation are made only for the user who is doing the installation. For example, if some on the Information Technology staff is logged into the user's machine to do the installation, the business user will not be able to access the OneStream Excel Add-in.

3. The registration process of the OneStream Excel Add-in requires certain Microsoft .NET Framework rights to execute a program stored in that folder. The folder is C:\Windows\Microsoft.NET\Framework (or Framework64 if the user is running the 64-bit version of Excel). To manually register the Add-in, first open a Command Prompt by clicking Start | Run and type "cmd".
   Type this command:

   cd C:\Program Files\OneStream Software\OneStream Excel AddIn

   (or wherever your OneStream Excel Add-in is installed)

   Then type this command:

   C:\Windows\Microsoft.NET\Framework64\v4.0.30319\RegAsm.exe OneStreamExcelAddIn.dll

4. Other Office Add-ins may conflict with the OneStream Excel Add-in, so discuss other registered Add-ins when discussing installation issues with OneStream Support.

5. The OneStream Excel Add-in will not register with a machine properly if Microsoft Office has not been installed with the required settings. Ensure that the optional ".NET Programmability Support" is selected under Excel when installing Microsoft Office.

6. If the OneStream Excel Add-in is installed and appears registered but the user still cannot see the OneStream menu in the Excel ribbon, check to see if the Add-in is disabled. To do so, go to File | Excel Options | Add-ins and scroll to the bottom of the list of Add-ins to see if the OneStream Excel Add-in is listed under "Disabled Application Add-ins." If it is, lower on that dialog under Manage, select Disabled Items and click Go. Select the OneStream Excel Add-in, click Enable and click OK.

Brower Issues

1. If you use the 32-bit version of Internet Explorer, you may experience an out of memory situation after several hours of extended use. Restart the web browser to resolve this issue. If the user is working with a 64-bit version of their Windows operating system, we recommend that they use the 64-bit version of Internet Explorer or an another browser.

2. If you run both the client and server on a Virtual Machine and use the 64-bit version of Windows and the 32-bit version of Internet Explorer 11, browser errors may occur. If the user is working with a 64-bit version of their Windows operating system, we recommend they use the 64-bit version of Internet Explorer, upgrade to Internet Explorer 9 64 bit or higher, or use another browser. Microsoft also recommends disabling the PC Tablet Input Service or running IE outside of the VM when running in this mode.

Appendix 5: Setting Up Encrypted Database Connections

1. Launch the OneStream Database Configuration utility.

2. Right-click a database and select **Copy to Clipboard > Connection String**.
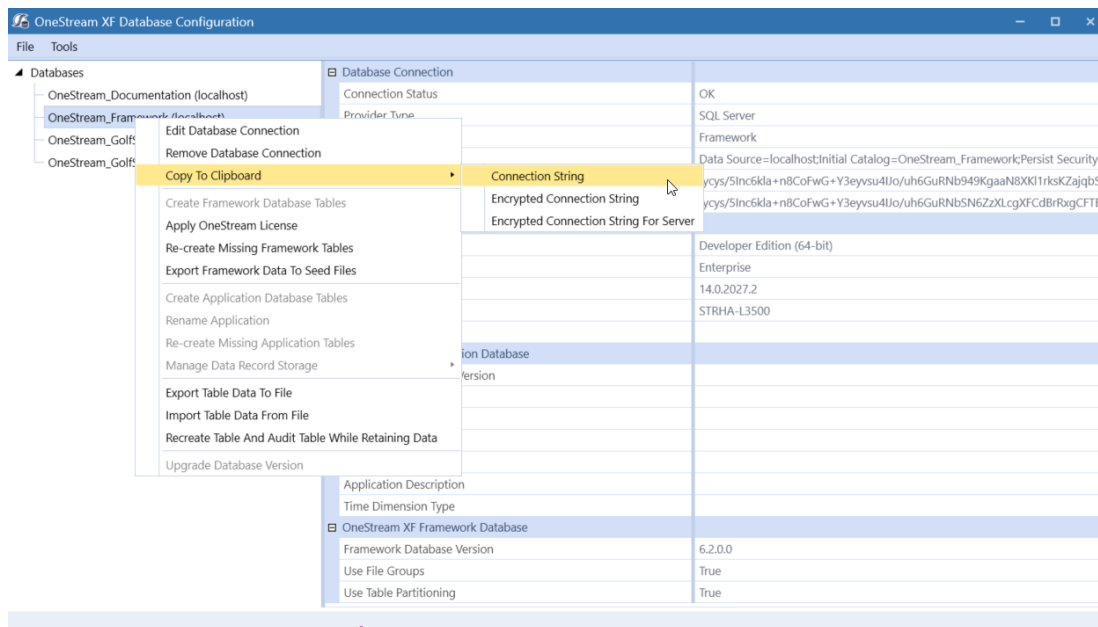
3. Copy the encrypted string.

   Email Connection String Example:

   ```
   SmtpHost=[smtp.office365.com], SmtpPort=[587],
   EnableSSL=[True], SmtpSourceMailAccount=
   [userID@youremaildomain.com],
   SmtpSourceMailAccountPassword=[password]
   ```

Email Connection String Example with From Address (optional)

```
SmtpHost=[smtp.office365.com], SmtpPort=[587],
EnableSSL=[True], SmtpSourceMailAccount=[username],
SmtpSourceMailAccountPassword=[password],
FromAddress=[userID@youremaildomain.com]
```

SAP Connection String Example:

```
"USER=YourUserID LANG=EN CLIENT=800 SYSNR=00 ASHOST=HostServerName
PASSWD=YourPassword"
```

4. The encrypted string is stored to the clipboard.

Encrypted Email String Example

XScpc
Dzo3an7EWAAp6Wvz1IZWE3Q+Fo8DSx0OpacS0Yp5X4sHXtiQGdAoNmNGU48xOa
3xD3C68yA17B8aXUbv02lj4921ErgN2R+E1qlLG24p98O8a62X0n/q4PS70xzo
sL7R9HzKWT0txtfbpJMUPYrIhCDz6Ubd52/buVABQyUxf2c0Y1BmgIKE/
/6eOcMCv0D5abX6oKbEmZLlms7vXyuicD+KaYheBNX/vbtkA4=

Encrypted SAP String Example

MF/HrDU0zQupeiyGGSUZ431S1guSfOCDoss4T7JYmiNm8BPTw7inI97W5en
ORZfrVN1Z8ADUHkavsRXKnomFKqBLmddbamOIt5s9bO3jxfXiLI
9B26SDQDKwJer1e6J**C**

5. Open the OneStream Application Server Configuration tool.

6. Open **Database Server Collections**.

7. Add a database connection.
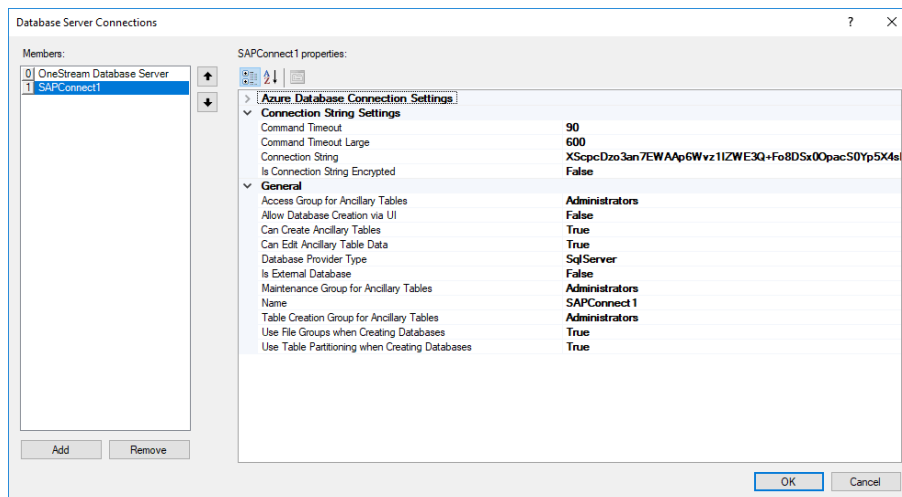
8. (Email Connection Only) In the OneStream Business Rule using the email connection, add the code to call the mail function:

```
'Prepare the message
Dim emailConnectionName As String = "OneStreamEmail"
Dim toEmail As New List(Of String)
toEmail.Add("tsmith@OneStreamSoftware.Com")
Dim subject As String = "Test Mail"
Dim body As String = "Test Mail Body"
Dim attachments As New List(Of String)
attachments.Add("\\share1\FileShare\Applications\GolfStream_v30\TestFile.csv")

'Send the message
BRApi.Utilities.SendMail(si, emailConnectionName, toEmail, subject, body, attachments)
```

9. (SAP Connection Only) In the OneStream Business Rule using the SAP connection add the code to call the mail function:

```
myR3Connection = BRApi.Database.CreateSAPConnection(si, sapConnectionName, openConnection)

si = SessionInfo
sapConnectionName = The name of the connection setup in the DB configuration.
openConnection = Boolean value stating whether or not to keep the connection open.
```

Appendix 7: Web.config Hardening Process Overview

ASP.NET uses a hierarchy of XML text files named Web.config to store configuration information that controls how IIS operates and how ASP.Net applications function on the Web Server.

These files are typically reviewed during server security audits. IT security teams and auditors will use various security scanning tools to determine if there are known vulnerabilities on the Web Server.

OneStream recommends making the following changes to the Web.config files on all of the Web Servers and Mobile Servers that are deployed in each of the OneStream environments. This will provide optimal security and consistent reporting by security scanning tools.

Customers hosted in the OneStream Cloud environment will have these changes managed by OneStream Cloud Support.

Web.config file changes

1. Stop IIS on the Web Server/Mobile Server prior to modifying the Web.config file.

2. Open File Explorer and navigate to the OneStream Software installation directory on the server:
    "C:\Program Files\OneStream Software\OneStreamWebRoot\OneStreamWeb"
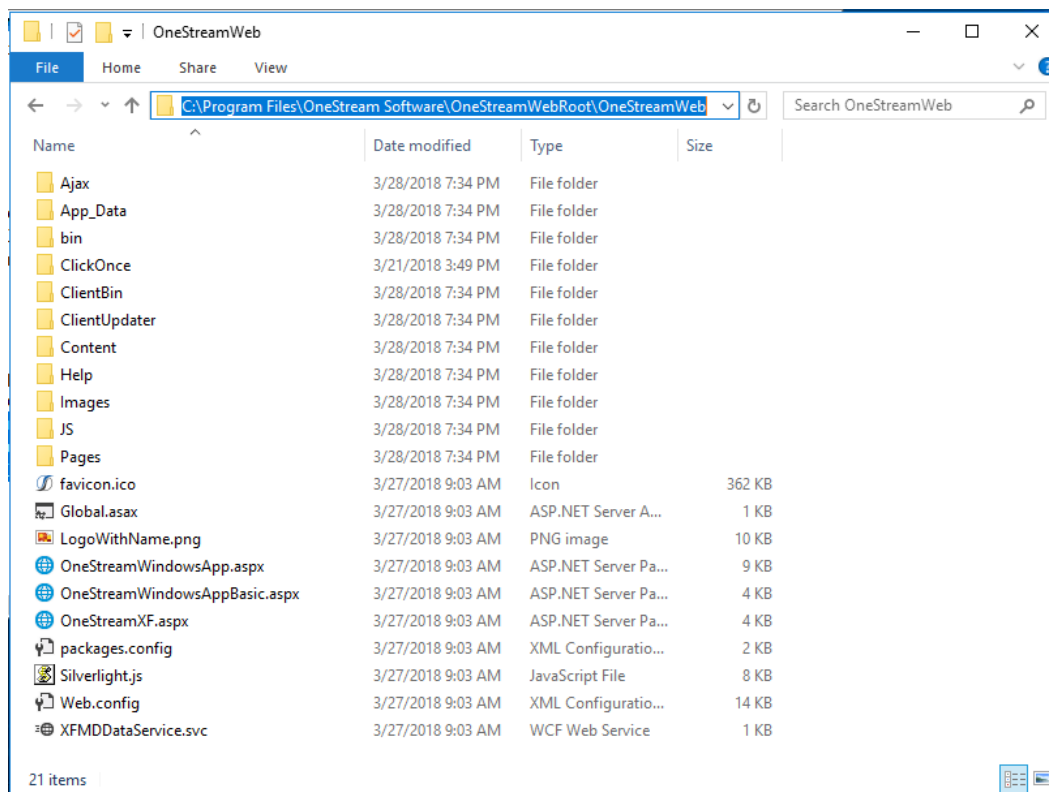
3. Right-click the Web.config file and open it in Notepad.
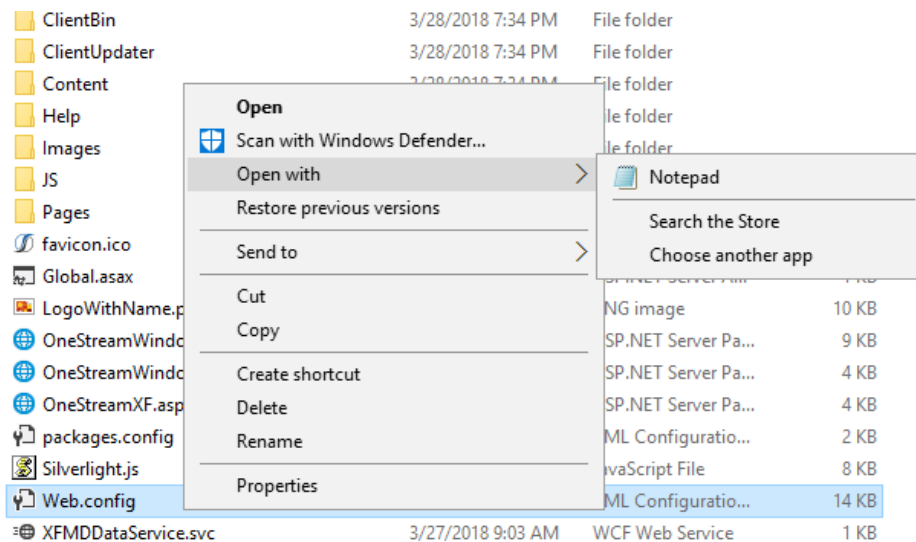
4. Find this string of text and delete it out of the web.config file:

```
<httpProtocol>
  <customHeaders>
    <!--Override the IE broweser's Compatibility View Settings for intranet
sites.-->
    <add name="X-UA-Compatible" value="IE=Edge" />
    <add name="X-Frame-Options" value="SAMEORIGIN" />
  </customHeaders>
</httpProtocol>
```

5. In the same file, find the following string and delete as well and save the updated file:

```
<!--Disabling OPTIONS per security assessment recommendation-->

<verbs allowUnlisted="true">

  <add verb="OPTIONS" allowed="false" />

</verbs>
```

```
<system.serviceModel>

  <serviceHostingEnvironment aspNetCompatibilityEnabled="true" />
```

6. Navigate to attributes:
    system.serviceModel>behaviors>serviceMetadata>httpGetEnabled|httpsGetEnabled and
    set their value to false:

```
    <behaviors>

 <serviceBehaviors>

   <behavior name="XFWebServerServiceBehavior">

     <serviceMetadata httpGetEnabled="false" httpsGetEnabled="false" />

     <serviceDebug includeExceptionDetailInFaults="false" />

     <dataContractSerializer maxItemsInObjectGraph="2147483647" />

</behavior>

   </serviceBehaviors>

  </behaviors>
```
…

7. In file explorer, navigate up one directory to:
   "C:\Program Files\OneStream Software\OneStreamWebRoot"
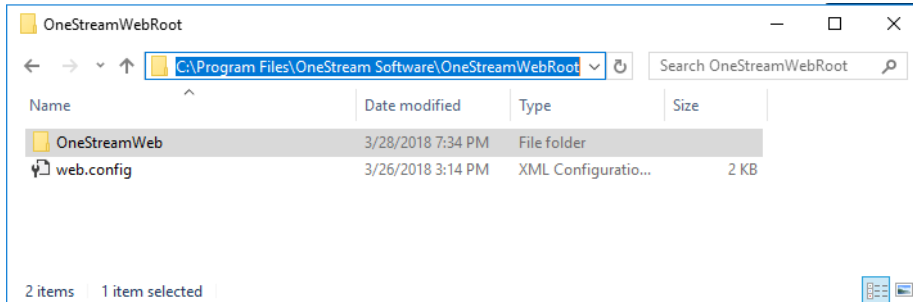
8. Right-click the Web.config file and open it in Notepad.
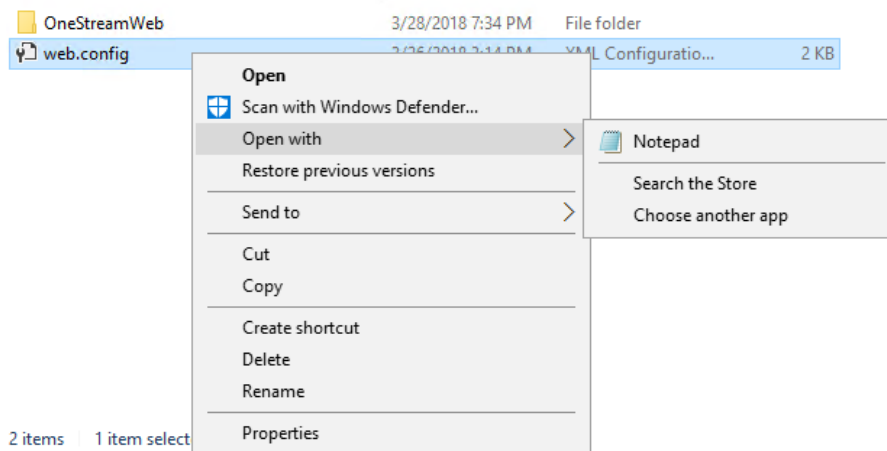
9. Place your cursor at the end of <directoryBrowse enabled="false" /> line and click **Enter**.
   Paste the following text on the newly created blank line and save the file.

```
<httpProtocol>

  <customHeaders>

    <!--Override the IE browser's Compatibility View Settings for intranet
sites.-->

    <add name="X-UA-Compatible" value="IE=Edge" />

    <add name="X-Frame-Options" value="SAMEORIGIN" />

  </customHeaders>
```

```
    </httpProtocol>

        <security>

            <requestFiltering removeServerHeader="true">

                <verbs>

                    <add verb="OPTIONS" allowed="false" />

                </verbs>

            </requestFiltering>

        </security>
```
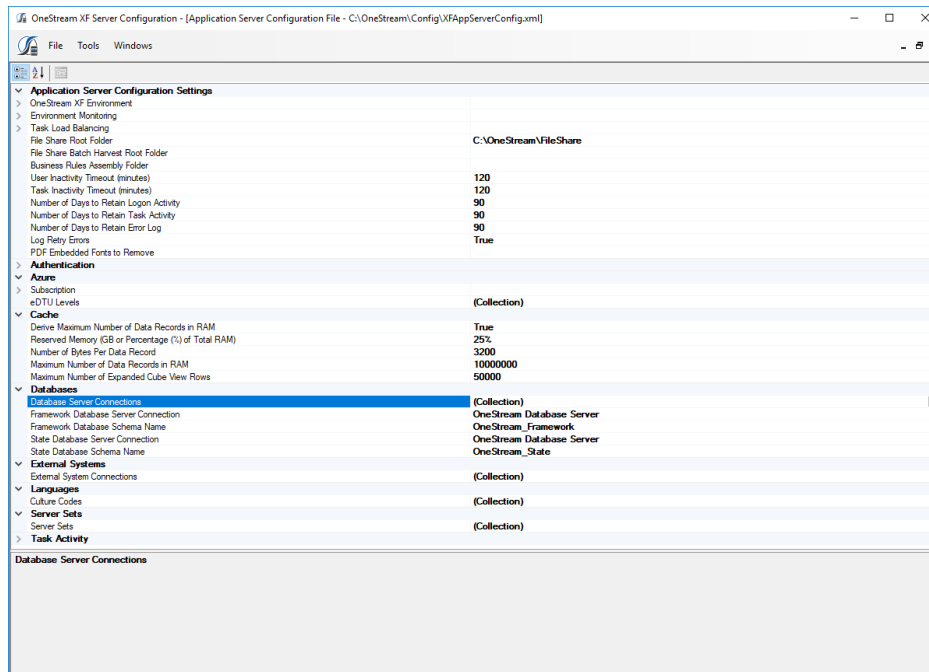
> **NOTE:** If you are running on Windows 2012 R2, you need to remove
> removeServerHeader="true" from requestFiltering. This is only supported for IIS
> 10.

Repeat steps 2-9 starting in the following Mobile Server directories as above for the Web
Server directories:
"C:\Program Files\OneStream Software\OneStreamMVCRoot\OneStreamMVC" directory
"C:\Program Files\OneStream Software\OneStreamMVCRoot\" directory

10. Restart IIS on the Web Server/Mobile Server.

11. Repeat steps 1-11 on each OneStream Web and/or Mobile Server.

Appendix 8: Web.config Proxy Settings

This section describes updating the Web.Config File if there is a proxy in the environment.

To set the default .NET Proxy Settings you will need to edit the web.config file in the OneStream Web Server Directory located in the following path:

C:\Program Files\OneStream Software\OneStreamWebRoot

1. Open the Web.Config file using a text editor

2. Add the following section to the file:

```
<system.net>

  <defaultproxy>

  <proxy proxyaddress= "http://proxyserver" bypassonlocal= "true"
usesystemdefault="false">

  </proxy>

    </defaultproxy>

  </system.net>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <system.net>
        <defaultProxy>
            <proxy proxyaddress="http://proxy.onestream.com:8080" bypassonlocal="true"  usesystemdefault="false"/>
        </defaultProxy>
    </system.net>
    <system.webServer>
        <handlers accessPolicy="Read, Script" />
        <directoryBrowse enabled="false" />
        <httpProtocol>
            <customHeaders>
            <!--Override the IE browser's Compatibility View Settings for intranet sites.-->
            <add name="X-UA-Compatible" value="IE=Edge" />
            <add name="X-Frame-Options" value="SAMEORIGIN" />
            <add name="X-Content-Type-Options" value="nosniff" />
            <add name="X-XSS-Protection" value="1" />
            <add name="Cache-Control" value="no-store,no-cache" />
            </customHeaders>
        </httpProtocol>

        <security>
            <requestFiltering>
                <verbs>
                    <add verb="OPTIONS" allowed="false" />
                </verbs>
            </requestFiltering>
        </security>
    </system.webServer>
</configuration>
```

This can also be referenced from the following Microsoft document:https://docs.microsoft.com/en-us/dotnet/framework/configure-apps/file-schema/network /defaultproxy-element-network-settings

3. Save the changes to the web.config file

4. Recycle IIS on the web server for the change to take effect

This will be required if you have a proxy server that is required to access the internet from the OneStream Web/Application Servers in the environment if you are using Azure SSO to successfully resolve the external Microsoft address for logon (Login.microsoft.net).

Appendix 9: Installing and Configuring PingFederate

This section describes in detail an example of a full PingFederate installation and configuration as OneStream's Infrastructure team experienced it in order to implement and test user authentication in PingFederate.

PingIdentity components installation and configuration

PingFederate is Ping Identity's enterprise identity bridge. PingFederate enables outbound and inbound solutions for single sign-on (SSO), federated identity management, mobile identity security, API security, and social identity integration. Browser-based SSO extends employee, customer and partner identities across domains without passwords, using only standard identity protocols (Security Assertion Markup Language—SAML, WS-Federation, WS-Trust, and OAuth).

PingFederate Installation process:

1. Create a PingIdentity developer account here.

2. Request a license key via Ping Identity licensing website.

3. Download PingIdentity's PingFederate from this site.

4. System Requirements for PingFederate installation here.

5. Download and install Java SE RunTime Environment (Server JRE) here.

6. Set the JAVA_HOME environment variable to the Server JRE installation directory path and add its bin directory to the PATH environment variable.

7. Install PingFederate by following these instructions.

8. Open PingFederate Admin Console and sign in as an Administrator account.

PingFederate and OAuth server configuration steps:

1. **Server Configuration > Server Settings > Roles and Protocols** screen. Select the **Enable OAuth 2.0 Authorization Server (AS) role** check box. Select the **OpenID Connect** check box. Select Enable Identity Provider then SAML 2.0 and WS-TRUST check boxes. Save changes (or hit Next button until Save appears).

2. Server Configuration > SSL Server certificates > Create one (make sure it's CN matches the server name to avoid certification errors when accessed from the clients).

3. Server Configuration > Trusted CAs > Import the just created cert (this same certificate will need to be installed on the client side in '**Trusted Root Certification Authorities**' store).

4. Server Configuration > Signing & Decryption Keys & Certificates > Create New (follow instructions to create a new signing certificate that will be used later to validate access tokens for Resource Owner Password flow. Ex. I created one for my dev environment with a CN that hints to its use: CN=Config Signing Cert, OU=Dev, O=Ping, L=Denver, ST=CO, C=US).

5. Server Configuration > Password Credential Validators > Create New Instance. Enter values for Instance Name (ex.'UserPass') and ID, Select Type: Simple User Name Password Validator > Next. In the Instance Configuration screen Add a new row to 'Users' > Add all your test user names and passwords (store at least one of these values because these are the user(s) that will be added in OneStream security with PingIdentity Authentication Provider Type). Hit Next until able to Save.

6. OAuth Server > Scope Management > Add scopes: address, email, openid, phone, profile > Save

7. Identity Provider > Manage IdP Adapter Instances > Create Instance: Example: Name = HTMLFormSimplePCV; ID=HTMLFormSimplePCV; Type: HTML Form IdP Adapter > Next. In IdP Adapter tab add a new row to Credential Validators, select 'UserPass' created above. Extended Contract tab: policy.action and username should be listed under Core Contract

8. Adapter Attributes tab: check Pseudonym checkbox for username> hit Next until able to Save.

9. OAuth Server > Authorization Server Settings

- select 'UserPass' for OAuth Administrative Web Services Settings / Password Credential Validator

- check "Implict", "Authorization code", "Resource Owner Password Credentials" and "Allow unidentified clients to make Resource Owner Password credentials grants" boxes

10. OAuth Server > Access Token Management > Create new (fill fields similar to below)

- Instance Name: JSON Web Tokens

- Instance ID: jwt

- Type: JSON Web Tokens

- Class Name :
  com.pingidentity.pf
  .access.token.management.plugins.JwtBearerAccessTokenManagementPlugin

- Parent Instance Name: None

- Instance Configuration

- Certificates: k1, CN=Config Signing Cert, OU=Dev, O=Ping, L=Denver, ST=CO, C=US (This is the signing certificate created in #4)

- Token Lifetime: 120

- JWS Algorithm: RSA using SHA-256

- Active Symmetric Key ID: None Selected

- Active Signing Certificate Key ID: k1

- JWE Algorithm: None Selected

- JWE Content Encryption Algorithm: None Selected

- Active Symmetric Encryption Key ID: None Selected

- Asymmetric Encryption Key

- Asymmetric Encryption JWKS URL: http://<serverName>:<port>/pf/jkws

- Include Key ID Header Parameter: TRUE

- Include X.509 Thumbprint Header Parameter: TRUE

- Default JWKS URL Cache Duration: 720

- Include JWE Key ID Header Parameter: TRUE

- Include JWE X.509 Thumbprint Header Parameter: TRUE

- Client ID Claim Name: client_id_name

- Scope Claim Name: scope

- Space Delimit Scope Values: FALSE

- Issuer Claim Value: http://<serverName>:<port>

- Audience Claim Value: OneStreamClient

- JWT ID Claim Length: 0

- Access Grant GUID Claim Name: agid

- JWKS Endpoint Path: /oauth/jwks

- JWKS Endpoint Cache Duration: 720

- Publish Key ID X.509 URL: TRUE

- Publish Thumbprint X.509 URL: TRUE

- Session Validation:

- Check Session Validation Status: FALSE

- Check Session Revocation Status: FALSE

- Update Authentication Session Activity: FALSE

- Access Token Attribute Contract:

- Attribute: OrgName

- Attribute: sub

- Attribute: Username

- Resource URIs :

- Access Control :

- Restrict Allowed Clients : FALSE

11. OAuth Server > OpenID Connect Policy Management > Create New (see example policy below)

12. OAuth Server > Resource Owner Credentials Mapping > Map 'UserPass' to Persistent Grant Contract

13. OAuth Server > Access Token Attribute Mapping > Map Default (Context) to JSON Web Tokens (Token Manager)

   - OrgName: example mapping: Source=Text, Value=Ping Federate Corporation

   - Username: Source : Persistent Grant, Value:USER_KEY

   - sub: Source : Persistent Grant, Value:USER_KEY (needed to retrieve user claims)

14. OAuth Server > IdP Adapter Mappings: Map HTMLFormSimplePCV To Persistent Grant Contract

15. Add OneStreamWeb client:

   - OAuth Server > Clients > Create New:

   - Client ID = OneStreamWeb

   - Client Name = OneStreamWeb

   - Description = Authorization Code flow for OneStreamWeb application (example)

   - Client Authentication = Client Secret > Generate Secret (store this value)

   - Redirect URIs: Add: http://<serverName>:<port>/OneStream/OneStreamXF.aspx, and http://<serverName>:<port>/OneStream/OneStreamWindowsApp.aspx

   - Bypass Authorization Approval = Check (this will be a trusted app; there is no need for an extra Authorization Approval form)

- Allowed Grant Types: Authorization Code; Implicit

- Open Id Connect: ID Token Signing Algorithm = Default

- Save

16. Add OneStreamMvc client:

- OAuth Server > Clients > Create New:

- Client ID = OneStreamMvc

- Client Name = OneStreamMvc

- Description = Authorization Code flow for OneStreamMvc application (example)

- Client Authentication = Client Secret > Generate Secret (store this value)

- Redirect URIs: Add: 'http://<serverName>:<port>/Authentication/Logon'

- Bypass Authorization Approval = Check (this will be a trusted app; there is no need for an extra Authorization Approval form)

- Allowed Grant Types: Authorization Code; Implicit

- Open Id Connect: ID Token Signing Algorithm = Default

- Save

17. Add OneStreamClient client

- OAuth Server > Clients > Create New:

- Client ID = OneStreamClient

- Client Name = OneStreamClient

- Description = PingFederate placeholder for OneStream native apps authentication

- Client Authentication: None

- Redirect URI: Add: https://[SeverName]: [SSLPortNumber]/OneStreamWeb/OnestreamLogonCallback.aspx/

- Bypass Authorization Approval = Check (this will be a trusted app; there is no need for an extra Authorization Approval form)

- Allowed Grant Types: Authorization Code; Resource Owner Password Credentials, Refresh Token

- Open Id Connect: ID Token Signing Algorithm = Default

- Save

PingFederate IWA Integration Kit V3.1

Installed and configured IWA Integration kit following documentation below:

https://docs.pingidentity.com/bundle/ix_m_
downloadDocumentation/page/IWAIK31UserGuide.pdf

Configure Supported Browsers for Kerberos and NTLM

Install and configure the Kerberos Integration Kit using these instructions:

https://ping.force.com/Support/PingFederate/Integrations/How-to-configure-supported-browsers-for-Kerberos-NTLM

PingFederate Notes

If SSL Settings > RequireSSL setting is enabled in IIS, ensure Accept Client Certificates option is selected. Typically, the exception "IDX10500: Signature validation failed. Unable to resolve SecurityKeyIdentifier: 'SecurityKeyIdentifier'" will be thrown if the certificate is not passed to the client.

Policy Management Example

Important: when creating the Policy Management mappings, ensure that both sub and name attributes map to Username (Token)

Configure a new policy similar to below:	
Manage Policy	

Policy ID	OAuthPlayground
Policy Name	OAuthPlayground
Access Token Manager	JSON Web Tokens
ID Token Lifetime	5
Include Session Identifier in ID Token	false
Include User Info in ID Token	false
Include State Hash in ID Token	false
Attribute Contract	
Attribute	sub
Attribute	name
Attribute	address.country
Attribute	address.formatted
Attribute	address.locality
Attribute	address.postal_code
Attribute	address.region
Attribute	address.street_address
Attribute	birthdate
Attribute	email
Attribute	email_verified
Attribute	family_name
Attribute	gender
Attribute	given_name
Attribute	locale
Attribute	middle_name
Attribute	name
Attribute	nickname
Attribute	phone_number
Attribute	phone_number_verified

Attribute	picture
Attribute	preferred_username
Attribute	profile
Attribute	updated_at
Attribute	website
Attribute	zoneinfo
Attribute Scopes	
Attribute Sources & User Lookup	
Data Sources	(None)
Contract Fulfillment	
sub	Username (Token)
name	Username (Token)
address.locality	Smallville (Text)
birthdate	1977-12-31 (Text)
gender	female (Text)
preferred_username	mgsample (Text)
Locale3	en_US (Text)
address.country	USA (Text)
updated_at	2011-01-03T23:58:42+0000 (Text)
address.postal_code	11223 (Text)
address.region	ME (Text)
nickname	Name (Text)
email	auser@example.com (Text)
website	https://www.pingidentity.com/ (Text)
email_verified	true (Text)
profile	https://www.pingidentity.com/products/pingfederate/ (Text)
phone_number_verified	true (Text)
given_name	Mary (Text)

middle_name	Good (Text)
picture	https://www.pingidentity.com/images/ping-logo.png (Text)
phone_number	(555) 555-5555 (Text)
address.formatted	123 Main Street, Smallville, ME USA 11223 (Text)
family_name	Sample (Text)
address.street_address	123 Main Street (Text)
Issuance Criteria	
Criterion	(None)

Appendix 10: Reserve URL for Native Application Authentication

This section describes how to configure client PCs running pre -5.0 versions of My Company Name, LLC native applications.

Disregard this section if yo are installing version 5.0 or higher.

SAML 2.0 authentication with ADFS:

1. Ensure that a localhost certificate is installed in the client's Local Machine. This certificate needs to be trusted to avoid IE warnings. Take note of the certificate's thumbprint (ex: d7a045f8xxxxxxxxb9702066b88bbecf)

2. Open Command Prompt with elevated permissions and run (ex. for port 8443):

   a. netsh http add sslcert ipport=0.0.0.0:8443 certhash= d7a045f8xxxxxxxxb9702066b88bbecf appid={C183BFDB-31C2-49AE-A3ED-BEA979A269C6}

where appid identifies OneStream Windows application.

3. If an error is returned run: netsh http delete sslcert ipport=0.0.0.0:8443 then rerun : netsh http add sslcert ipport=0.0.0.0:8443 certhash= d7a045f8xxxxxxxxb9702066b88bbecf appid={C183BFDB-31C2-49AE-A3ED-BEA979A269C6}

Non ADFS SAML 2.0 or OIDC authentication:

1. Open Command Prompt with elevated permissions and run (ex. for port 8080):

netsh http add urlacl url=http://127.0.0.1:8080/ user=Everyone

If an error is returned run first: netsh http delete urlacl url=http://127.0.0.1:8080/, followed by netsh http add urlacl url=http://127.0.0.1:8080/ user=Everyone

Appendix 11: Configure SAML 2.0 SSO with Different IdPs and OneStream as Service Provider

Identity Providers (IdPs) may adopt SAML 2.0 specification with minor differences. Some IdPs such as Okta support multiple ACS URLs, but other IdPs support only one ACS per connected application, such as SalesForce. The examples below show how to configure SAML 2.0 with Okta and SalesForce while using OneStream as the Service Provider (SP).

Okta as IdP

This section is an example of how to configure SAML 2.0 as an IdP.

> **NOTE:** We strongly recommend you use OpenID Connect, instead of SAML when using Okta.

Add SAML Enabled Apps in Okta

1. In the Applications Dashboard, click **Create New App**.

2. Select **Web Platform**.

3. Select **SAML 2.0** as the **Sign on Method**, then click **Create**.

4. Enter an application name, such as OneStreamWebSaml.

5. For Single Sign On, enter:
   `https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx`

6. In **Requestable SSO URLs** enter:

   - `https://sitename.onestreamcloud.com/onestreamweb/onestreamwin dowsapp.aspx` with Index 0

- `https://sitename.onestreamcloud.com/onestreamweb/onestreamlog oncallback.aspx/` with Index 1

8. In **Recipient URL**, enter:
   `https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx`

9. In **Destination URL**, enter:
   `https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx`

10. In **Audience Restriction**, enter:
    `https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx`

11. Enable **SAML Single Logout.**

12. Upload the public key of the SP Signing Certificate.

13. In **Attribute** enter these mappings:

Name	Name Format	Value
FirstName	Unspecified	user:firstName
LastName	Unspecified	user:lastName
Email	Unspecified	user:email

14. Click **Save** and **Finish**.

15. On the **Sign On** tab, click **View Setup Instructions** and copy the values for:

    - The identity provider SSO URL

    - The identity provider single issuer

    - Download the X.509 IdP certificate.

    - IdP Metadata

    - Create an XML file with the content from the identity provider SSO URL and save the file with an intuitive, descriptive name (e.g.OktaIdPMetadataWeb.xml) in the shared configuration folder.

16. Repeat steps 1 – 6, then 8 – 13.

For the My Company Name, LLC mobile application:

1. Repeat steps 1 - 6, then 8 - 13 in the above procedure, replacing:

   `https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx`

   with:

   `https://sitename.onestreamcloud.com/onestreammvc/authentication/logon`

2. Click **Create New App**, then select **Web Platform**.

3. Select **SAML 2.0** as the **Sign on Method**.

4. Click **Create** and enter a descriptive application name such as OneStreamMobileSaml.

5. In **Single Sign On**, enter:
   `https://sitename.onestreamcloud.com/onestreammvc/authentication/logon`

6. In **Recipient URL**, enter:
   `https://sitename.onestreamcloud.com/onestreammvc/authentication/logon`

7. In **Destination URL**, enter:
   `https://sitename.onestreamcloud.com/onestreammvc/authentication/logon`

8. In **Audience Restriction**, enter:
   `https://sitename.onestreamcloud.com/onestreammvc/authentication/logon`

9. Enable **SAML Single Logout**.

10. Upload the public key of SP's signing certificate.

11. In **Attributes** enter the following mappings:

12. Click **Save** and **Finish**.

13. Click the **Sign On** tab, then **View Setup Instructions**, and copy the values for:

    - The Identity Provider Single Sign-On URL

    - The Identity Provider Single Issuer

- Download the X.509 IdP Certificate.

- IdP Metadata

14. Create an XML file with the metadata content from step 13, and save the file using a descriptive name (OktaIdPMetadataMobile.xml) in the shared configuration folder.

15. Install the **X.509 IdP certificate** in the web server machine and note the certificate store name, location, FindBy mode and value.

16. Launch the Web Server Configuration Utility and specify this SAML content:

SAML 2.0 Identity Provider	
Multiple ACS URLs Supported.	True
ACS URL for Web Application.	https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx
ACS URL for Windows Application.	https://sitename.onestreamcloud.com/onestreamweb/onestreamwindowsapp.aspx
ACS URL for Mobile Application.	https://sitename.onestreamcloud.com:50004/onestreammvc/authentication/logon
Unique ID for Windows Application.	https://sitename.onestreamcloud.com/onestreamweb/onestreamlogoncallback.aspx
Single Sign-On URL for Web Application.	https://xxxxxxxxxxx.oktapreview.com/app/ossdev4xxxxxxxxxx0_onestreamwebsaml_1/exkeg5fzm3bIQ1AXc0h7/sso/saml
Single Sign-On URL for Native Applications.	https://xxxxxxxxxxx.oktapreview.com/app/ossdev47xxxxxxxxx0_onestreamwebsaml_1/exkeg5fzm3bIQ1AXc0h7/sso/saml
Single Sign-On URL for Mobile Application.	https://xxxxxxxxxxx.oktapreview.com/app/ossdev47xxxxxxxxx0_onestreammvcsaml_1/exkersovteU9GOnfW0h7/sso/saml
Entity ID for Web Application.	http://www.okta.com/xxxxxxxxxxxx1AXc0h7
Entity ID for Native Applications.	http://www.okta.com/xxxxxxxxxxxx1AXc0h7
Entity ID for Mobile Application.	http://www.okta.com/xxxxxxxxxxxxxxOnfW0h7
Metadata File Name for Web Application	OktaIdpMetadataWeb.xml
Metadata File Name for Mobile Application	OktaIdpMetadataMvc.xml
Metadata Content For Web Application.	<?xml version="1.0" encoding="UTF-8"?><md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" entityID
Metadata Content For Native Applications.	
Metadata Content For Mobile Application.	
Signing Certificate Store Name	Root
Signing Certificate Store Location	LocalMachine
Signing Certificate Find Mode	FindBySerialNumber
Signing Certificate Find Value	015cxxxxxxxxxxxxxad46
Logout Certificate Store Name	Root
Logout Certificate Store Location	LocalMachine
Logout Certificate Find Mode	FindBySerialNumber
Logout Certificate Find Value	30831ffdfxxxxxxxxxxx3ba2bb6afdf
User Inactivity Timeout (minutes)	12

Multiple ACS URLs Supported

True (This is IdP specific. Okta supports multiple ACS values)

ACS URL for Web Application

`https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx`

ACS URL for Windows Application

`https://sitename.onestreamcloud.com/onestreamweb/onestreamwindowsapp.aspx`

ACS URL for Mobile Application

`https://sitename.onestreamcloud.com:50004/onestreammvc/authentication/logon`

Unique ID for Windows Application:

https://sitename.onestreamcloud.com/onestreamweb/onestreamlogoncallback.aspx/

Single Sign-On URL for Web Application

Enter the value for the identity provider SSO URL.

Single Sign-On URL for Native Application

Enter the value for the identity provider SSO URL.

Single Sign-On URL for Mobile Application

Enter the value for the identity provider SSO URL.

Entity ID for Web Application

Enter the value for the identity provider single issuer.

Entity ID for Mobile Application

Enter the value for the identity provider single issuer.

Metadata File Name for Web Application

Enter the name of the XML file containing the metadata that you saved to the configuration folder (OktaIdPMetadataWeb.xml).

Metadata File Name for Mobile Application

Enter the name of the file that contains the IdP metadata.

17. Install the public key of the Okta Signing certificate on the web server, entering values to identify how to extract the identity provider's certificate.

Example

- Identity Provider's Certificate Store Name: Root

- Identity Provider's Certificate Store Location: LocalMachine

- Identity Provider's Certificate Find Mode: FindBySerialNumber

- Identity Provider's Certificate Find Value: xxxxxxxxxxxxxxxxxxx

1. Create and install in the web server and get a private certificate.

2. Upload the public Okta key and enter the respective values for the single logout certificate section:

    *   Logout Certificate Store Name: My (Personal)

    *   Logout Certificate Store Location: LocalMachine

    *   Logout Certificate Find Mode: FindBySerialNumber

    *   Logout Certificate Find Value: xxxxxxxxxxxxxxxxxxx

3. Open **OktaIdpMetadataWeb.xml**, copy the content and paste it into **Metadata Content For Web Application**.

4. Save and close.

5. Open **XFAppServerConfig.xml**.

6. Create a SAML authentication SSO provider section.

7. Specify the identity provider single issuer as the External Provider SSO key.

8. Save and close.

Add SAML Enabled Connected Apps in SalesForce Classic / Lightning Experience (SF)

In SF dashboard follow tutorials on how to create connected apps. Create one connected app for each OneStream application type:

1. For application type: OneStreamWeb create a SAML enabled connected app with values below:

    Entity ID: https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx

    ACS URL: https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx

    Issuer: https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx

Download Metadata and copy the file in the shared config folder

Download IdP Certificate

2. For application type: OneStreamNative create a SAML enabled connected app with values below:

   Entity ID:
   https://sitename.onestreamcloud.com/onestreamweb/onestreamlogoncallback.aspx

   ACS URL:
   https://sitename.onestreamcloud.com/onestreamweb/onestreamlogoncallback.aspx

   Issuer:
   https://sitename.onestreamcloud.com/onestreamweb/onestreamlogoncallback.aspx

   Download Metadata, copy the content somewhere as it will be needed later in this process.

3. For application type: OneStreamMobile create a SAML enabled connected app with values below:

   Entity ID:
   https://sitename.onestreamcloud.com:50004/onestreammvc/authentication/logon

   ACS URL:
   https://sitename.onestreamcloud.com:50004/onestreammvc/authentication/logon

   Issuer: https://sitename.onestreamcloud.com:50004/onestreammvc/authentication/logon

   Download Metadata and copy the file in the shared config folder

4. Install X.509 IdP Certificate from #1 in the web server machine and take note of the certificate store name, location, FindBy mode and value

5. Open Web Server Configuration Utility and complete SAML specific fields like the image below:

Multiple ACS URLs Supported.	**False**
ACS URL for Web Application.	https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx
ACS URL for Windows Application.	https://sitename.onestreamcloud.com/onestreamweb/onestreamwindowsapp.aspx
ACS URL for Mobile Application.	https://sitename.onestreamcloud.com:50004/onestreammvc/authentication/logon
Unique ID for Windows Application.	https://sitename.onestreamcloud.com/onestreamweb/onestreamlogoncallback.aspx
Single Sign-On URL for Web Application.	https://accountname.my.salesforce.com/idp/endpoint/HttpPost
Single Sign-On URL for Native Applications.	https://accountname.my.salesforce.com/idp/endpoint/HttpRedirect
Single Sign-On URL for Mobile Application.	https://accountname.my.salesforce.com/idp/endpoint/HttpRedirect
Entity ID for Web Application.	https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx
Entity ID for Native Applications.	https://sitename.onestreamcloud.com/onestreamweb/onestreamlogoncallback.aspx
Entity ID for Mobile Application.	https://sitename.onestreamcloud.com:50004/onestreammvc/authentication/logon
Metadata File Name for Web Application	SAMLIdP-WebApp.xml
Metadata File Name for Mobile Application	SAMLIdP-MobileApp.xml
Metadata Content For Web Application.	<?xml version="1.0" encoding="UTF-8"?><md:EntityDescriptor xmlns:md="urn:oasis:n
Metadata Content For Native Applications.	<?xml version="1.0" encoding="UTF-8"?><md:EntityDescriptor xmlns:md="urn:oasis:n
Metadata Content For Mobile Application.	<?xml version="1.0" encoding="UTF-8"?><md:EntityDescriptor xmlns:md="urn:oasis:n
Signing Certificate Store Name	My
Signing Certificate Store Location	LocalMachine
Signing Certificate Find Mode	FindBySerialNumber
Signing Certificate Find Value	0xxxxxxxxxxxxxxxxxxxxx052ac3eca
Logout Certificate Store Name	My
Logout Certificate Store Location	LocalMachine
Logout Certificate Find Mode	FindBySerialNumber
Logout Certificate Find Value	430xxxxxxxxxxxxxxxa3249d000000000012

Multiple ACS URLs Supported

False (This is IdP specific. SF supports one ACS value per connected app.)

ACS URL for Web Application

https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx

ACS URL for Windows Application

https://sitename.onestreamcloud.com/onestreamweb/onestreamwindowsapp.aspx

ACS URL for Mobile Application

https://sitename.onestreamcloud.com:50004/onestreammvc/authentication/logon

Unique ID for Windows Application

https://sitename.onestreamcloud.com/onestreamweb/onestreamlogoncallback.aspx/

Single Sign-On URL for Web Application

Identity Provider's Entity ID for Web Application (For example,
https://accountname.my.salesforce.com/idp/endpoint/HttpPost). This is the value of the attribute
'EntityDescriptor/entityID' in Identity Provider's metadata for the Web application.

Single Sign-On URL for Native Application

Identity Provider's Entity ID for Web Application (For example,
https://accountname.my.salesforce.com/idp/endpoint/HttpPost). This is the value of the attribute
'EntityDescriptor/entityID' in Identity Provider's metadata for the Native application.

Single Sign-On URL for Mobile Application

Identity Provider's Entity ID for Web Application (For example, https://accountname.my.salesforce.com/idp/endpoint/HttpPost). This is the value of the attribute 'EntityDescriptor/entityID' in Identity Provider's metadata for the Mobile application.

Entity ID for Web Application

https://sitename.onestreamcloud.com/onestreamweb/onestreamxf.aspx

Entity ID for Native Application

https://sitename.onestreamcloud.com/onestreamweb/onestreamlogoncallback.aspx

Entity ID for Mobile Application

https://sitename.onestreamcloud.com:50004/onestreammvc/authentication/logon

Metadata File Name for Web Application

Enter the name of the IdP Metadata file downloaded in #1

Metadata File Name for Mobile Application

Enter the name of the IdP Metadata file downloaded in #3

After installing the public key of SL Signing certificate on the web server enter values indicating how to programmatically extract the installed certificate for Identity Provider's Certificate.

Example

Identity Provider's Certificate Store Name: Root

Identity Provider's Certificate Store Location: LocalMachine

Identity Provider's Certificate Find Mode: FindBySerialNumber

Identity Provider's Certificate Find Value: xxxxxxxxxxxxxxxxxxx

Logout Certificate Store Name: My (Personal)

Logout Certificate Store Location: LocalMachine

Logout Certificate Find Mode: FindBySerialNumber

Logout Certificate Find Value: xxxxxxxxxxxxxxxxxxx

Metadata Content For Web Application:

Copy the content of the Metadata file from #1 and paste it here

Save and close

Metadata Content For Native Application

Copy the content of the Metadata file from #2 and paste it here

Save and close

Metadata Content For Mobile Application

Copy the content of the Metadata file from #3 and paste it here

Save and close

Open XFAppServerConfig.xml file

Create a SAML authentication SSO provider section

Enter the value from step 'Entity ID for Web Application' for External Provider SSO Key field

Save and close

> **NOTE:** Grant Permission to use Signing Certificate Private Key for non-admin IIS User

In the Certificate Management Console, right-click the Signing Certificate, select All Tasks, then Manage Private Keys

In the new window, click Add, which opens the Select Users, Computers, Service Accounts, or Groups window

Type Authenticated Users in the Object Names field and click Check Names. Click OK.
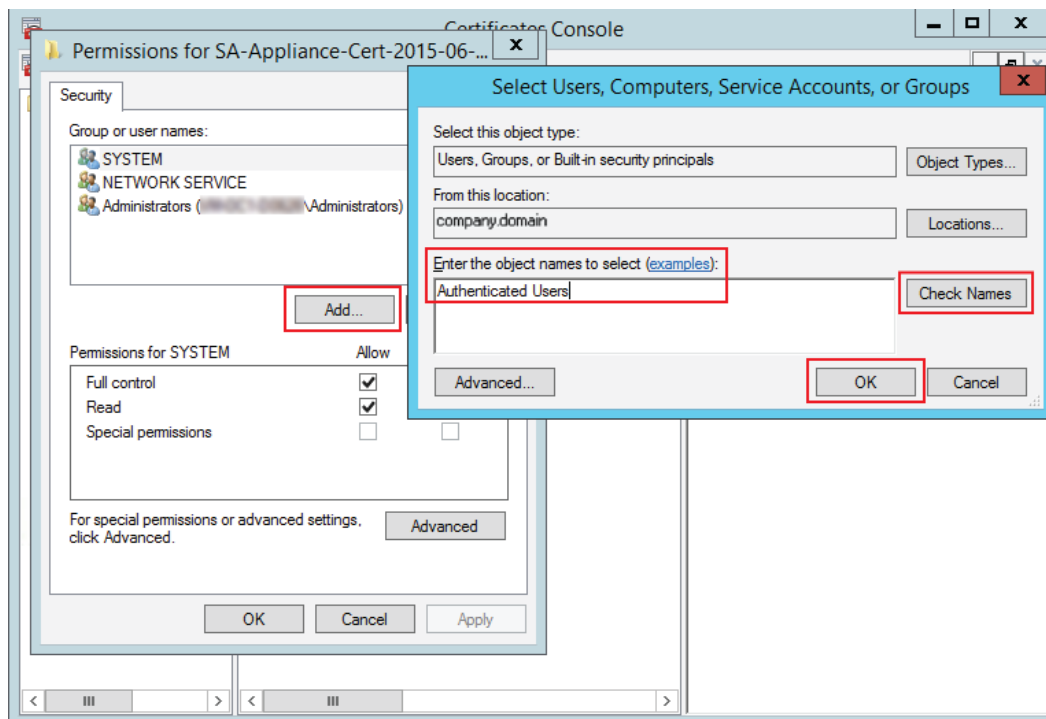
Appendix 12: Context Option Values To Use With Active Directory + SSL

Specifies the options that are used for binding to the server. The application can set one or multiple options. This is a list of possible values that can be used along with the description:

Negotiate

The client is authenticated by using either Kerberos or NTLM. When the user name and password are not provided, the Account Management API binds to the object by using the security context of the calling thread, which is either the security context of the user account under which the application is running or of the client user account that the calling thread represents.

Sealing

The data is encrypted by using Kerberos. This flag can only be used with the Negotiate context option and is not available with the simple bind option.

Secure Socket Layer

The channel is encrypted by using the Secure Sockets Layer (SSL). Active Directory requires that the Certificate Services be installed to support SSL.

Server Bind

Specify this flag when you use the domain context type if the application is binding to a specific server name.

Signing

The integrity of the data is verified. This flag can only be used with the Negotiate context option and is not available with the simple bind option.

Simple Bind

The client is authenticated by using the Basic authentication.

> **CAUTION:** Communications may be sent over the Internet in clear text if the Secure Sockets Layer option is not specified with simple bind.

When no context options are specified the default values are Negotiate, Signing, Sealing.

Appendix 13: Configure OneStream API for External Authentication

To secure OneStream API with OAuth 2.0, configure authentication with these supported external providers:

- Azure

- Okta

- PingFederate

Azure Configuration

Create an application registration in Azure

- Sign in to the Azure portal.

- In the left navigation pane, select the **Azure Active Directory** service, and then select **App registrations > New registration**.

- When the **Register an application** page appears, enter your application's registration name. Click Register.

- In application's **Overview** tab, note {Client Id}, {Tenant Id}

- In **Authentication** tab, Advanced settings, check boxes for Access and ID tokens. In Default Client Type, select Yes for Treat application as a public client. In Supported Account types, select **Accounts in this organizational directory only (Default Directory)**. Save.

- In **Certificates & secrets**, add **New client secret** and note the value. Save.

- In **Expose an API tab**, add a custom scope needed for user-machine use case. Note the scope name and the {AppId Uri} values. Save.

- Our implementation supports v2.0 Azure endpoints, so in Manifest tab, find accessTokenAcceptedVersion. Set value to 2 if different. Save.
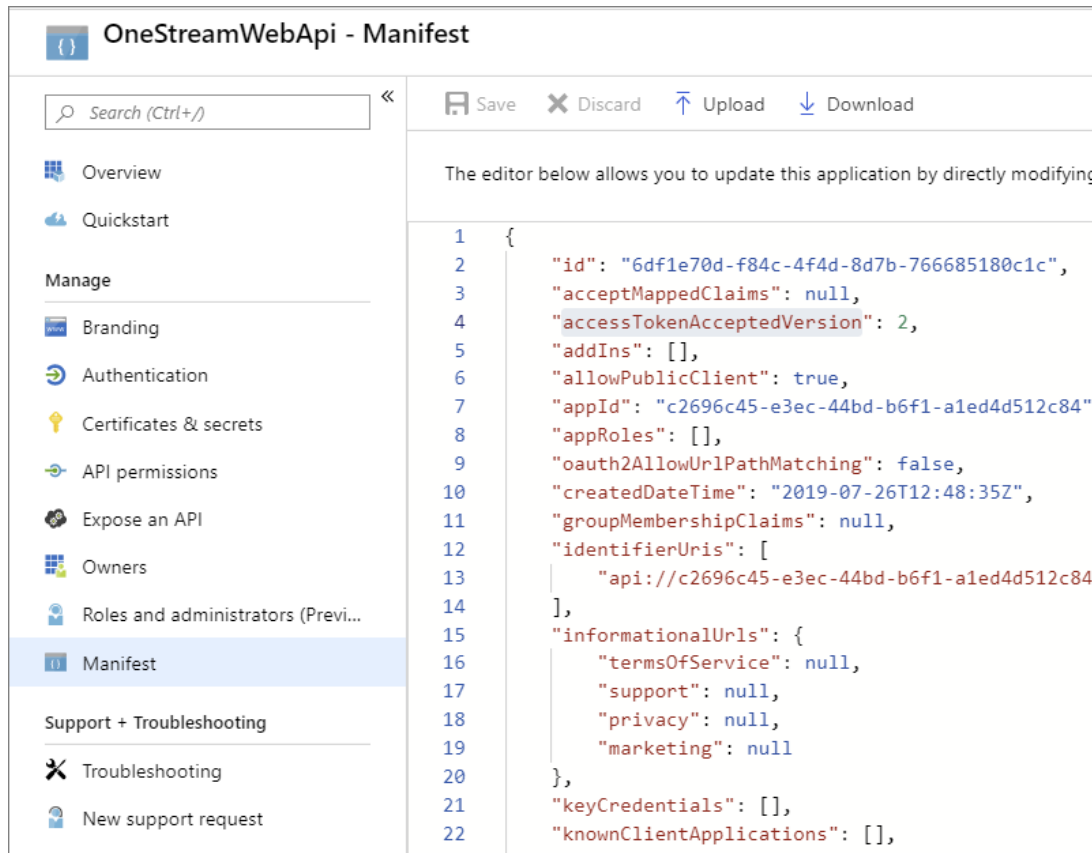
Setup Postman access token requests

- Create a new POST request. Set url to https://login.microsoftonline.com/{Tenant Id}/oauth2/v2.0/token with tenantid value from #4 above

- In Authorization tab, select Basic Auth for type. In Username and Password fields enter respectively ClientId and Client secret from the app registration section above

- In Headers tab, enter the following keys:

1. Accept:application/json

2. Authorization:Basic

3. Content-Type:application/x-www-form-urlencoded

- In Body, enter either:

    1. grant_type:client_credentials

    2. scope:{AppId Uri}/.default for machine to machine use case or

    3. grant_type:password

    4. username:{Azure ad user name}

    5. password:Azure ad user password}

    6. scope:{custom scope}

- Click Send and notice the value of access_token in the response

Update the Server Config Utility

Go to the Server config utility and enter the values for the four Web Api properties.

Create a U2M Application Registration in Okta

1. Create a Native Application and specify a label.

2. On the General tab, select all the options for **Allowed Grant Types**.

3. Copy the values for **Logout Redirect URIs**, **Client ID**, and **Client Secret**.

4. Select **Use Client Authentication**.

5. Click **API** > **Authorization servers**.

6. Click **Add Authorization Server**.
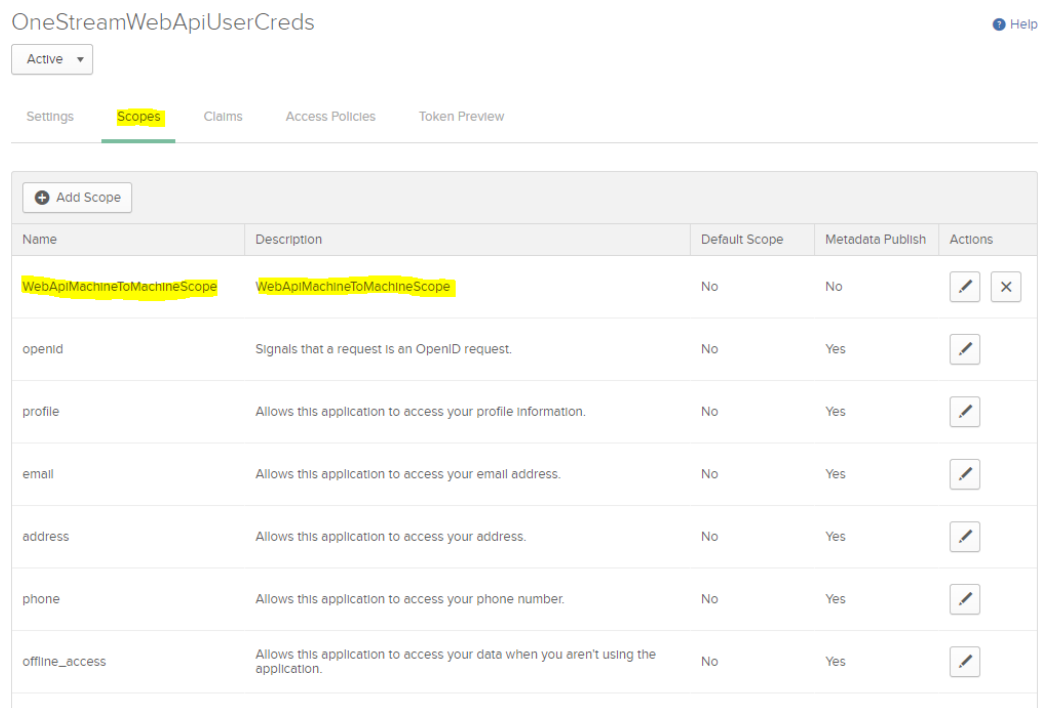
7. Enter a name, specify **Client ID** in **Audience**, and click **Save**.

8. Click **Add Scope** to define a scope needed for the Machine-to-Machine scenario. For example:

9. Do not change the other scopes.

10. Create an M2M Application Registration
    (grant_type = client_credentials)

11. Create a new OAuth Service App.

12. Specify a name, then click **OK**.

13. Copy the values for Client ID and Client Secret.

← Back to Applications

OneStreamWebApiClientCredentials

Active ▾   🔒   View Logs

General

General Settings   Edit

APPLICATION

Application label   OneStreamWebApiClientCredentials

Client Credentials   Edit

Client ID   Ooaml220rqTFFZkca0h7   🗎

Public identifier for the client that is required for all OAuth flows.

Client secret   •••••••••••••••••••••••••••••••••   👁 🗎

14. Launch the Server Config utility and enter the values for the four Web Api properties:
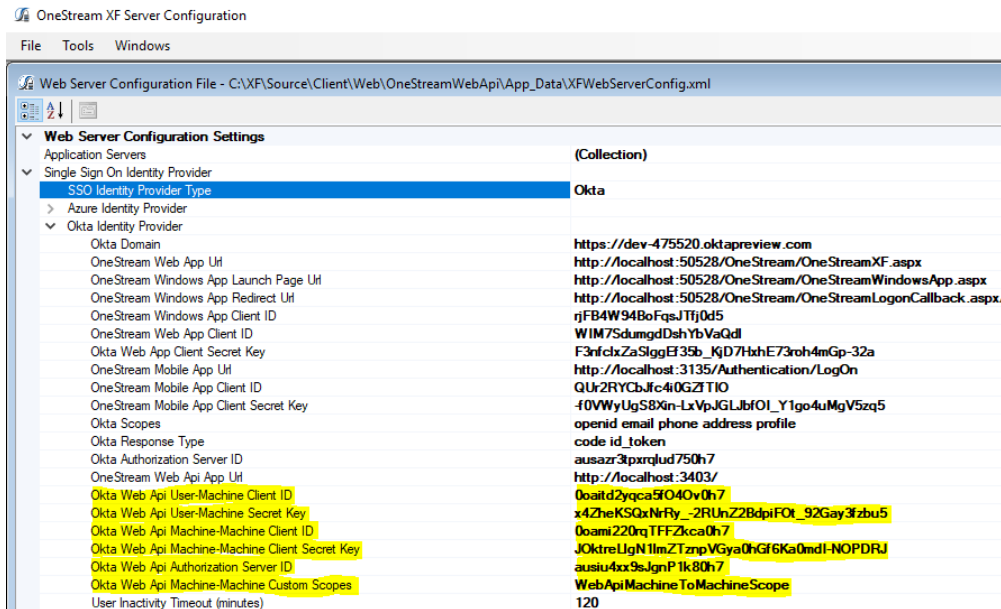
15. Save and restart IIS.

16. For the PingFederate configuration: Add a client in the Admin Console to represent the OneStreamWeb Api application.

17. In Okta, create an U2M Application Registration (grant_type = password).

18. Create an Access Token Manager (ATM) copy the **Access Token Manager ID** or client credentials. See Appendix 9.10 for setting up an ATM.

19. Set **Client Authentication** to **Client Secret**.

20. Click **Generate Secret**, then **Update**.

21. Copy the values of **Client ID** and **Client Secret**.

22. In **Allowed Grant Types**, select **Authorization Code, Resource Owner Password Credentials, Client Credentials.**

23. Set **Default Access Token Manager** to the value you copied.

24. Save.

25. Launch the Server Config utility and enter the values for the four Web Api properties.

26. Save and restart IIS.

Appendix 14: OneStream Api Endpoints

This Api implementation is client agnostic, so every Api test capable third-party tool can be pointed to OneStreamWeb Api endpoints. For the purpose of this walkthrough, Postman is used.

- **Versioning**: This implementation will start with api-version=5.2.0

- **HelpPage endpoint** url: browse to http(s)://[serverName]:[port]/OneStreamApi. This is an interactive help page that lists details about the endpoints, the argument lists, expected return values and potential error messages.

- **Authentication Execute LogonAndReturnCookie endpoint.** Returns a one-time cookie value that indicates authentication state. Used mostly by enablement team to verify the installation of OneStream Web Api completed successfully.

  1. Create new POST request in Postman.

  2. Url = http(s)://[serverName]:[port]/api/Authentication/LogonAndReturnCookie?api-version=5.2.0

  3. Authorization: Type=Bearer Token. Token={{webapi_access_token}}

  4. Headers: Content-Type=application/json

     ○ Body (raw / jSON):

     ○ Arguments:
     BaseWebServerUrl - The URL for the Web Service - string - Required
     ApplicationName - The name of the Application attempted to access - string - Required
     <response code="200">Returns a JSON representation of the resulting DataSet.</response>
     <response code="400">Bad Request. Missing Authentication arguments.</response>
     <response code="500">Error Message. Authentication Failed. Please check the Error Log for more details</response>

◦ Click Send and observe the response at the bottom pane. If successful, an one-time cookie value that indicates authentication state will be returned. Otherwise the error message will be shown. More details will be logged in the Error and Activity logs.

Appendix 15: Alternative Methods for Running Windows on a Mac®

This appendix outlines the My Company Name, LLC requirements for accessing the platform on a Mac®. We recommend that only the information technology professionals responsible for installing, maintaining and supporting OneStream use these methods.

These methods allow running a Windows Operating System on a Mac®:

- Parallels Desktop Pro Edition that supports:

    ○ Windows 8.1 and 10

    ○ Up to 128 GB vRam for each VM

    ○ Up to 32 vCPU for each VM

- Virtual Desktop

    ○ Allows a user to run a Virtual Desktop with Windows installed locally on the Mac.

    ○ User can launch My Company Name, LLC in the Virtual Desktop to perform daily activities.

- Boot Camp Assistant that allows Windows 10 to be installed on a Mac and switching between MacOS and Windows when restarting. See:https://support.apple.com/en-us/HT201468

You would then use the Windows Operating System to access the My Company Name, LLCWindows application and client utilities. See the System Requirements and Architecture Guide for information about accessing OneStream from the Client Interface.