OneStream®

Installation and Configuration Guide

8.1.3 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.

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.

Use the checklist to prepare your environment before making an installation appointment.

- 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.

- Enabled a High Performance Power Plan.

- Microsoft .NET 6

    ○ Latest version of .NET Desktop Runtime x64 (minimum 6.0.14).

    ○ Latest version of ASP.NET Core Runtime "Hosting Bundle"(minimum 6.0.14).

- Configured load balancing.

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

- A supported version of Internet Information Server (IIS).

- Discussed and planned for the size of each Web Server and implement the CPU and Memory minimums.

- 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 portal on the Solution Exchange 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).

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**.
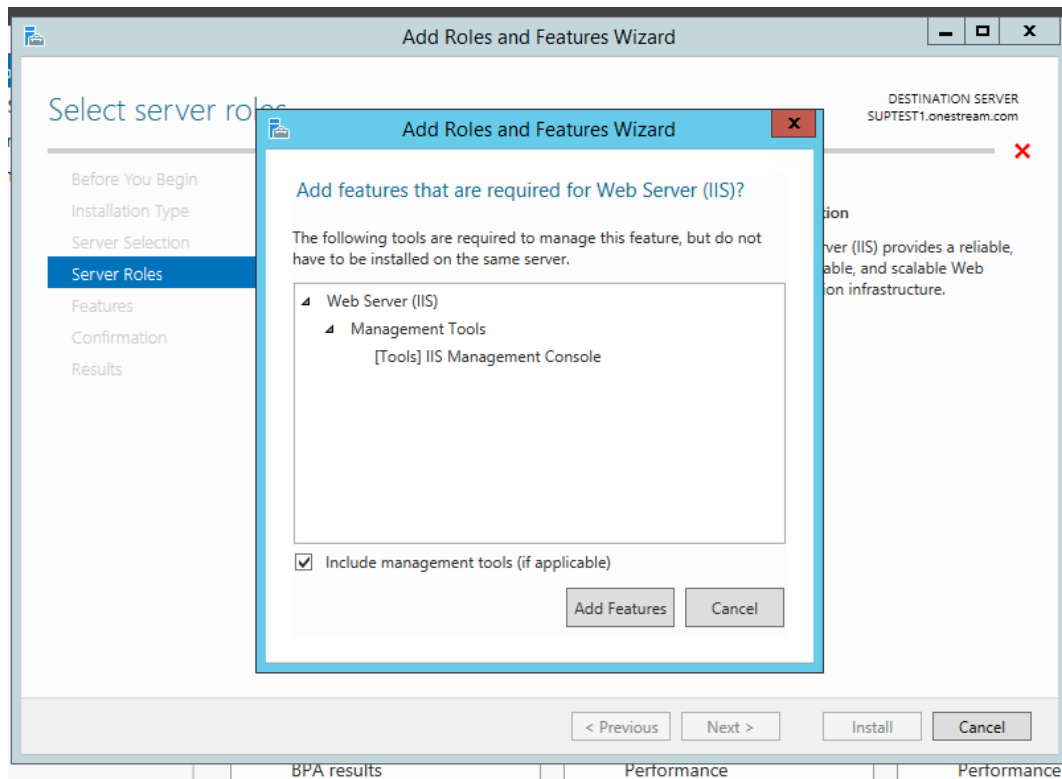
- Click **Next** and select all **.NET Framework 6.0** (or greater) features including **WCF Services** Features as shown.
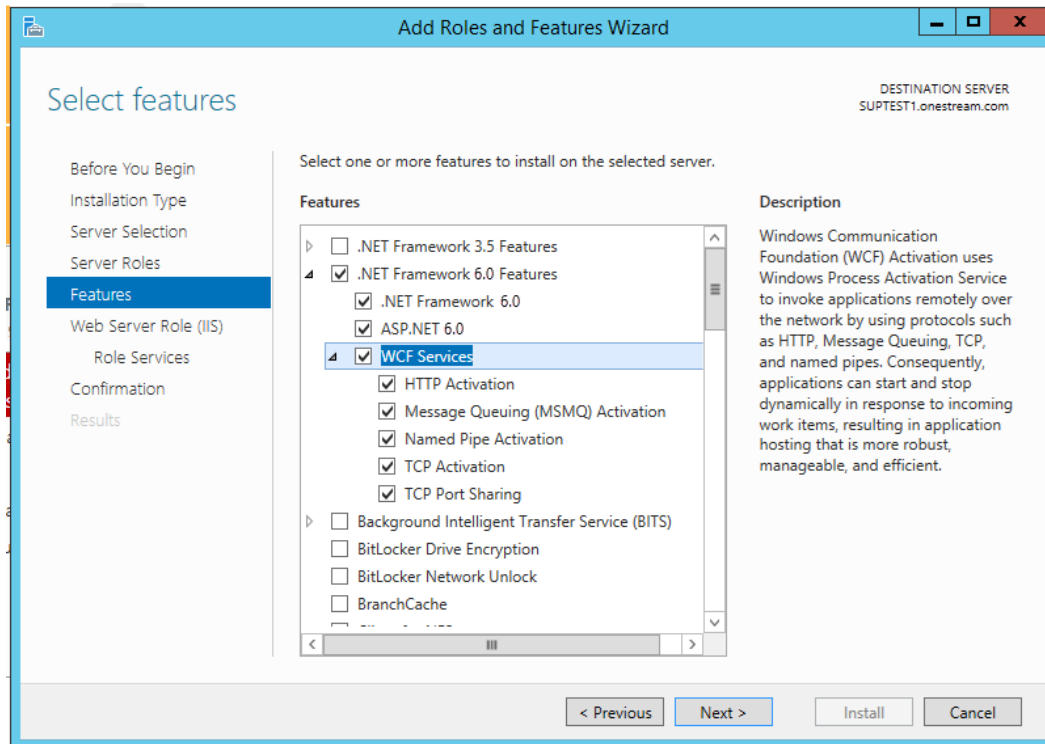
- Select all options under **Windows Process Activation Services**.
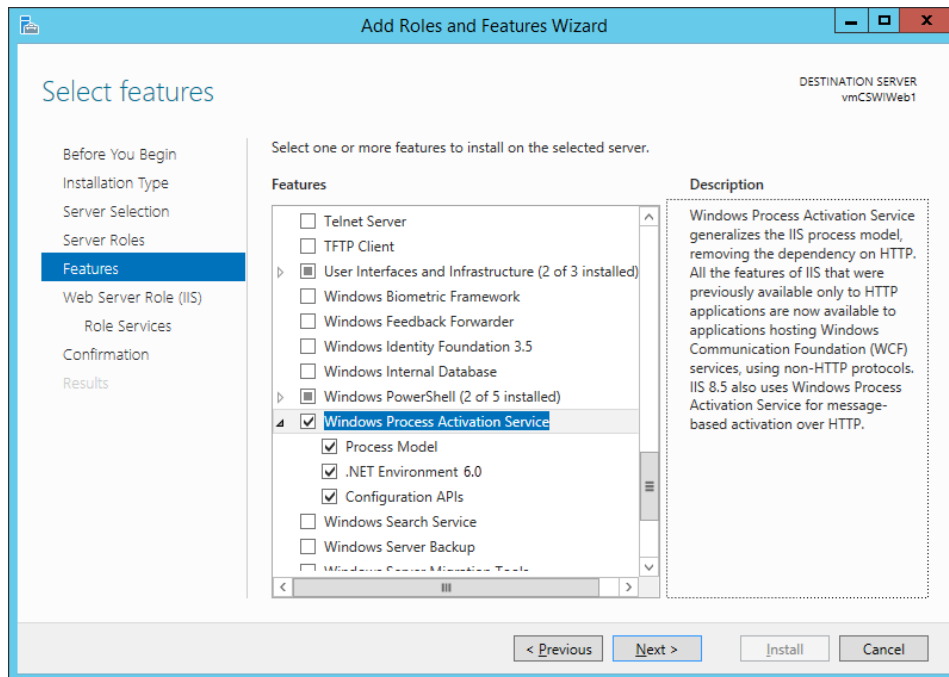
- Click **Next** and select these role services:

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

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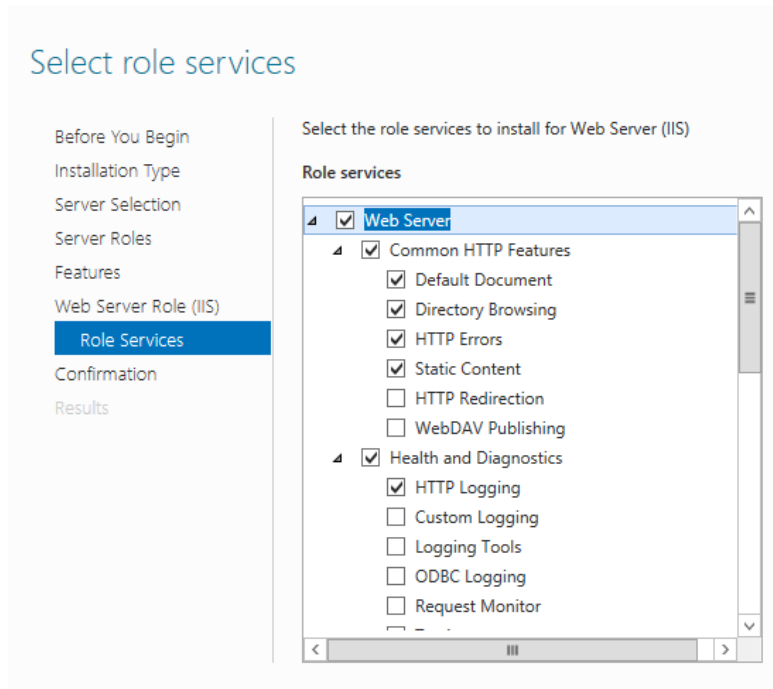
- **Performance: Static Content Compression**

  ○ **Security: Request Filtering**

  ○ **ASP.NET 6.0** (or greater)

  ○ **ISAPI Extensions**

  ○ **ISAPI Filters**

  ○ **Management Tools: IIS Management Consol**e

- 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.

- Install the .NET Framework.

- 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.

- 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:

Select server roles

Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results

Select one or more roles to install on the selected server.

Roles

◢ ■ Performance (1 of 2 installed)
   ☑ Static Content Compression (Installed)
   ☐ Dynamic Content Compression
◢ ■ Security (1 of 9 installed)
   ☑ Request Filtering (Installed)
   ☐ Basic Authentication
   ☐ Centralized SSL Certificate Support
   ☐ Client Certificate Mapping Authenticatic
   ☐ Digest Authentication
   ☐ IIS Client Certificate Mapping Authentic
   ☐ IP and Domain Restrictions
   ☐ URL Authorization
   ☐ Windows Authentication
▷ ■ Application Development (6 of 11 installed)
  ☐ FTP Server

Add Roles and Features Wizard

Select role services

Before You Begin
Installation Type
Server Selection
Server Roles
Features
Web Server Role (IIS)
Role Services
Confirmation
Results

Select the role services to install for Web Server (IIS)

Role services

◢ ☑ Application Development
   ☐ .NET Extensibility 3.5
   ☑ .NET Extensibility 6.0
   ☑ Application Initialization
   ☐ ASP
   ☐ ASP.NET 3.5
   ☑ ASP.NET 6.0
   ☐ CGI
   ☑ ISAPI Extensions
   ☑ ISAPI Filters
   ☐ Server Side Includes
   ☐ WebSocket Protocol
◢ ☐ FTP Server
   ☐ FTP Service

Description
Applicat...
expensiv...
initializa...
web pag...

< Previous   Next >

- Enable Application Initialization role on the Application Servers.

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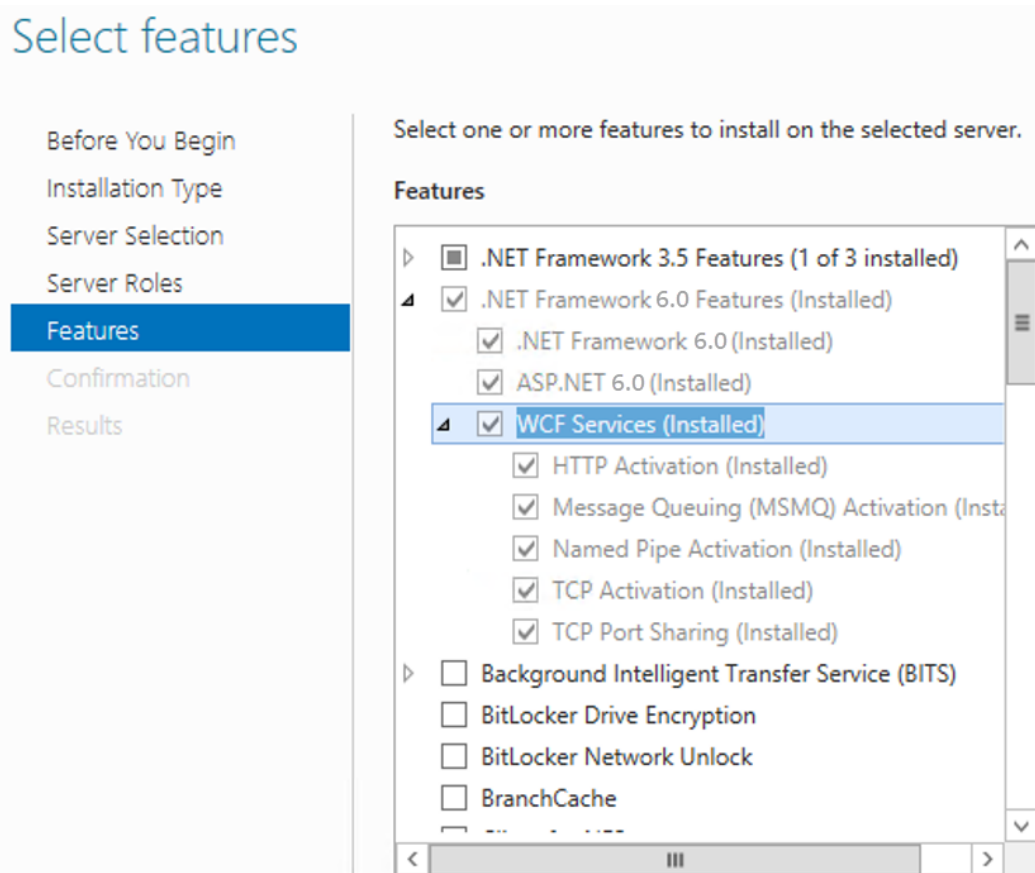
Database Server Requirements and Considerations

- Windows OS version installed with all updates.

- Proper version of 64-bit Microsoft SQL Server installed with all updates applied. SQL Server Enterprise Edition or versions beyond SQL Server Standard 2017 are recommended for larger, more complicated deployments due to support for table partitioning.

    - 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.

- 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 Applications 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 accounts 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 folders:

    - 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/Write 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

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 OneStream Servers.

- The OneStream 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.

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)

Client API

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 2017 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

Customers in a self-hosted environment can configure an environment to use native authentication, one external identity provider, or both native authentication and one external identity provider.

The following external identity providers are supported:

- Microsoft Active Directory (MSAD)

- Lightweight Directory Access Protocol (LDAP)

- Three OpenID Connect (OIDC) identity providers:

  - Azure Active Directory (Azure AD [Microsoft Entra ID])

  - Okta

  - PingFederate

- SAML 2.0 identity providers (for example, Okta, PingFederate, Active Directory Federation Services [ADFS], and Salesforce)

Customers in a OneStream-hosted environment can use OneStream IdentityServer for authentication. OneStream IdentityServer supports combinations of most OIDC compliant and SAML compliant external identity providers (IdPs) or native authentication coupled with external IdPs. This enhances user authentication by supporting multiple providers in one environment. See the *Identity and Access Management Guide*for information about authentication with OneStream IdentityServer.

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 Solution Exchange.

OneStream Servers Installation Package

- Application Server

- Web Server

- Server Configuration Utility

- Database Configuration Utility

- Upgrade Assistant Utility

OneStream Windows App Client Installation Package

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

OneStream Excel Add-In Client Installation Package

OneStream Excel Add-In

Desktop Application Side-by-Side Installer

You can install the OneStream Desktop application in a side-by-side manner.

Use the EXE installer to manually install the desktop application on your own computer, or if more than one version of the desktop application needs to be installed on the same system. You do not need to be a local administrator to perform a per-user installation, but you must be a local administrator to perform per-machine installations.

Prerequisites

To install the desktop application, you must have the OneStream Desktop EXE installer file. This file is located on the OneStream MarketPlace where each of the platform installer packages are located.

> **NOTE:** Only administrators with access to the MarketPlace can download the package and distribute to their users.

To download the desktop application package:

1. Log into the OneStream MarketPlace.

2. Go to Platform.

3. Select a platform version.

4. Click **Download** for either the Client Software or the On-Premise Full Package.

5. Unzip the files to the desired location.

Hardware and Software Requirements

Hardware and Software	Requirements
Supported Operating Systems	• Windows 10 • Windows 11
Web Browser	• Chrome or Microsoft Edge (needed only if deploying OneStream Desktop to end-users through the web)
Recommended Hardware	• Exceed the minimum requirements for Operating System, and browser. • 64-bit Architecture • 8 GB RAM or higher
Required Software	• Microsoft .NET 6.0 • Microsoft Office 2019 64-bit version or above • Microsoft Edge WebView2 Runtime Control (for features that embed external web content inside the OneStream Desktop application) *included with Windows 11+ and Office 2019. Install separately if using older Windows and Office versions.
Recommended Software	• 64-bit Windows OS • 64-bit Microsoft Office Excel version 2019 or higher (for optional Excel Add-in)

Single Instance Installation

To install a single instance of the OneStream Desktop application:

1. Double-click the **OneStream Desktop EXE** installer file to launch the wizard, then click **Next**.

2. Accept the terms of the license agreement, and then click **Next**.

3. Change the folder path if needed, and then click **Next**.

4. Click **Install**.

You can now log in and use the application.

Multiple Instance Installation

The most common scenario for needing more than one version of OneStream Desktop installed on the same computer is that your company has setup a test environment with a newer version in order to perform testing prior to upgrading production servers. To test the new version and maintain access to the current version, you might need to install an additional instance of the Desktop application on your computer.

You can install up to eight instances of the desktop application on a single computer. Each installation instance has a specific named instance 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.

> **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, in which case the installation attempt fails.

To install additional desktop application instances:

1. Double-click the **OneStream Desktop EXE** installer file to launch the wizard, and then click **Next**.

2. Select one of the following:

- **Install a new side side by side instance for just me**: installs the application for the current user.

- **Install a new side side by side instance for all users**: installs the application for all users on the computer.

3.
   NOTE: You must be a local administrator to see both options. Otherwise, you can only install a single instance.

4. Click **Next**.

5. Click **Next** to begin the installation of the new instance. The instance number is indicated in the title.

6. Accept the terms of the license agreement, then click **Next**.

7. Change the folder path if needed, then click **Next**.

8. Click **Install**.

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

You can now log in and use the application.

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 Servers](#)

6. [Configuring Web Servers](#)

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.
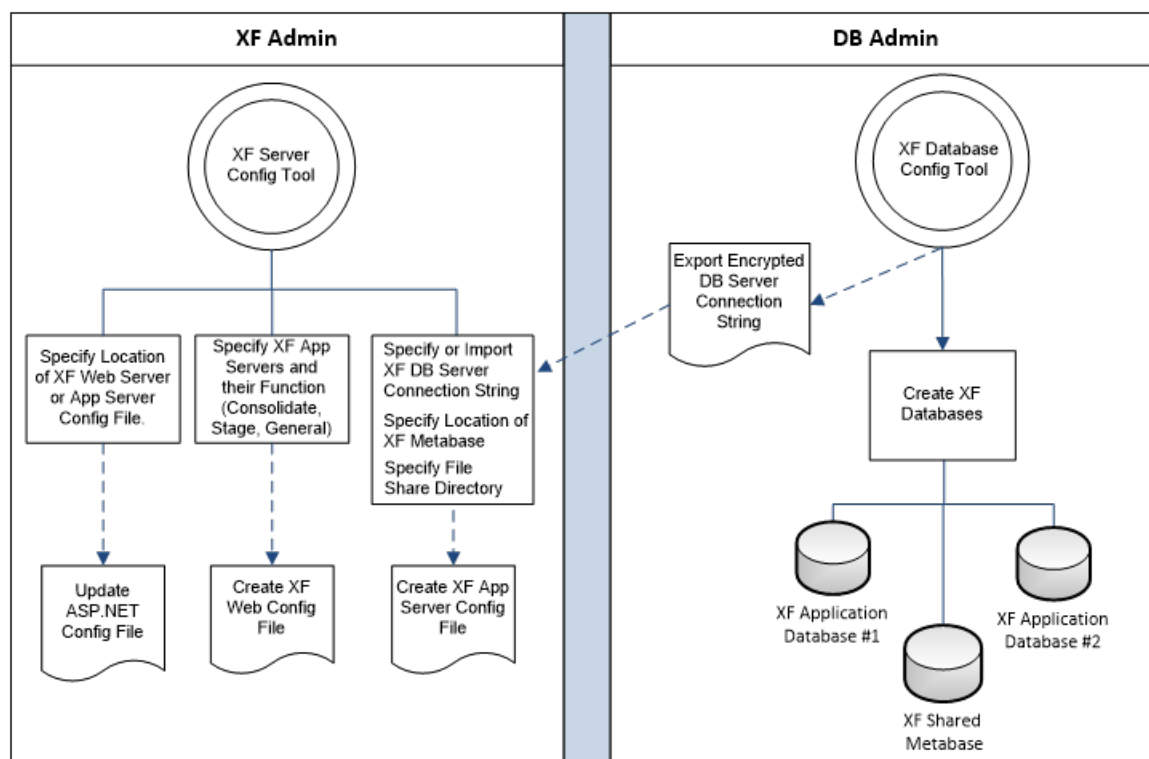
XFWebServerConfig.xml

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 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.

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:** When upgrading, Database Schema updates may 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).

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 2017 and higher. A new application database will need to be created to utilize Azure in the future.
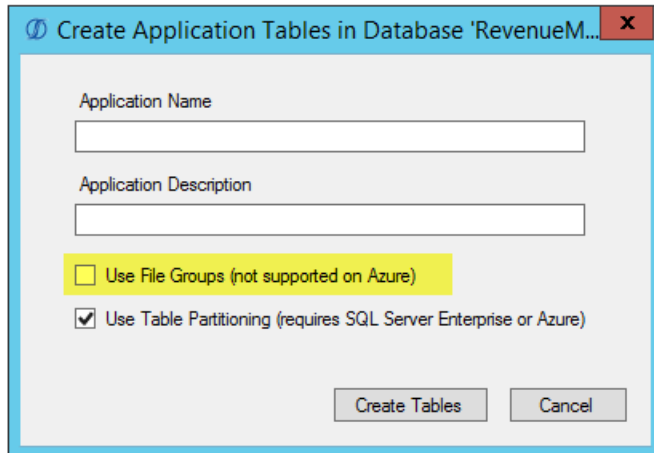
Microsoft does offer a utility to migrate from a classic SQL Enterprise database using File Groups to a non-File Group. The information can be found here: https://azure.microsoft.com/en-us/blog/migration-cookbook-now-available-for-the-latest-azure-sql-database-update-v12/

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:
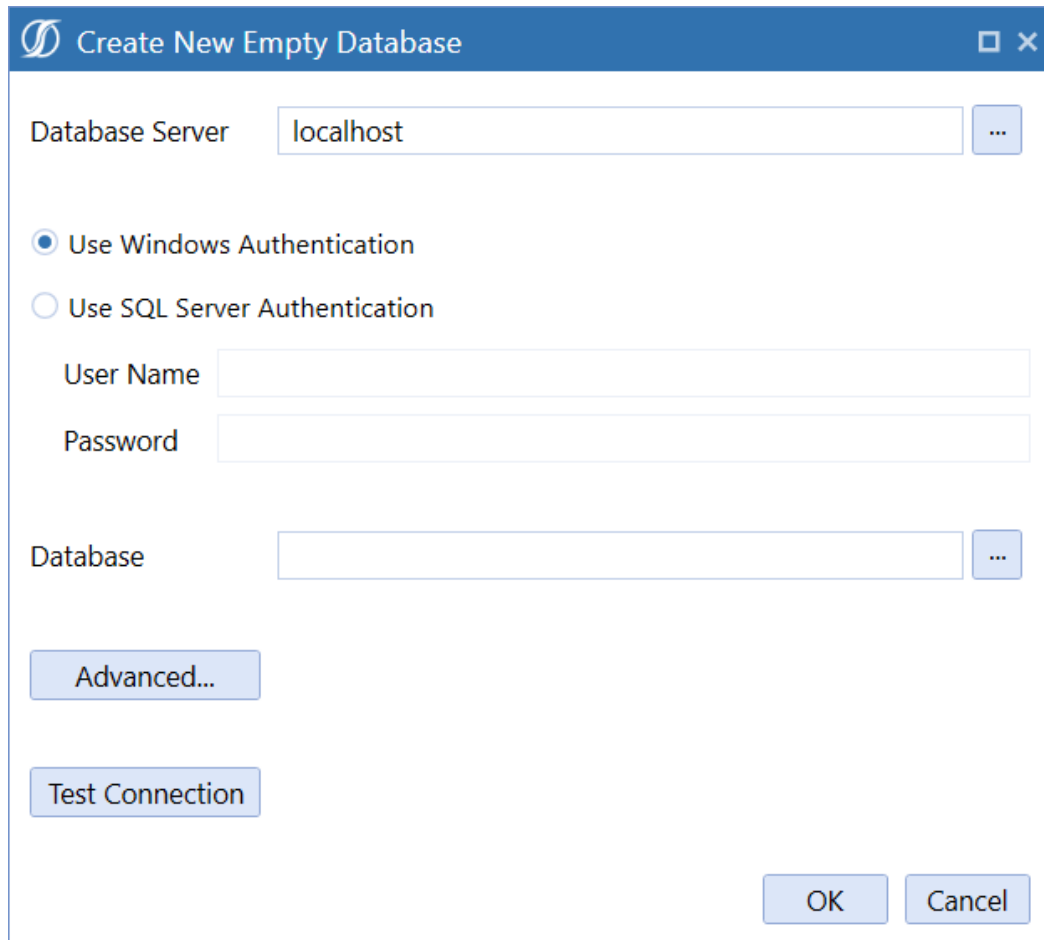
- Users

- 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 Databases

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
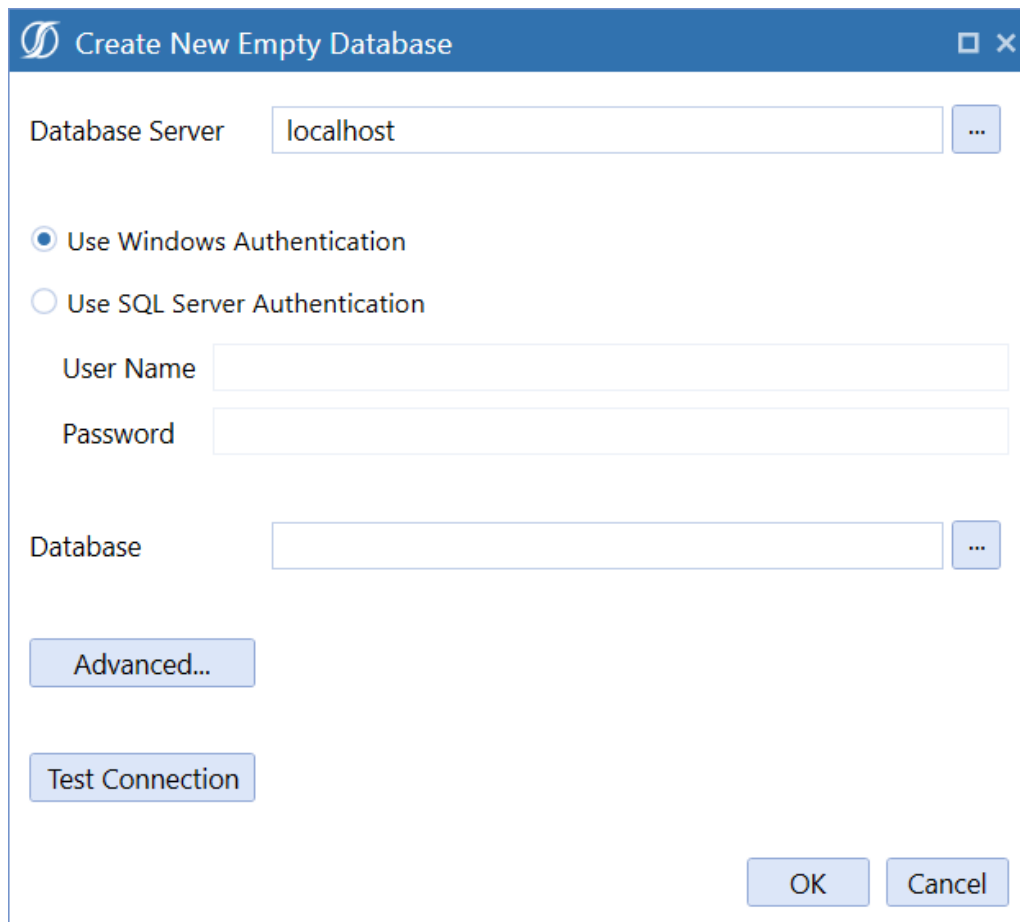
Creating an Application Database

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 Servers

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.

   > **TIP:** Create a central location accessible for all server configurations in which to house this file. For example, C:\OneStreamShare\Config

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

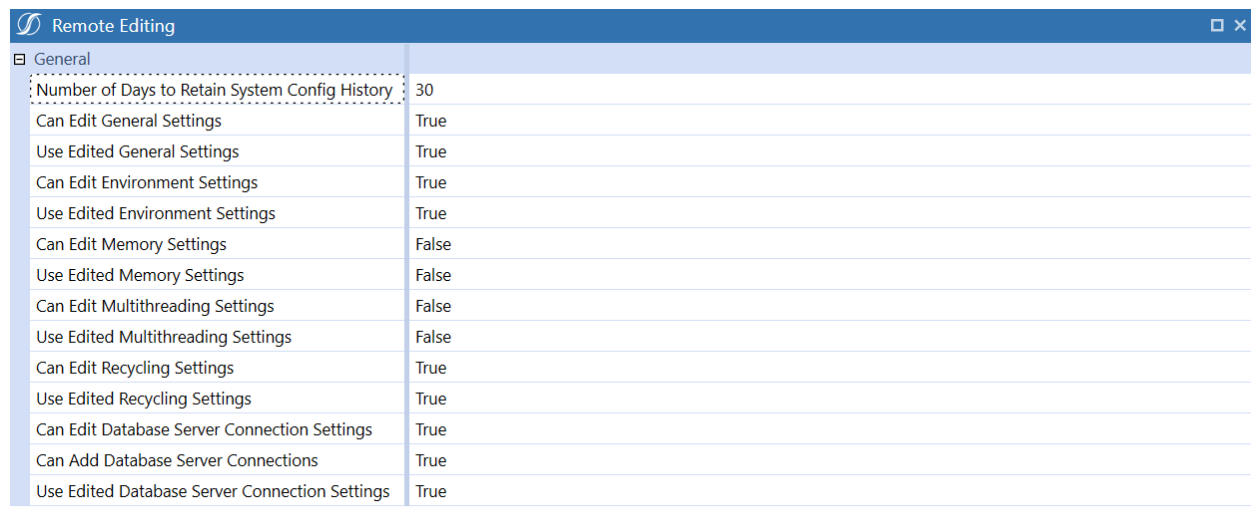# Application Server Configuration Settings

Remote Editing

Remote Editing allows adjusting Application Server Configuration setting access for Administrators and advanced IT persona. It is enabled by default but can be adjusted by Customer Support in the following manners:

- Disable Full Feature by XML/App Config

- Disable sections by XML/App Config

- Disable property changes

Remote Editing	
General	
Number of Days to Retain System Config History	30
Can Edit General Settings	True
Use Edited General Settings	True
Can Edit Environment Settings	True
Use Edited Environment Settings	True
Can Edit Memory Settings	False
Use Edited Memory Settings	False
Can Edit Multithreading Settings	False
Use Edited Multithreading Settings	False
Can Edit Recycling Settings	True
Use Edited Recycling Settings	True
Can Edit Database Server Connection Settings	True
Can Add Database Server Connections	True
Use Edited Database Server Connection Settings	True

- **Number of Days to Retain System Config History**: Set the number of days to retain the system config history.

- **Can Edit** selections: When True, users can make changes to the settings in the application. When False, modifying settings will no longer be available in the UI.

- **Use Edited** selections: When True, the user-defined settings apply. When False, the default settings from the configuration file apply.

- **Can Add Database Server Connections**: When True, users can add Custom Database Server Connections. When False, users are cannot add Custom Database Server Connections.

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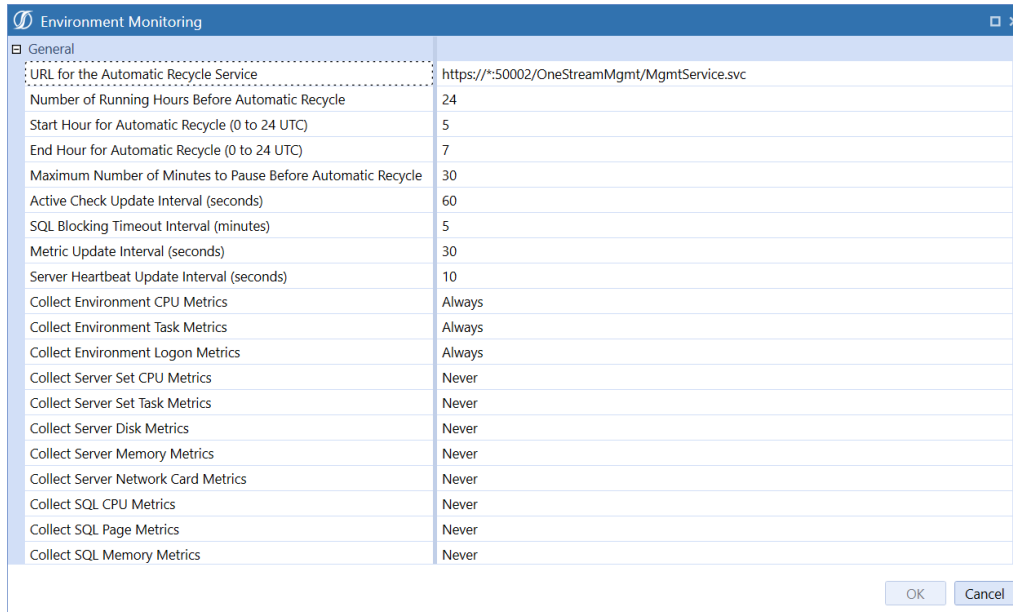
Use Environment Monitoring settings to define the real time update frequency, and the KPIs and metrics to monitor. These metric categories help you manage and optimize applications and the application environment:

- Environment

- Application server

- Database server

- Server set

Use 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

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.

Business Rules Default Access Group
Default is Administrators. The access group for new Business Rules.

Business Rules Default Maintenance Group
Default is Administrators. The maintenance group for new Business Rules.

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

Security

The Security section includes settings for Security, External Identity Providers, and Native Authentication.

Security

Whitelisted Domains: The Whitelisted Domains setting gives you the option to control the URLs and domains allowed to be referenced. You can update this setting with the following options:

- **AllowAll** (default): This option will allow all http and https formatted domains.

- **DenyAll**: This option will deny all URLs and domains. This option is recommended if there are concerns about the security of platform download features.

- Enter specific domains, URLs, or wildcard matches to accept.

If the Whitelisted Domains field is empty (displays Collection) or displays Default, it will default to AllowAll. If the OneStream Application Server Configuration file is manually updated to remove the Whitelisted Domains field, it will default to AllowAll.

> **IMPORTANT:** Regardless of the option selected, only http and https formatted domains can be referenced. References to non-http and non-https file schemes are not allowed.

To add an option for Whitelisted Domains:

1. In the **Application Server Configuration** file, click the ellipsis to the right of **Security**.

2. In the **Security** dialog box, click the ellipsis to the right of **Whitelisted Domains**.
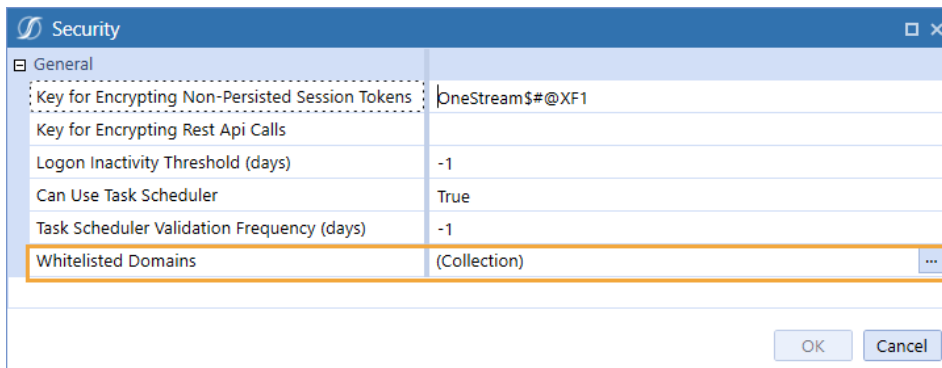
3. In the **Whitelisted Domains** dialog box, click the **+** icon.

4. Update the **Domain** field with the selected approach:

   - Type **AllowAll** to allow all http and https formatted domains.
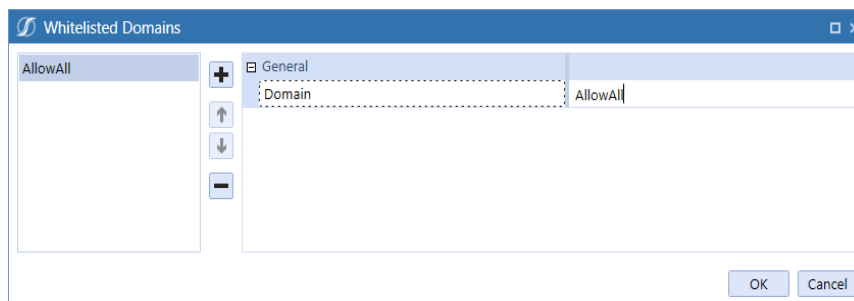
   - Type **DenyAll** to deny all URLs and domains.
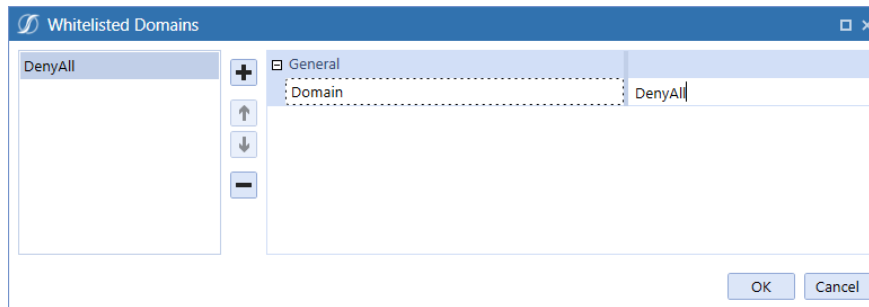
- Or, enter an absolute domain, an absolute URL, or a wildcard match using an * (for example, https://*.onestream.com or https://*.onestream.com/api/*) to allow specific URLs and domains.
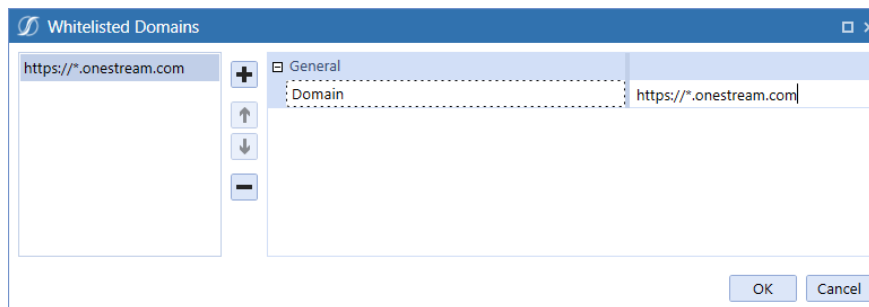
> **IMPORTANT:** If you add specific URLs or domains, only those identified URLs or domains will be allowed. All others will be denied.

5. Click the **OK** button.

> **NOTE:** If you are adding specific domains, URLs, or wildcard matches to accept, follow this process to add as many as needed.

6. Save changes and reset IIS.

To remove an option for Whitelisted Domains:

1. In the **Application Server Configuration** file, click the ellipsis to the right of **Security**.

2. In the **Security** dialog box, click the ellipsis to the right of **Whitelisted Domains**.

3. In the **Whitelisted Domains** dialog box, select the option from the list on the left and click the **-** icon.

4. Click the **OK** button.

5. Save changes and reset IIS.

External Authentication Providers

You can configure authentication with external identity providers. To configure, click the ellipsis to the right of **External Authentication Providers**.

> **NOTE:** When upgrading OneStream from a version prior version, note that the Windows Section of the external authentication provider may have 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 in the Name of Account Store field and the appropriate binding options and type of account store should be specified.

In the **External Authentication Providers** dialog box, click **+**.

Create a name for the provider and complete the information based on the installation. The final configuration is then performed in the security section of the individual user profile. See Authentication.

> **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 for an example of default configuration values.

Name of Account Store: The domain or server name 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.

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 drop-down menu or, if UseCombinedContextOptions is specified, enter multiple comma-separated values in the Combined Multiple Server Binding Options field. For a detailed description of these options, see Appendix: Context Option Values To Use With Active Directory + SSL.
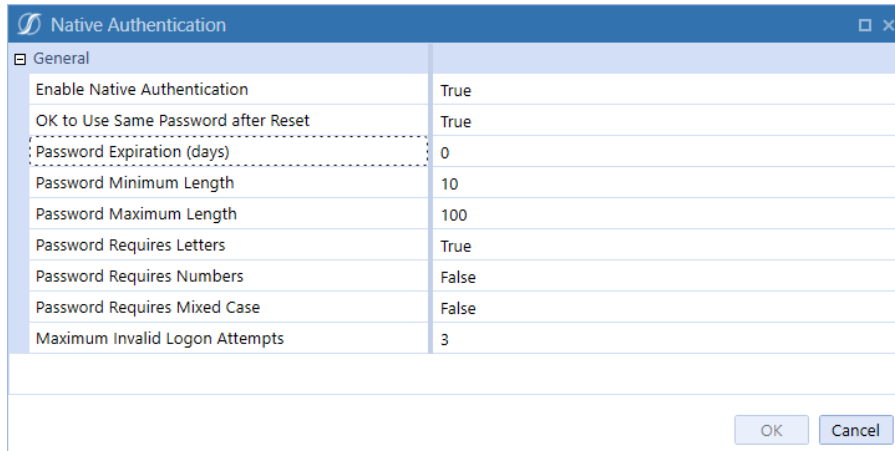
Type of Account Store: Specifies the type of store to which the principal (user) belongs. Possible values are ApplicationDirectory (represents the AD LDS store), Domain (represents AD DS store), and Machine (represents SAM store). Defaults value is Domain.

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

Native Authentication

See Authentication for information about setting up native authentication.

Control the standards for creating and updating passwords by defining the minimum and maximum character length, variety of characters, expiration dates, and maximum invalid logon attempts. If a positive number is provided for the Maximum Invalid Logon Attempts field and a user exceeds the number, their account will be disabled. After an administrator re-enables their account, the user will be required to change their password.

> **NOTE:** Similarly, the administrator's account can also be disabled if the maximum invalid logon attempts value is exceeded.

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.

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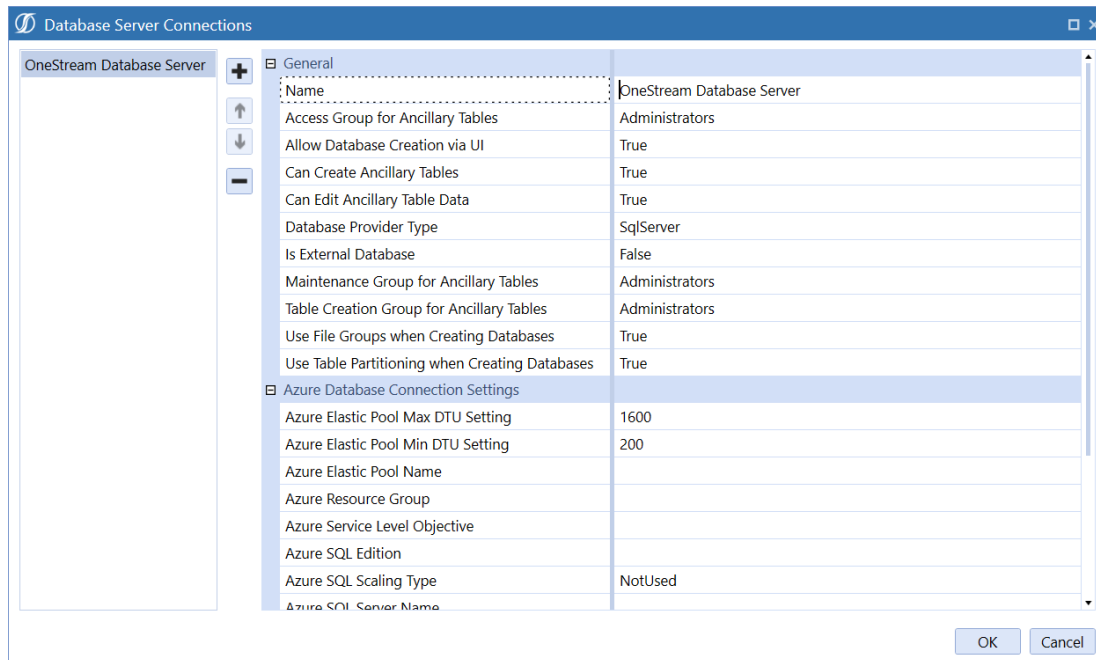
Maintenance Group for Ancillary Tables
This should be set to a group who will create the tables.

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:**
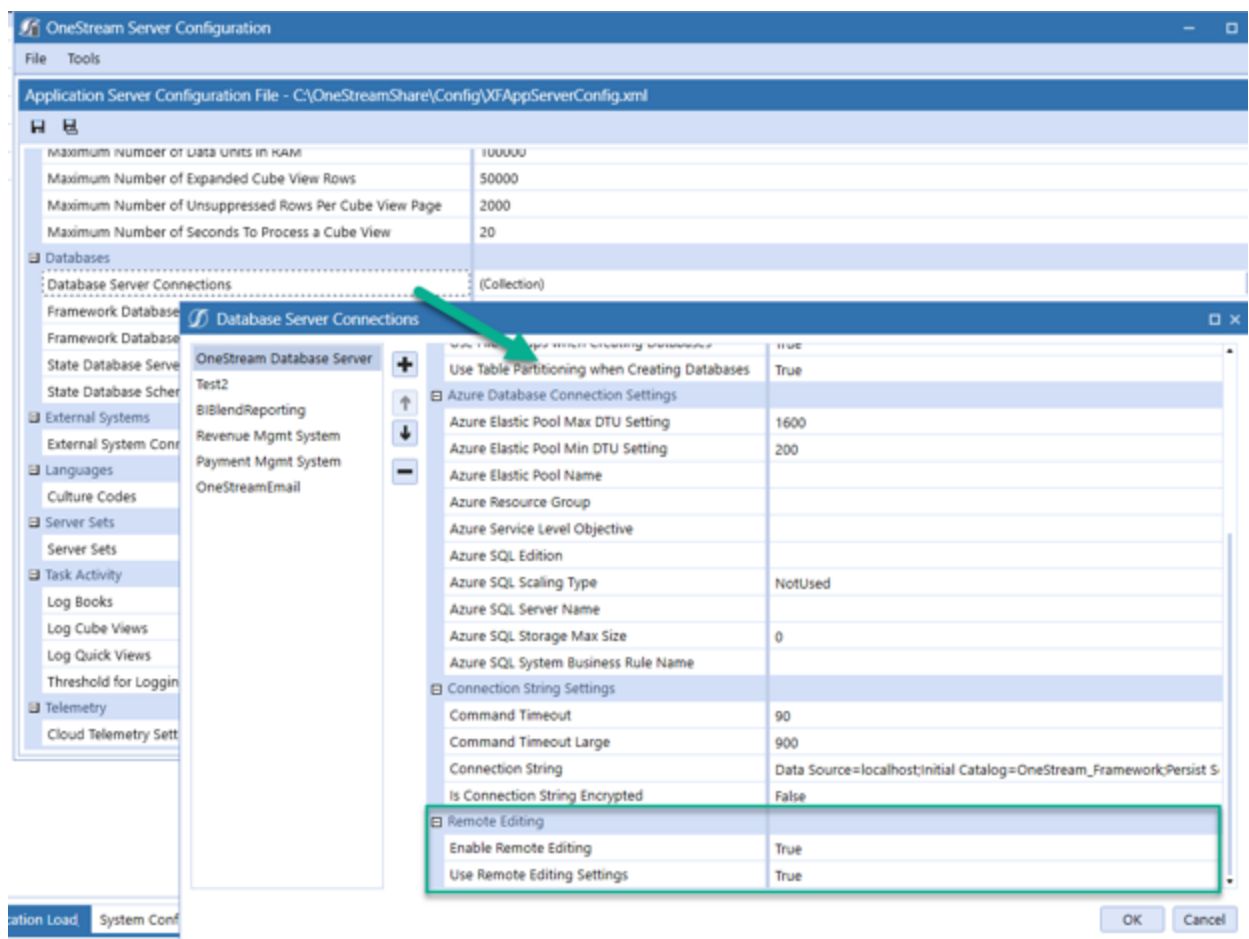
Enabling Remote Editing

Enable your database server connections for remote editing from the configuration file.

- **Enable Remote Editing**: When True, users can make changes to the settings in the application. When False, modifying Enable Remote Editing settings will no longer be available in the UI.

- **Use Remote Editing Settings**: When True, the user-defined settings apply. When False, the default settings from the configuration file apply.

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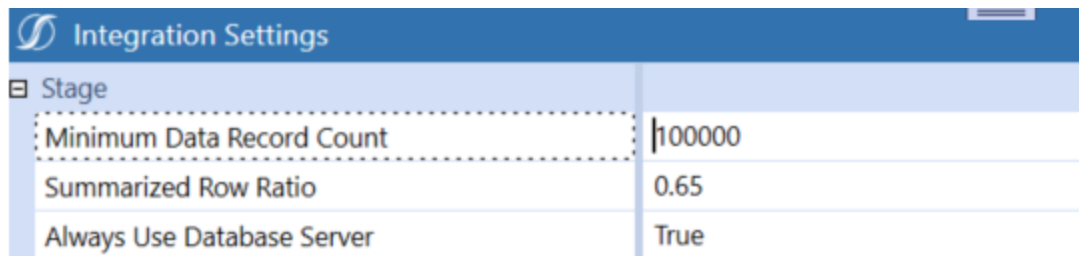
Can Recycle App Pool on Severs
If set to true, then the user can recycle the app pool from the Environment page via the Reset IIS button.

Integration Settings

These properties enable the modification of the Stage Summary Rows and Stage Load Values. These settings allow optimization and tuning of the Workflow Import behavior related to the summarization of records for the StageSummaryTargetData table to be performed on the application or database server. The summarization method of using the Application Server is determined by the calculated Summarized Row Ratio (SRR). The Summarized Row Ratio configuration property sets the threshold for the calculated SSR. Having less than 100k records, or high summarization, which is a calculated SRR less than the property, or default .65, will utilize the Application server. All Append, use of multiple Source ID's or a calculated SRR greater than the property value, or default .65, will use the Database server. The calculated SSR is done on subsequent Stage Import loads and stored in the Framework MetricsValue table.

Integration Settings		
⊟ Stage		
Minimum Data Record Count	100000	
Summarized Row Ratio	0.65	
Always Use Database Server	True	

Minimum Data Record Count
The default is 100,000. This determines the minimum data record count to trigger Database Server Processing. This threshold determines the use of the Application or Database server.

Summarized Row Ratio
The default is Greater than .65. This is the ratio of Total Summarized Rows/Total Number of Rows Imported. Subsequent loads generate a new Summarized Row Ratio. This threshold is used to evaluate the calculated SSR for the Application or Database servers.

Always Use Database Server
The default is "False". When set to "True", the feature that allows for Application Server Processing in Stage Import and Load during the Workflow processing is turned off. This disables the summarization of the Application server, Minimum Data Record Count and Summarized Row Ratio no longer apply.

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.

Zip File Protection

Zip File Protection		
Maximum Zip Archive Entry Threshold	■	1000
Maximum Zip File Extraction Size (MB)	■	2000
Maximum Zip File Compression Ratio (N:1)	■	800

Maximum Zip Archive Entry Threshold: Set the maximum number of files that can be included in a zip file. The default value is 1,000.

Maximum Zip File Extraction Size (MB): Set the maximum extraction size in MB (sum of all files) allowed in a zipped file. The default is 2,000.

Maximum Zip File Compression Ratio (N:1): Set the ratio between the uncompressed and compressed size (zip file size divided by packed size). The default is 800.

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 Servers

Configuring OneStream web servers requires one or more application servers 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.

SSO Logging

Logging is used to collect data to support troubleshooting issues. The following image shows the default logging settings.

```
"StartupSettings": {
                        "ConfigurationFolder":  "\\\\sgolympus.file.core.windows.net\\onestreamshareprd1\\config",
                        "EnableHTTP":  false,
                        "EnableHTTPS":  true
        },
"Serilog": {
                "MinimumLevel": {
                                "Default":  "Information",
                                "Override": {
                                                "Microsoft":  "Warning",
                                                "Microsoft.Hosting.Lifetime":  "Information"
                                }
                }
        },
"AllowedHosts":  "*"
}
```

To update the logging settings, you must perform these steps for each service you want to log on each server that is running.

1. Go to the file location:

   - **<installdrive>Program Files** > **OneStream Software** > **OneStreamWebRoot** > **OneStreamWeb**

   - **<installdrive>Program Files** > **OneStream Software** > **OneStreamAppRoot** > **OneStreamApp**

   - **<installdrive>Program Files** > **OneStream Software** > **OneStreamWebApiRoot** > **OneStreamWebApi**

   > **NOTE:** The location of the logs directory may vary depending on deployment and logging configuration, so log files may reside on each server.

2. Right-click the **appsettings.json** file and select the preferred text file editor.

You can update the following settings:

- **Default**: This value identifies the type of entries to be captured in the logs. It can be updated with the following values:

   - **Verbose**: The noisiest level, rarely enabled for a production application.

   - **Debug**: Used for internal system events that are not necessarily observable from the outside. It is useful when determining how something happened.

   - **Information**: Describes things happening in the system that correspond to its responsibilities and function. Generally, these are observable actions the system can perform.

   - **Warning**: Used when service is degraded, endangered, or may be behaving outside of its expected parameters.

   - **Error**: Used when functionality is unavailable or expectations are broken.

   - **Fatal**: The most critical level. Fatal events demand immediate attention.

- **Override**: Logging can be customized as needed by adding override levels (for example, SsoLogonService, Rsk.Saml, and Microsoft.AspNetCore.Authentication).

There are some values that are not included by default, but you can add them.

- **Name**: This value specifies which method should be used to write the audit data. Enter **File** to write the audit data to a file in the specified location indicated by the value entered in the path.

- **path**: This value specifies where the audit data is written (for example, C:\Logs).

- **rollingInterval**: This value indicates the time period in which the logs should be generated. All audit data will be captured and saved on a time interval as indicated in this field. You can update it with the following values: **Year**, **Month**, **Day**, **Hour**, or **Minute**.

The following image shows the logging settings updated to include the Name, path, and rollingInterval values.

```
1    "Serilog": {
2         "MinimumLevel": {
3                    "Default":  "Debug",
4                    "Override": {
5                              "Microsoft":  "Debug",
6                              "Microsoft.Hosting.Lifetime": "Information",
7                              "SsoLogonService":  "Debug",
8                              "Rsk.Saml": "Debug",
9                              "Microsoft.AspNetCore.Authentication":  "Debug"
10                       }
11             },
12             "WriteTo":  [
13    {
14    "Name": "File",
15    "Args": {
16      "path": "C:\\Logs\\log-.txt",
17      "rollingInterval":  "Day"
18    }
```
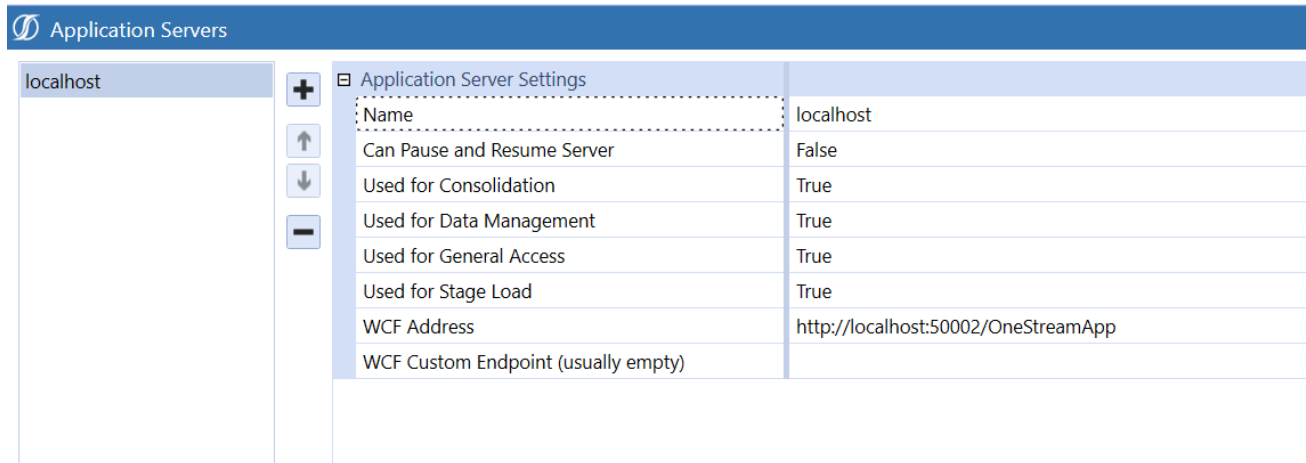
Creating Application Server Definitions

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.

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.

Authentication

This section provides an overview for OneStream authentication for customers in a self-hosted environment.

For customers in a OneStream-hosted environment, see the *Identity and Access Management Guide* for information about authentication with OneStream IdentityServer.

When users sign into a OneStream application, they go through an authentication process where they are required to confirm their identity. This is typically done by entering a user ID and password in the OneStream application. The user ID and password could be through an external authentication that is Cloud-based in which OneStream is not storing a user password or through OneStream native users in which 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 Authentication properties of the user indicate whether OneStream should use native authentication or an external single sign-on (SSO) to authenticate the user.

- Federated single sign-on enables applications to redirect to Azure AD (Microsoft Entra ID), 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 presents the user with a dialog box to enter the single user ID and password previously configured in an external provider that they may then use to sign into 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 that can eliminate or reduce the need to enter a user ID and password multiple times throughout the day. Consult with your identity provider for information on how to configure this experience.

    - For Azure AD (Microsoft Entra ID), 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."

The steps in this section provide information on configuring OneStream for external authentication with Transport Layer Security (TLS) encryption (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.

OneStream supports native authentication, authentication with one external identity provider, or both native authentication and one external identity provider.

OneStream supports the following external identity providers:

- Microsoft Active Directory (MSAD)

- Lightweight Directory Access Protocol (LDAP)

- Three OpenID Connect (OIDC) identity providers:

  - Azure Active Directory (Azure AD [Microsoft Entra ID])

  - Okta

  - PingFederate

- SAML 2.0 identity providers (for example, Okta, PingFederate, Active Directory Federation Services [ADFS], and Salesforce)

Set up authentication in the Application Server Configuration and Web Server Configuration in OneStream. This section includes instructions to set up for the following configurations:

- Native authentication only

- Single sign-on with an external identity provider only

- Native authentication and single sign-on with an external identity provider

Set Up for Native Authentication

1. Open the **OneStream Server Configuration Utility** application.

2. Go to **File** > **New Application Server Configuration File**.

   > **NOTE:** Alternatively, you can open an existing file to edit it.

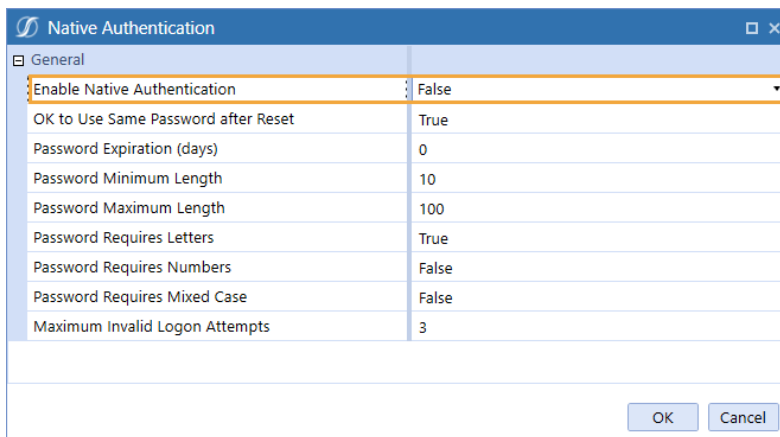3. In the **Security** section, click the ellipsis to the right of **Native Authentication**.
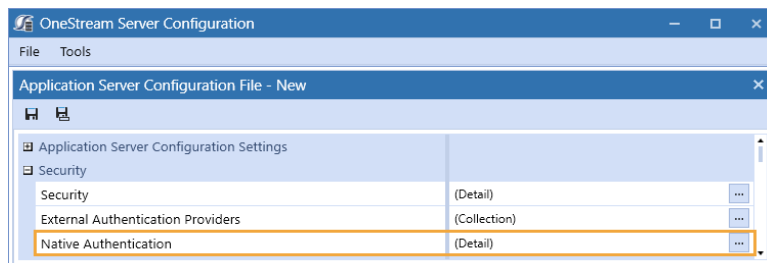
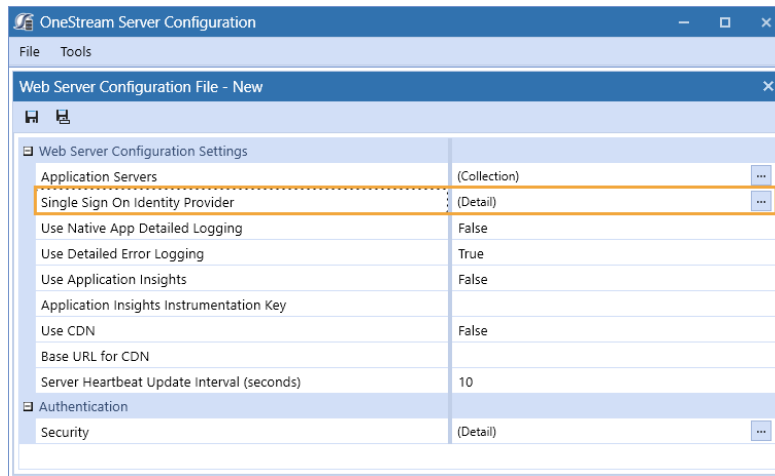4. In the **Enable Native Authentication** drop-down menu, select **True**.

> **IMPORTANT:** If **Enable Native Authentication** is set to False, the user will receive an error when attempting to log into OneStream using native authentication.

Native Authentication	□ ×
☐ General	
Enable Native Authentication	True ▾
OK to Use Same Password after Reset	True
Password Expiration (days)	0
Password Minimum Length	10
Password Maximum Length	100
Password Requires Letters	True
Password Requires Numbers	False
Password Requires Mixed Case	False
Maximum Invalid Logon Attempts	3
	OK Cancel

5. Click the **OK** button and save changes.

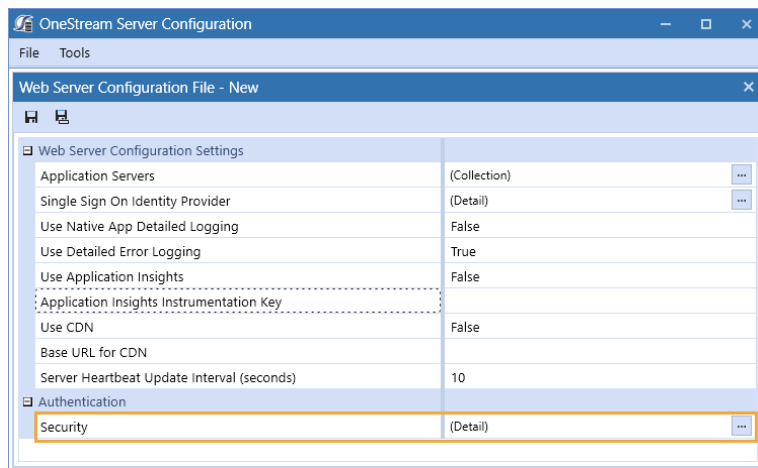6. Go to **File** > **New Web Server Configuration File**.

> **NOTE:** Alternatively, you can open an existing file to edit it.

7. In the **Web Server Configuration Settings** section, click the ellipsis to the right of **Single Sign On Identity Provider**.
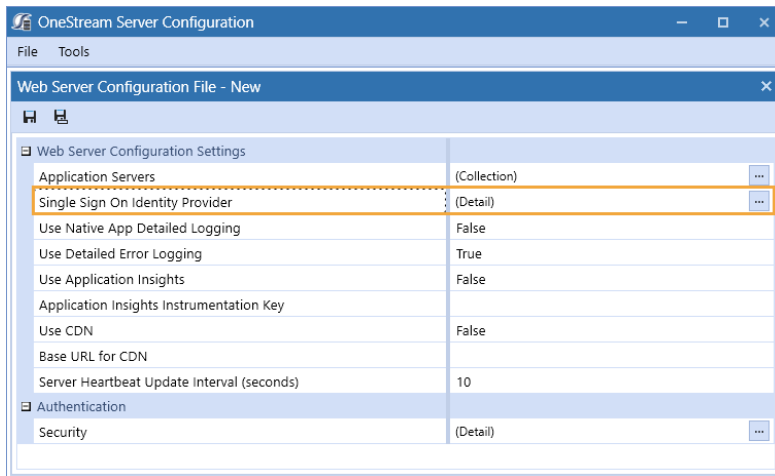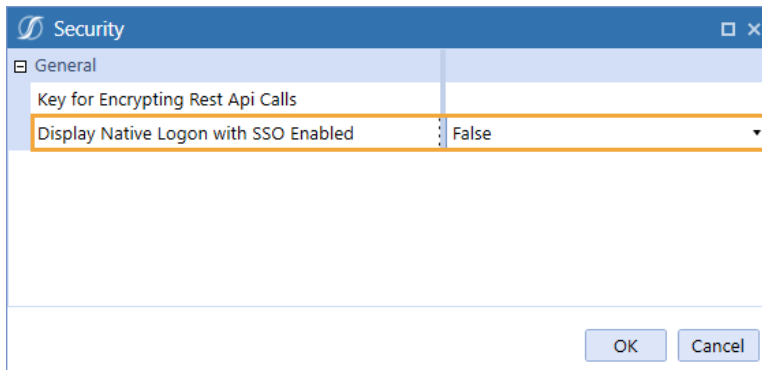
8. In the **General** section, in the **SSO Identity Provider Type** drop-down menu, select **NotUsed**.

9. In the **Authentication** section, click the ellipsis to the right of **Security**.
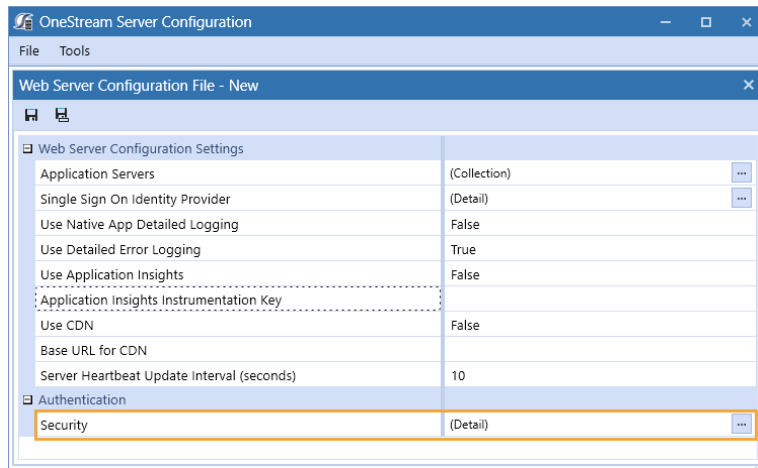
10. In the **Display Native Logon with SSO Enabled** drop-down menu, select **False**.
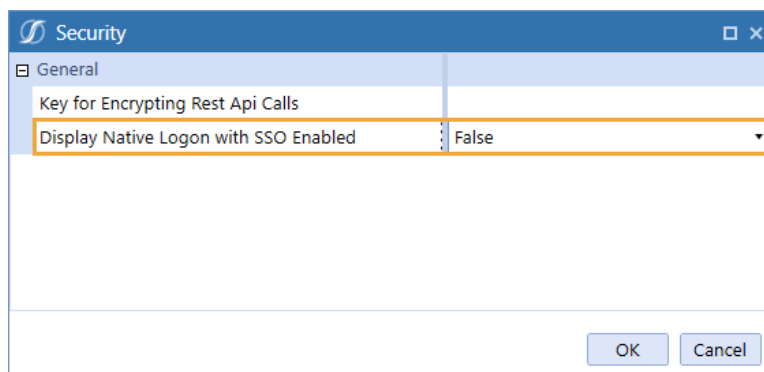
11. Click the **OK** button.
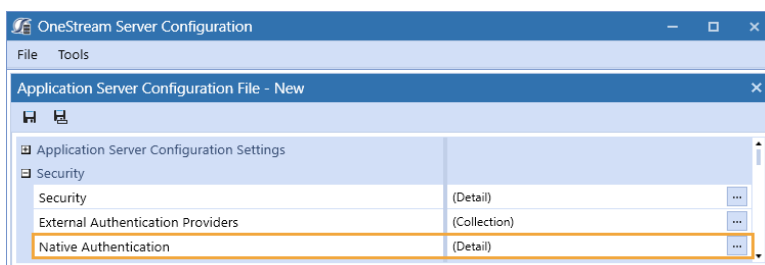
12. Save changes and reset IIS.

> **NOTE:** Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.

Set Up for Single Sign-on with an External Identity Provider

1. Open the **OneStream Server Configuration Utility** application.

2. Go to **File** > **New Application Server Configuration File**.

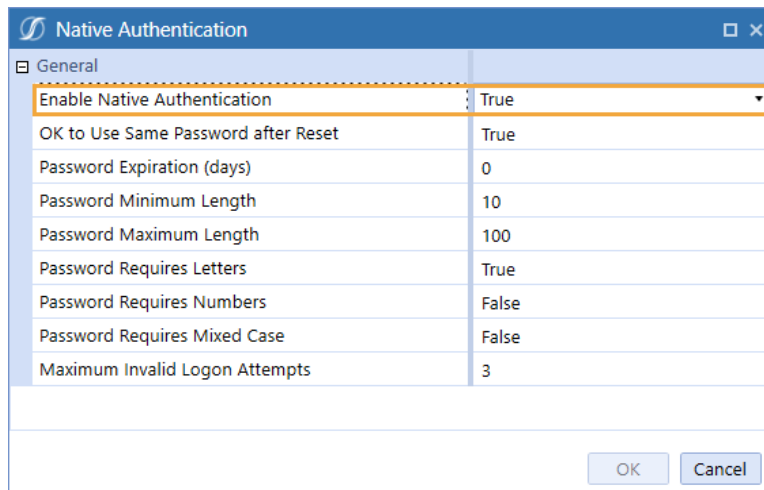> **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Security** section, click the ellipsis to the right of **Native Authentication**.
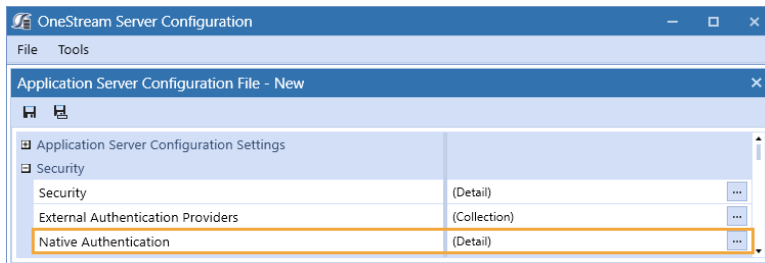
4. In the **Enable Native Authentication** drop-down menu, select **False**.

5. Click the **OK** button and save changes.

6. Go to **File** > **New Web Server Configuration File**.

> **NOTE:** Alternatively, you can open an existing file to edit it.

7. In the **Web Server Configuration Settings** section, click the ellipsis to the right of **Single Sign On Identity Provider**.

8. In the **General** section, in the **SSO Identity Provider Type** drop-down menu, select the type of external identity provider: Azure, Okta, PingFederate, Saml, or OpenId.

9. In the **Authentication** section, click the ellipsis to the right of **Security**.

10. In the **Display Native Logon with SSO Enabled** drop-down menu, select **False**.

11. Click the **OK** button.

12. Save changes and reset IIS.

> **NOTE:** Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.

Set Up for Native Authentication and Single Sign-on with an External Identity Provider

1. Open the **OneStream Server Configuration Utility** application.

2. Go to **File** > **New Application Server Configuration File**.

> **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Security** section, click the ellipsis to the right of **Native Authentication**.

4. In the **Enable Native Authentication** drop-down menu, select **True**.

> **IMPORTANT:** If Enable Native Authentication is set to False, the user will receive an error when attempting to log into OneStream using native authentication.

5. Click the **OK** button and save changes.

6. Go to **File** > **New Web Server Configuration File**.

    > **NOTE:** Alternatively, you can open an existing file to edit it.

7. In the **Web Server Configuration Settings** section, click the ellipsis to the right of **Single Sign On Identity Provider**.
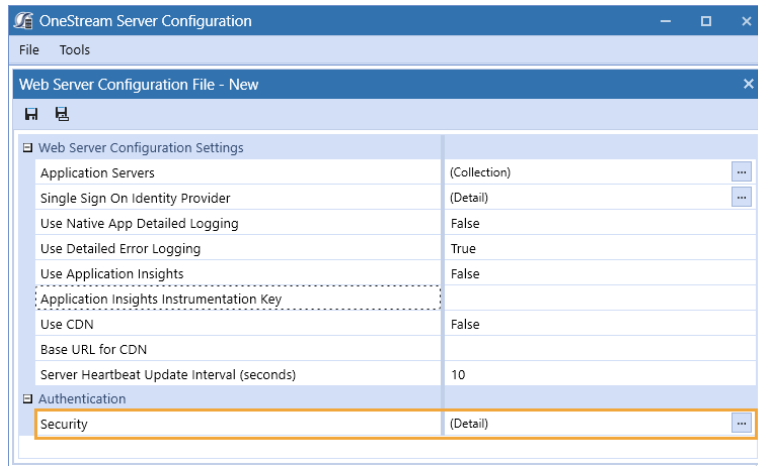
8. In the **General** section, in the **SSO Identity Provider Type** drop-down menu, select the type of external identity provider: Azure, Okta, PingFederate, Saml, or OpenId.

9. In the **Authentication** section, click the ellipsis to the right of **Security**.

10. In the **Display Native Logon with SSO Enabled** drop-down menu, select **True**.

11. Click the **OK** button.

12. Save changes and reset IIS.

    > **NOTE:** Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.

This section includes instructions for configuring the following types of authentication:

- [Native Authentication Configuration](#)

- [MSAD Configuration](#)

- [LDAP Configuration](#)

- [Microsoft Azure AD (Microsoft Entra ID) Configuration (OIDC)](#)

- [Okta Configuration (OIDC)](#)

- [PingFederate Configuration (OIDC)](#)

- [SAML 2.0 Configuration with Okta](#)

- [SAML 2.0 Configuration with PingFederate](#)

- [SAML 2.0 Configuration with ADFS](#)

- [SAML 2.0 Configuration with Salesforce](#)

Native Authentication Configuration

To enable native authentication, follow these steps:

1. [Set Up for Native Authentication](#) or [Set Up for Native Authentication and Single Sign-on with an External Identity Provider](#).

2. [Set Up OneStream Login with Native Authentication](#).

3. [Log into OneStream with Native Authentication](#).

Set Up OneStream Login with Native Authentication

1. In the OneStream Desktop application, go to **System** > **Security** > **Users** > **<user>**.

2. In the **Authentication** section, complete the following fields for native authentication.

   - **External Authentication Provider**: In the drop-down menu, select **(Not Used)**.

   - **External Provider User Name**: Leave this field blank.

- **Internal Provider Password**: Enter a password.

☐ Authentication	
External Authentication Provider	(Not Used)
External Provider User Name	
Internal Provider Password	••••••

3. Click the **Save** icon.

Log into OneStream with Native Authentication

1. On the Desktop application Logon screen, for Server Address, specify the URL or a client connection.

2. Click the **Connect** button.

3. Enter your user profile name and internal provider password.

4. Click the **Logon** button.

5. Select an application from the drop-down menu.

6. Click the **Open Application** button.

> **NOTE:** To log off, on any screen, click the **Logoff** icon and then click the **End Session** button.

MSAD Configuration

To enable single sign-on with MSAD, follow these steps:

1. Set Up for Native Authentication or Set Up for Native Authentication and Single Sign-on with an External Identity Provider.

2. Set Up the Application Server Configuration in OneStream.

3. Set Up the Web Server Configuration in OneStream.

4. Set Up OneStream Login with MSAD.

5. Log into OneStream with MSAD.

Set Up the Application Server Configuration in OneStream

1. Open the **OneStream Server Configuration Utility** application.

2. Go to **File** > **New Application Server Configuration File**.

   > **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Security** section, click the ellipsis to the right of **External Authentication Providers**.
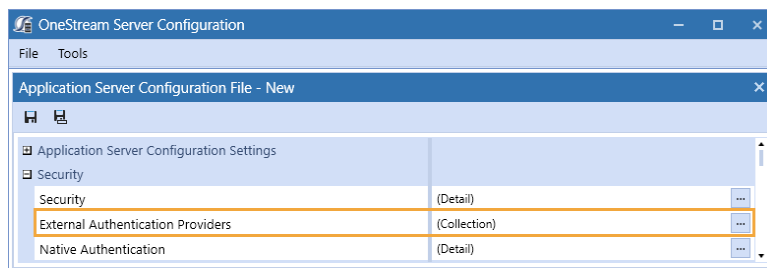
4. Click the **+** icon to add an item.

5. In the **General** and **Windows** sections, complete the following fields:

   - **Name** : Enter the name of the identity provider. This name will display when configuring users to their external SSO.

   - **Authentication Provider Type**: Select **Windows** in the drop-down menu.

   - **Name of Account Store**: Enter the domain name of the active directory.

     > **TIP:** In **Active Directory Users and Computers**, you can view the domain name of the active directory.

   - **Name of Account Store Container**: Enter the name of the domain controller (DC). Leave the other values as default.

     > **TIP:** In **Active Directory Users and Computers**, under the active directory, click **Domain Controllers** to view the name of the domain

controller.

- **Type of Account Store**: Select **Domain** in the drop-down menu.

See [External Authentication Providers](#).

6. Click the **OK** button.

7. Save changes and reset IIS.

> **NOTE:** Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.
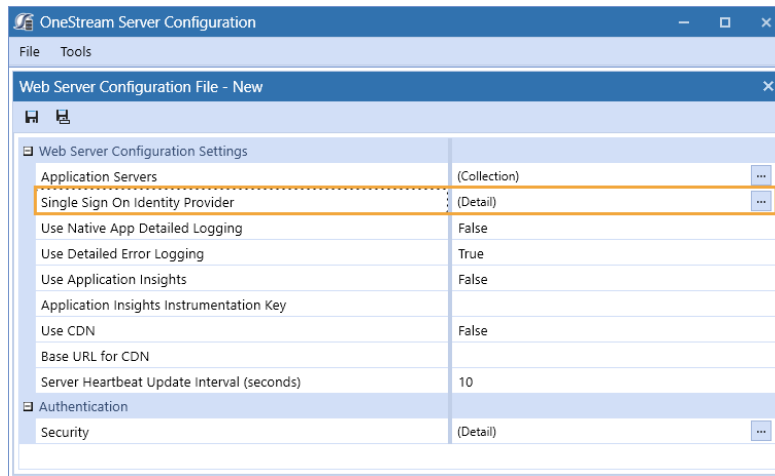
Set Up the Web Server Configuration in OneStream

1. Open the **OneStream Server Configuration Utility** application.

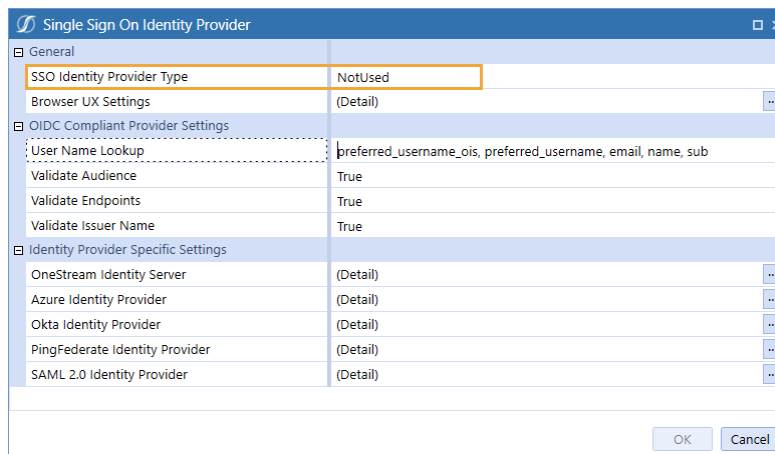2. Go to **File** > **New Web Server Configuration File**.

> **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Web Server Configuration Settings** section, click the ellipsis to the right of **Single Sign On Identity Provider**.

4. In the **SSO Identity Provider Type** drop-down menu, select **NotUsed**.

5. Click the **OK** button.

6. Save changes and reset IIS.

> **NOTE:** Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.

Set Up OneStream Login with MSAD

1. In the OneStream Desktop application, go to **System** > **Security** > **Users** > **<user>**.

2. In the **General** and **Authentication** properties, complete the following fields for MSAD authentication.

    - **Name**: Enter the user logon name listed in active directory.

        > **TIP:** In **Active Directory Users and Computers**, under the active directory, click **Users** to view the list of users. Select the user. Click the **Account** tab to view the user logon name.

    - **External Authentication Provider**: In the drop-down menu, select the MSAD configuration.

    - **External Provider User Name**: Leave this field blank.

⊟ General	
Name	Username
Description	
User Type	(Select One)
Is Enabled	True
⊞ Status	
⊟ Authentication	
External Authentication Provider	
External Provider User Name	
Internal Provider Password	••••••

3. Click the **Save** icon.

Log into OneStream with MSAD

1. On the Desktop application Logon screen, for Server Address, specify the URL or a client connection.

2. Click the **Connect** button.

3. Enter your user name and password set up in MSAD.

4. Click the **Logon** button.

5. Select an application from the drop-down menu.

6. Click the **Open Application** button.

> **NOTE:** To log off, on any screen, click the **Logoff** icon and then click the **End Session** button.

LDAP Configuration

To enable single sign-on with LDAP, follow these steps:

1. Set Up for Native Authentication or Set Up for Native Authentication and Single Sign-on with an External Identity Provider.

2. Set Up the Application Server Configuration in OneStream.

3. Set Up the Web Server Configuration in OneStream.

4. Set Up OneStream Login with LDAP.

5. Log into OneStream with LDAP.

Set Up the Application Server Configuration in OneStream

1. Open the **OneStream Server Configuration Utility** application.

2. Go to **File** > **New Application Server Configuration File**.

   > **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Security** section, click the ellipsis to the right of **External Authentication Providers**.
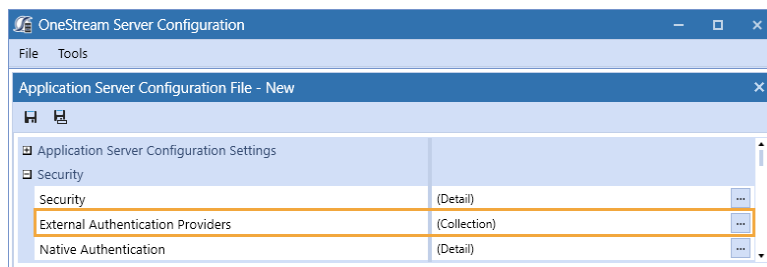
4. Click the **+** icon to add an item.

5. In the **General** and **LDAP** sections, complete the following fields:

   - **Name**: Enter the name of the identity provider. This name will display when configuring users to their external SSO.

   - **Authentication Provider Type**: Select **LDAP** in the drop-down menu.

   - **LDAP Authentication Type**: Select **Basic** in the drop-down menu.

   - **LDAP Base DN**: Enter the distinguished name (DN) of the container or organizational unit where your OneStream users' accounts are located.

     > **TIP:** The base DN is the full path to the location of the user accounts in the authentication hierarchy. LDAP will look for user accounts in this location. If needed, contact your IT Support for this information.

   - **LDAP Server or Domain**: Enter the fully qualified domain name (FQDN) of your active directory domain or domain controller.

See [External Authentication Providers](#).

6. Click the **OK** button.

7. Save changes and reset IIS.

> **NOTE:** Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.

Set Up the Web Server Configuration in OneStream

1. Open the **OneStream Server Configuration Utility** application.

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

   > **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Web Server Configuration Settings** section, click the ellipsis to the right of **Single Sign On Identity Provider**.

4. In the **SSO Identity Provider Type** drop-down menu, select **NotUsed**.
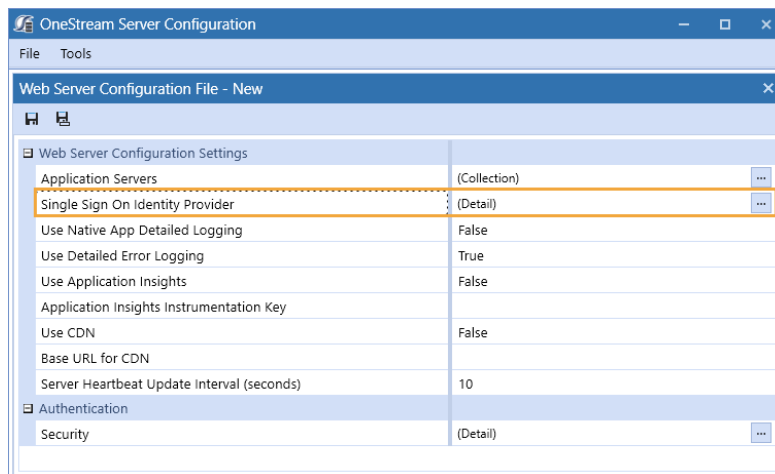
5. Click the **OK** button.

6. Save changes and reset IIS.

> **NOTE:** Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.

Set Up OneStream Login with LDAP

1. In the OneStream Desktop application, go to **System** > **Security** > **Users** > **<user>**.

2. In the **General** and **Authentication** properties, complete the following fields for LDAP authentication:

   - **Name**: Enter the user logon name listed in active directory.

     > **TIP:** In **Active Directory Users and Computers**, under the active directory, click **Users** to view the list of users. Select the user. Click the **Account** tab to view the user logon name.

   - **External Authentication Provider**: In the drop-down menu, select the LDAP configuration.

   - **External Provider User Name**: Enter the display name listed in the active directory.

> **TIP:** In **Active Directory Users and Computers**, under the active directory, click **Users** to view the list of users. Select the user. Click the **General** tab to view the display name.

General	
Name	Username
Description	
User Type	(Select One)
Is Enabled	True
⊞ Status	
⊟ Authentication	
External Authentication Provider	
External Provider User Name	Displayname
Internal Provider Password	••••••

3. Click the **Save** icon.

Log into OneStream with LDAP

1. On the Desktop application Logon screen, for Server Address, specify the URL or a client connection.

2. Click the **Connect** button.

3. Enter your user name (user logon name) and password set up in the active directory.

4. Click the **Logon** button.

5. Select an application from the drop-down menu.

6. Click the **Open Application** button.

> **NOTE:** To log off, on any screen, click the **Logoff** icon and then click the **End Session** button.

Microsoft Azure AD (Microsoft Entra ID) Configuration

To enable single sign-on with Azure AD (Microsoft Entra ID) for the OneStream Desktop application, which includes the Windows Client application and the Excel Add-In, using OIDC protocol, follow these steps:

1. [Set Up for Single Sign-on with an External Identity Provider](#) or [Set Up for Native Authentication and Single Sign-on with an External Identity Provider](#).

2. [Set Up the Application Server Configuration in OneStream](#).

3. [Set Up the Application in Azure AD (Microsoft Entra ID)](#).

4. [Set Up the Web Server Configuration in OneStream](#).

5. [Set Up OneStream Login with Azure AD (Microsoft Entra ID)](#).

6. [Log into OneStream with Azure AD (Microsoft Entra ID)](#).

To configure OneStream REST API to support Azure AD (Microsoft Entra ID) authentication, see the *REST API Implementation Guide*.

Set Up the Application Server Configuration in OneStream

1. Open the **OneStream Server Configuration Utility** application.

2. Go to **File** > **New Application Server Configuration File**.

   > **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Security** section, click the ellipsis to the right of **External Authentication Providers**.

4. Click the **+** icon to add an item.

5. In the **General** and **External Provider Single Sign On** sections, complete the following fields:

- **Name**: Enter the name of the identity provider. This name will display when configuring users to their external SSO.

- **Authentication Provider Type**: Select **AzureSSO** in the drop-down menu.

- **External Provider Web SSO Secret Key**: Enter a unique value. This key is used in the OneStream Application Server Configuration and the OneStream Web Server Configuration. It enables your application server to communicate with the web server.
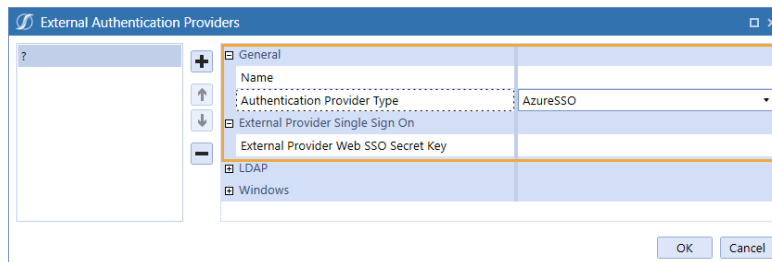
6. Click the **OK** button.

7. Save changes and reset IIS.

> **NOTE:** Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.

Set Up the Application in Azure AD (Microsoft Entra ID)

To set up the application in Azure AD (Microsoft Entra ID), you must copy the application (client) ID and tenant ID (directory ID) from Azure AD (Microsoft Entra ID) and paste them into the Web Server Configuration in OneStream. Also, enter the redirect URI in Azure AD (Microsoft Entra ID) in this format: https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx

1. Log into your Azure AD account.

2. On the Home screen, click the **App registrations** icon.

3. On the **App registrations** page, click the **+ New registration** tab.

4. On the **Register an application** page, complete the following fields:

       a. Enter a name for the application.

       b. For **Supported account types**, select **Accounts in this organization directory only**.

       c. For **Redirect URI**, select **Public client/native**.

5. Click the **Register** button.

6. On the page for the application, click **Add a Redirect URI**.

7. On the **Authentication** page, click the **+ Add a platform** button.

8. In the **Configure platforms** section to the right, select the **Mobile and desktop applications** icon.

9. In the **Custom redirect URIs** field, enter the redirect URI in this format:
   https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx

10. Click the **Configure** button.

> **TIP:** Return to the **App registrations** page in Azure AD and then select the application to access information needed for the Web Server Configuration in OneStream: application (client) ID, directory (tenant) ID, and redirect URI.

Set Up the Web Server Configuration in OneStream

1. Open the **OneStream Server Configuration Utility** application.

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

> **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Web Server Configuration Settings** section, click the ellipsis to the right of **Single Sign On Identity Provider**.
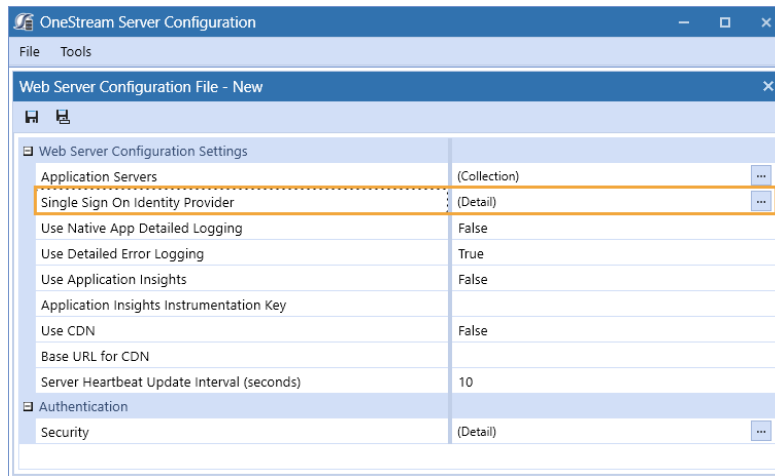
4. Click the ellipsis to the right of **Azure Identity Provider**.
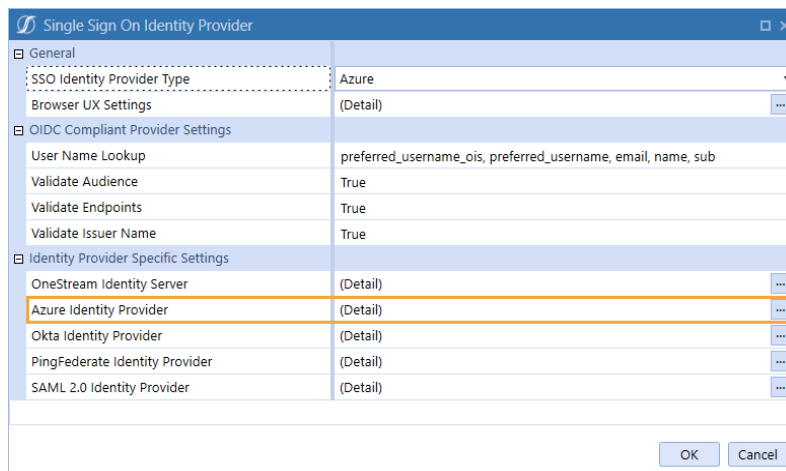
5. In the **Azure Identity Provider** dialog box, in the **General**, **Browser UX Settings**, and **Windows Desktop Client Settings** sections, complete the following fields:

- **Azure AD Tenant Id**: Enter the directory (tenant) ID from Azure AD .

> **TIP:** To view the directory (tenant) ID in Azure AD, go to the page for the application and select **Overview** in the list on the left.

- **Azure OpenID Connect Scopes**: Enter scopes, or leave as default (openid email profile).

- **OneStream Web App Client Secret Key**: Enter the same value from the External Provider Web SSO Secret Key field in the Application Server Configuration.

- **OneStream Windows App Client ID**: Enter the application (client) ID from Azure AD. See Set Up the Application in Azure AD (Microsoft Entra ID) step 10.

- **OneStream Windows App Redirect Url**: Enter the value you entered in Azure AD as the redirect URI. See Set Up the Application in Azure AD (Microsoft Entra ID) step 10. Use this format:
https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx

> **IMPORTANT:** The redirect URI entered in Azure and in the OneStream Windows App Redirect Url field need to match exactly, including capitalization.

6. Click the **OK** button.

7. Save changes and reset IIS.

> **NOTE:** Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.

Set Up OneStream Login with Azure AD (Microsoft Entra ID)

1. In the OneStream Desktop application, go to **System** > **Security** > **Users** > **<user>**.

2. In the **Authentication** properties, complete the following fields for authentication through Azure AD.

    - **External Authentication Provider**: In the drop-down menu, select the Azure AD configuration.

    - **External Provider User Name**: Enter the username configured in Azure AD. This name must match the username set up in Azure AD and be used by only one user.

⊟ Authentication	
External Authentication Provider	AzureSSO
External Provider User Name	Username
Internal Provider Password	••••••

3. Click the **Save** icon.

Log into OneStream with Azure AD (Microsoft Entra ID)

1. On the OneStream Desktop application Logon screen, for **Server Address**, specify the URL or a client connection.

2. Click the **Connect** button.

3. Click the **External Provider Sign In** button.

4. Enter your Azure AD login credentials.

5. Select an application from the drop-down menu.

6. Click the **Open Application** button.

> **NOTE:** To log off, on any screen, click the **Logoff** icon and then click the **End Session** button.

Okta Configuration

To enable single sign-on with Okta for the OneStream Desktop application, which includes the Windows Client application and the Excel Add-In, using OIDC protocol, follow these steps:

1. [Set Up for Single Sign-on with an External Identity Provider](#) or [Set Up for Native Authentication and Single Sign-on with an External Identity Provider](#).

2. [Set Up the Application Server Configuration in OneStream](#).

3. [Set Up the Application in Okta](#).

4. [Set Up the Web Server Configuration in OneStream](#).

5. [Set Up OneStream Login with Okta](#).

6. [Log into OneStream with Okta](#).

To configure OneStream REST API to support Okta authentication, see the *REST API Implementation Guide*.

Set Up the Application Server Configuration in OneStream

1. Open the **OneStream Server Configuration Utility** application.

2. Go to **File** > **New Application Server Configuration File**.

   > **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Security** section, click the ellipsis to the right of **External Authentication Providers**.

4. Click the **+** icon to add an item.

5. In the **General** and **External Provider Single Sign On** sections, complete the following fields:

   - **Name**: Enter the name of the identity provider. This name will display when configuring users to their external SSO.

   - **Authentication Provider Type**: Select **Okta SSO** in the drop-down menu.

   - **External Provider Web SSO Secret Key**: Enter a unique value. This key is used in the OneStream Application Server Configuration and the OneStream Web Server Configuration. It enables your application server to communicate with the web server.
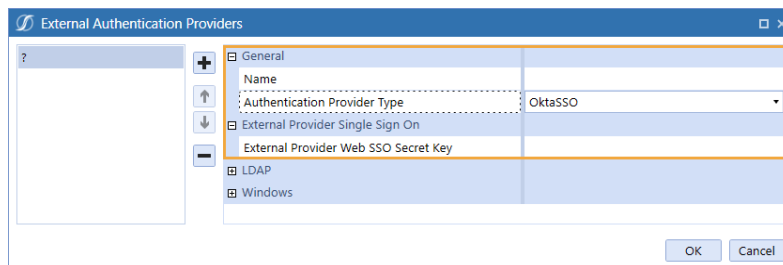
6. Click the **OK** button.

7. Save changes and reset IIS.

   > **NOTE:** Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.

Set Up the Application in Okta

To set up the application in Okta, you must copy the client ID from Okta and paste it into the Web Server Configuration in OneStream. Also, enter the redirect URI in Okta in this format: https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx

1. Log into your Okta account.

2. In the **Applications** list on the left, select **Applications**.

3. Click **Create App Integration**.

4. In the **Create a new app integration** dialog box, for **Sign-in method**, select **OIDC - OpenID Connect**.

5. For **Application type**, select **Native Application**.

6. Click the **Next** button.

7. On the **New Single-Page App Integration** page, complete the following fields:

    - **App integration name**: Enter the name of the application in Okta.

    - **Grant type**: Select **Authorization Code** and **Refresh Token**.

    - **Sign-in redirect URIs**: Enter the sign-in redirect URI in this format: https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx

    - **Sign-out redirect URIs**: Click **X** to clear the field.

    - **Controlled access**: Select the access option for the application.

8. Click the **Save** button.

9. Copy the client ID. You will need to paste this into the Web Server Configuration in OneStream.

Set Up the Web Server Configuration in OneStream

1. Open the **OneStream Server Configuration Utility** application.

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

   > **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Web Server Configuration Settings** section, click the ellipsis to the right of **Single Sign On Identity Provider**.
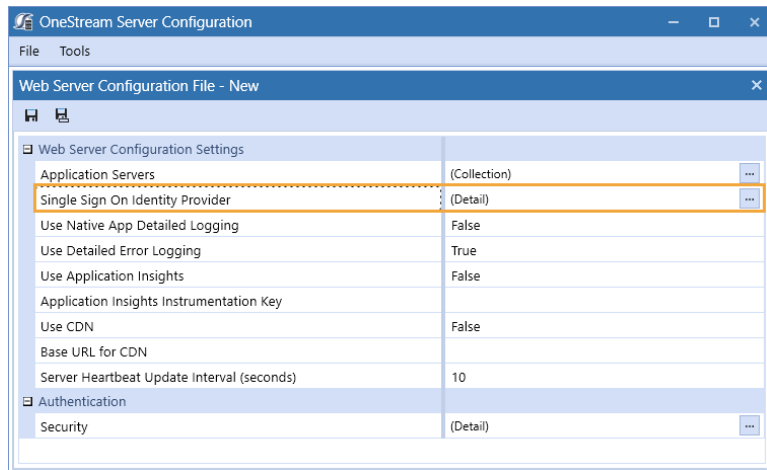
4. Click the ellipsis to the right of **Okta Identity Provider**.
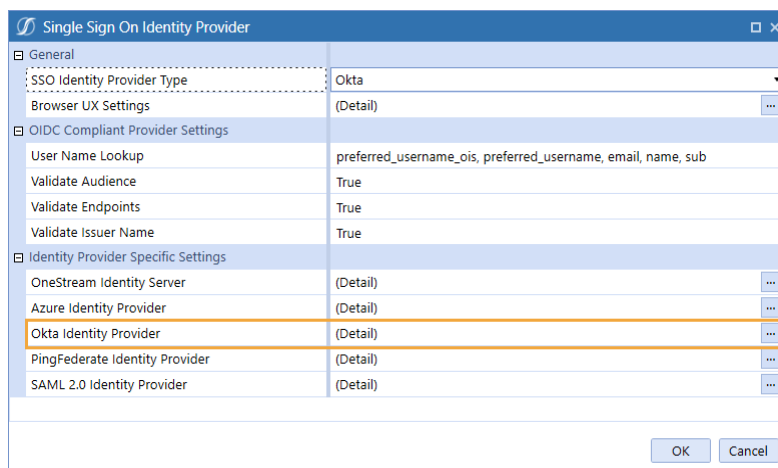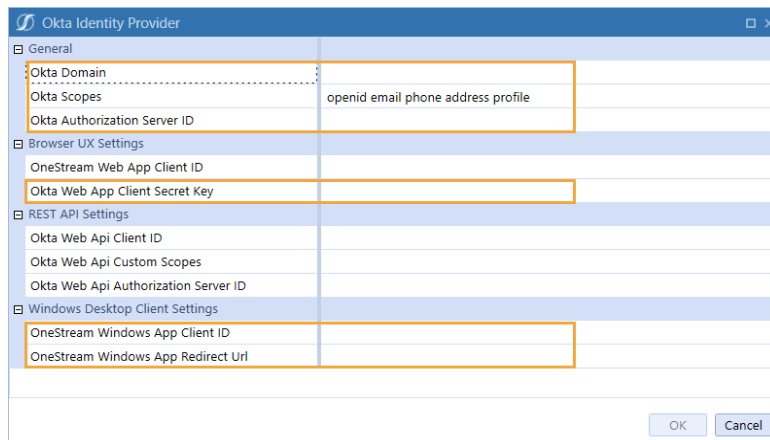
5. In the **Okta Identity Provider** dialog box, in the **General**, **Browser UX Settings**, and **Windows Desktop Client Settings** sections, complete the following fields:

    - **Okta Domain**: Enter the domain from the Okta developer page URL.

    - **Okta Scopes**: Enter scopes, or leave as default (openid email phone address profile).

    - **Okta Authorization Server ID**: Leave as default (null).

- **Okta Web App Client Secret Key**: Enter the same value from the External Provider Web SSO Secret Key field in the Application Server Configuration.

- **OneStream Windows App Client ID**: Enter the client ID from the Okta application. See Set Up the Application in Okta step 9.

- **OneStream Windows App Redirect Url**: Enter the value you entered in Okta as the sign-in redirect URI. See Set Up the Application in Okta step 7. Use this format: https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx

  > **IMPORTANT:** The sign-in redirect URI entered in Okta and in the OneStream Windows App Redirect Url field need to match exactly, including capitalization.

6. Click the **OK** button.

7. Save changes and reset IIS.

   > **NOTE:** Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.

Set Up OneStream Login with Okta

1. In the OneStream Desktop application, go to **System** > **Security** > **Users** > **<user>**.

2. In the **Authentication** properties, configure the user to authenticate through Okta.

- **External Authentication Provider**: In the drop-down menu, select the Okta configuration.

- **External Provider User Name**: Enter the username configured in Okta. This name must match the username set up in Okta and be used by only one user.

⊟ Authentication	
External Authentication Provider	OktaSSO
External Provider User Name	Username
Internal Provider Password	•••••

3. Click the **Save** icon.

Log into OneStream with Okta

1. On the OneStream Desktop Application Logon screen, for **Server Address**, specify the URL or a client connection.

2. Click the **Connect** button.

3. Click the **External Provider Sign In** button.

4. Enter your Okta login credentials.

5. Select an application from the drop-down menu.

6. Click the **Open Application** button.

> **NOTE:** To log off, on any screen, click the **Logoff** icon and then click the **End Session** button.

PingFederate Configuration

First, install and configure PingFederate. See Appendix: Installing and Configuring PingFederate.

To enable single sign-on with PingFederate for the OneStream Desktop application, which includes the Windows Client application and the Excel Add-In, using OIDC protocol, follow these steps:

1. [Set Up for Single Sign-on with an External Identity Provider](#) or [Set Up for Native Authentication and Single Sign-on with an External Identity Provider](#).

2. [Set Up the Application Server Configuration in OneStream](#).

3. [Set Up the Application in PingFederate](#).

4. [Set Up the Web Server Configuration in OneStream](#).

5. [Set Up OneStream Login with PingFederate](#).

6. [Log into OneStream with PingFederate](#).

To configure OneStream REST API to support PingFederate authentication, see the *REST API Implementation Guide*.

Set Up the Application Server Configuration in OneStream

1. Open the **OneStream Server Configuration Utility** application.

2. Go to **File** > **New Application Server Configuration File**.

   > **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Security** section, click the ellipsis to the right of **External Authentication Providers**.

4. Click the **+** icon to add an item.

5. In the **General** and **External Provider Single Sign On** sections, complete the following fields:

- **Name**: Enter the name of the identity provider. This name will display when configuring users to their external SSO.

- **Authentication Provider Type**: Select **PingFederateSSO** in the drop-down menu.

- **External Provider Web SSO Secret Key**: Enter a unique value. This key is used in the OneStream Application Server Configuration and the OneStream Web Server Configuration. It enables your application server to communicate with the web server.

6. Click the **OK** button.

7. Save changes and reset IIS.

> **NOTE:** Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.
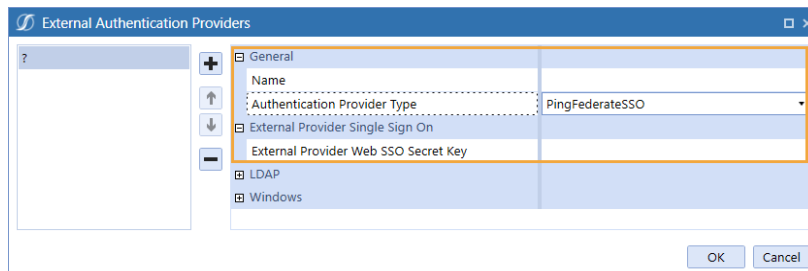
Set Up the Application in PingFederate

To set up the application in PingFederate, you must enter the same client ID in PingFederate and the Web Server Configuration in OneStream. Also, enter the redirect URI in PingFederate in this format: https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx

1. Log into your PingFederate account.

2. In the menu on the left, click **OAuth Server**.

3. Under the **CLIENTS** list, click the **Create New** button.

4. On the **Client** page, complete the following fields:

- **CLIENT ID**: Enter a client ID, which is a unique name or identifier for the application registration.

- **NAME**: Enter the name of the client.

- **REDIRECT URIS**: Enter the redirect URI in this format: https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx and then click the **Add** button.

- **ALLOWED GRANT TYPES**: Select **Authorization Code** and **Refresh Token**.

- **REQUIRE PROOF KEY FOR CODE EXCHANGE (PKCE)**: Select this option.

5. Click the **Save** button.

Set Up the Web Server Configuration in OneStream

1. Open the **OneStream Server Configuration Utility** application.

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

   > **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Web Server Configuration Settings** section, click the ellipsis to the right of **Single Sign On Identity Provider**.
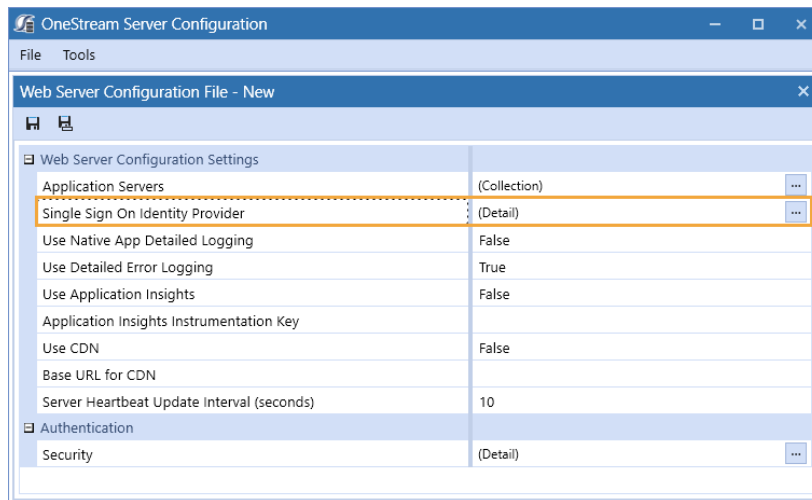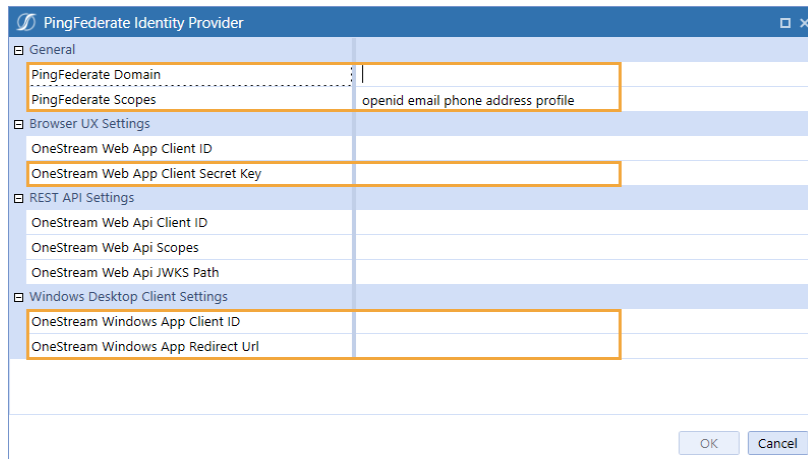
4. Click the ellipsis to the right of **PingFederate Identity Provider**.

5. In the **PingFederate Identity Provider** dialog box, in the **General**, **Browser UX Settings**, and **Windows Desktop Client Settings** sections, complete the following fields:

- **PingFederate Domain**: Enter the PingFederate domain. If needed, request this value from your IT Support.

- **PingFederate Scopes**: Enter scopes, or leave as default (openid email phone address profile).

- **OneStream Web App Client Secret Key**: Enter the same value from the External Provider Web SSO Secret Key field in the Application Server Configuration.

- **OneStream Windows App Client ID**: Enter the client ID you entered in PingFederate. See Set Up the Application in PingFederate step 4.

- **OneStream Windows App Redirect Url**: Enter the value you entered in PingFederate as the redirect URI. See Set Up the Application in PingFederate step 4. Use this format:
  https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx

  > **IMPORTANT:** The redirect URI entered in PingFederate and in the OneStream Windows App Redirect Url field need to match exactly, including capitalization.

6. Click the **OK** button.

7. Save changes and reset IIS.

    > **NOTE:** Reset IIS after you save any changes to the Application Server
    > Configuration or Web Server Configuration.

Set Up OneStream Login with PingFederate

1. In the OneStream Desktop application, go to **System** > **Security** > **Users** > **<user>**.

2. In the **Authentication** properties, configure the user to authenticate through PingFederate.

    - **External Authentication Provider**: In the drop-down menu, select the
      PingFederate configuration.

    - **External Provider User Name**: Enter the username configured in PingFederate.
      This name must match the username set up in PingFederate and be used by only
      one user.

3. Click the **Save** icon.

Log into OneStream with PingFederate

1. On the OneStream Desktop Application Logon screen, for **Server Address**, specify the URL or a client connection.

2. Click the **Connect** button.

3. Click the **External Provider Sign In** button.

4. Enter your PingFederate login credentials.

5. Select an application from the drop-down menu.

6. Click the **Open Application** button.

> **NOTE:** To log off, on any screen, click the **Logoff** icon and then click the **End Session** button.

SAML 2.0 Configuration with Okta

To enable single sign-on with Okta for the OneStream Desktop application, which includes the Windows Client application and the Excel Add-In, using SAML protocol, follow these steps:

1. Set Up for Single Sign-on with an External Identity Provider or Set Up for Native Authentication and Single Sign-on with an External Identity Provider.

2. Set Up the Application Server Configuration in OneStream.

3. Set Up the Application in Okta.

4. Set Up the Web Server Configuration in OneStream.

5. Set Up OneStream Login with Okta.

6. Log into OneStream with Okta.

Set Up the Application Server Configuration in OneStream

1. Open the **OneStream Server Configuration Utility** application.

2. Go to **File** > **New Application Server Configuration File**.

> **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Security** section, click the ellipsis to the right of **External Authentication Providers**.

4. Click the **+** icon to add an item.

5. In the **General** and **External Provider Single Sign On** sections, complete the following fields:

   - **Name**: Enter the name of the identity provider. This name will display when configuring users to their external SSO.

   - **Authentication Provider Type**: Select **SamlSSO** in the drop-down menu.

   - **External Provider Web SSO Secret Key**: Enter a unique value. This key is used in the OneStream Application Server Configuration and the OneStream Web Server Configuration. It enables your application server to communicate with the web server.

6. Click the **OK** button.

7. Save changes and reset IIS.

> **NOTE:** Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.

Set Up the Application in Okta

To set up the application in Okta, you must copy the single sign on (ACS) URL and metadata URL from Okta and paste them into the Web Server Configuration in OneStream. Also, enter the single sign-on URL and audience URI in Okta in this format: https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx/

> **IMPORTANT:** Follow the format listed for the single sign-on URL and audience URI exactly when you enter them in Okta. They must be an identical match.

1. Log into your Okta account.

2. In the **Applications** list on the left, select **Applications**.

3. Click **Create App Integration**.

4. In the **Create a new app integration** dialog box, for **Sign-in method**, select **SAML 2.0**.

5. Click the **Next** button.

6. On the **Create SAML Integration** page, on the **General Settings** tab, in the **App name** field, enter the name of the application in Okta.

7. Click the **Next** button.

8. On the **Create SAML Integration** page, on the **Configure SAML** tab, complete the following fields:

   - **Single sign-on URL**: Enter the single sign-on URL in this format: https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx/

     > **IMPORTANT:** Follow the format listed for the single sign-on URL exactly when you enter it in Okta. It must be an identical match.

   - **Audience URI (SP Entity ID)**: Enter the audience URI in this format: https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx/

     > **IMPORTANT:** Follow the format listed for the audience URI exactly when you enter it in Okta. It must be an identical match.

9. Click the **Next** button.

10. On the **Create SAML Integration** page, on the **Feedback** tab, for **Are you a customer or partner?**, select **I'm an Okta customer adding an internal app**.

11. Click the **Finish** button.

12. Copy the following information. You will need to paste this into the Web Server Configuration in OneStream.

    a. In the **General** tab, copy the single sign on URL.

    b. In the **Sign on** tab, copy the metadata URL.

    > **NOTE:** On the **Assignments** tab, you can assign users to the application.

Set Up the Web Server Configuration in OneStream

1. Open the **OneStream Server Configuration Utility** application.

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

    > **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Web Server Configuration Settings** section, click the ellipsis to the right of **Single Sign On Identity Provider**.

4. Click the ellipsis to the right of **SAML 2.0 Identity Provider**.

5. In the **SAML 2.0 Identity Provider** dialog box, in the **General** and **Windows Desktop Client Settings** sections, complete the following fields :

- **Application Server Pre-Shared Key**: Enter the same value from the External Provider Web SSO Secret Key field in the Application Server Configuration.

- **ACS URL for Windows Application**: Enter the single sign on URL from Okta . See Set Up the Application in Okta step 12. Use this format: https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx/

   > **IMPORTANT:** Follow the format listed for the single sign-on URL exactly when you enter it in the Web Server Configuration File. It must be identical match.

- **Metadata Content For Native Application**: Enter the metadata URL from Okta . See Set Up the Application in Okta step 12.

6. Click the **OK** button.

7. Save changes and reset IIS.

> **NOTE:** Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.

Set Up OneStream Login with Okta

1. In the OneStream Desktop application, go to **System** > **Security** > **Users** > **<user>**.

2. In the **Authentication** properties, configure the user to authenticate through Okta.

   - **External Authentication Provider**: In the drop-down menu, select the Okta configuration.

   - **External Provider User Name**: Enter the username configured in Okta. This name must match the username set up in Okta and be used by only one user.

3. Click the **Save** icon.

Log into OneStream with Okta

1. On the OneStream Desktop Application Logon screen, for **Server Address**, specify the URL or a client connection.

2. Click the **Connect** button.

3. Click the **External Provider Sign In** button.

4. Enter your Okta login credentials.

5. On the OneStream Desktop application Logon screen, select an application from the drop-down menu.

6. Click the **Open Application** button.

> **NOTE:** To log off, on any screen, click the **Logoff** icon and then click the **End Session** button.

SAML 2.0 Configuration with PingFederate

First, install and configure PingFederate. See Appendix: Installing and Configuring PingFederate.

To enable single sign-on with PingFederate for the OneStream Desktop application, which includes the Windows Client application and the Excel Add-In, using SAML protocol, follow these steps:

1. Set Up for Single Sign-on with an External Identity Provider or Set Up for Native Authentication and Single Sign-on with an External Identity Provider.

2. Set Up the Application Server Configuration in OneStream.

3. Set Up the Application in PingFederate.

4. Set Up the Web Server Configuration in OneStream.

5. Set Up OneStream Login with PingFederate.

6. Log into OneStream with PingFederate.

Set Up the Application Server Configuration in OneStream

1. Open the **OneStream Server Configuration Utility** application.

2. Go to **File** > **New Application Server Configuration File**.

   > **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Security** section, click the ellipsis to the right of **External Authentication Providers**.

4. Click the **+** icon to add an item.

5. In the **General** and **External Provider Single Sign On** sections, complete the following fields:

   - **Name**: Enter the name of the identity provider. This name will display when configuring users to their external SSO.

   - **Authentication Provider Type**: Select **SamlSSO** in the drop-down menu.

   - **External Provider Web SSO Secret Key**: Enter a unique value. This key is used in the OneStream Application Server Configuration and the OneStream Web Server Configuration. It enables your application server to communicate with the web server.

6. Click the **OK** button.

7. Save changes and reset IIS.

> **NOTE:** Reset IIS after you save any changes to the Application Server
> Configuration or Web Server Configuration.

Set Up the Application in PingFederate

To set up the application in PingFederate, you must export the metadata from PingFederate and
then copy the URL, entity ID, and single sign-on from the metadata and paste them into the Web
Server Configuration in OneStream. Also, enter the redirect URI and assertion consumer service
(ACS) URL in PingFederate in this format:
https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx/

> **IMPORTANT:** Follow the format listed for the redirect URI and ACS URL exactly when
> you enter them in PingFederate. They must be an identical match.

1. Log into your PingFederate account.

2. In the menu on the left, click **Identity Provider**.

3. Under the **SP CONNECTIONS** list, click the **Create New** button.

4. On the **SP Connection** page, in the **Connection Template** tab, select
   DO NOT USE A TEMPLATE FOR THIS CONNECTION, and click the **Next** button.

5. In the **Connection Type** tab, select **BROWSER SSO PROFILES**, and click the **Next**
   button.

6. In the **Connection Options** tab, click the **Next** button.

7. In the **Import Metadata** tab, click the **Next** button.

8. In the **General Info** tab, complete the following fields:

   - **PARTNER'S ENTITY ID (CONNECTION ID)**: Enter the redirect URI in this format:
     https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx/

     > **IMPORTANT:** Follow the format listed for the redirect URI exactly when you
     > enter it in PingFederate. It must be an identical match.

   - **CONNECTION NAME**: Enter the name of the connection.

9. Click the **Next** button.

10. In the **Browser SSO** tab, click the **Configure Browser SSO** button.

    a. In the **SAML Profiles** tab, in the **Single Sign-On (SSO) Profiles** list, select **SP-INITIATED SSO**. Click the **Next** button.

    b. In the **Assertion Lifetime** tab, click the **Next** button.

    c. In the **Assertion Creation** tab, click the **Configure Assertion Creation** button.

       a. In the **Identity Mapping** tab, click the **Next** button.

       b. In the **Attribute Contract** tab, click the **Next** button.

       c. In the **Authentication Source Mapping** tab, click the **Map New Adapter Instance** button.

          a. In the **Adapter Instance** tab, for **ADAPTER INSTANCE**, select **HTML Form Adapter** from the drop-down menu. Click the **Next** button.

          b. In the **Mapping Method** tab, click the **Next** button.

          c. In the **Attribute Contract Fulfillment** tab, for **SAML_SUBJECT**, in the **Source** drop-down menu, select **Adapter**. In the **Value** drop-down menu, select **username**. Click the **Next** button.

          d. In the **Insurance Criteria** tab, click the **Next** button.

          e. In the **Summary** tab, click the **Done** button.

       d. In the **Summary** tab, click the **Next** button and then the **Done** button.

    d. In the **Assertion Creation** tab, click the **Next** button.

    e. In the **Protocol Settings** tab, click the **Configure Protocol Settings** button.

       a. In the **Assertion Consumer Service URL** tab, In the **Binding** drop-down menu, select **POST**. Enter the **Endpoint URL** in this format: https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx/ and click the **Add** button. Click the **Next** button.

> **IMPORTANT:** Follow the format listed for the ACS URL exactly when you enter it in PingFederate. It must be an identical match.
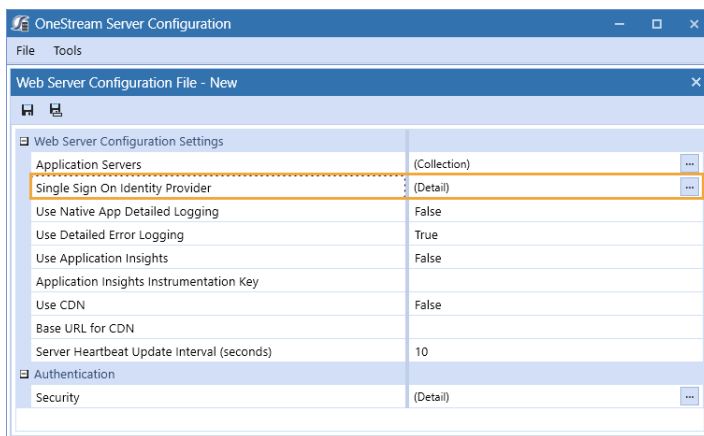
    b. In the **Allowable SAML Bindings** tab, clear the check boxes for **ARTIFACT** and **SOAP**. Select only **POST** and **REDIRECT**, and click the **Next** button.

    c. In the **Signature Policy** tab, click the **Next** button.

    d. In the **Encryption Policy** tab, click the **Next** button.

    e. In the **Summary** tab, click the **Done** button.

    f. In the **Summary** tab, click the **Next** button and then the **Done** button.

11. In the **Browser SSO** tab, click the **Next** button.

12. In the **Credentials** tab, click the **Configure Credentials** button.

    a. In the **Digital Signature Settings** tab, in the **SIGNING CERTIFICATE** drop-down menu, select the certificate. Click the **Next** button.

    b. In the **Summary** tab, click the **Done** button.

13. In the **Activation & Summary** tab, click the **Next** button and then the **Save** button.

14. Export the metadata to add to the Web Server Configuration File.

    a. In PingFederate, in the menu on the left, click **Identity Provider**.

    b. Under the **SP CONNECTIONS** list, click the **Manage All** button.

    c. On the **SP Connections** page, in the row for the connection, in the **Action** column, click **Select Action** to view the drop-down menu options.

    d. Select **Export Metadata**.

    e. Select the signing certificate from the drop-down menu, and click the **Next** button.

    f. Click the **Export** button.

Set Up the Web Server Configuration in OneStream

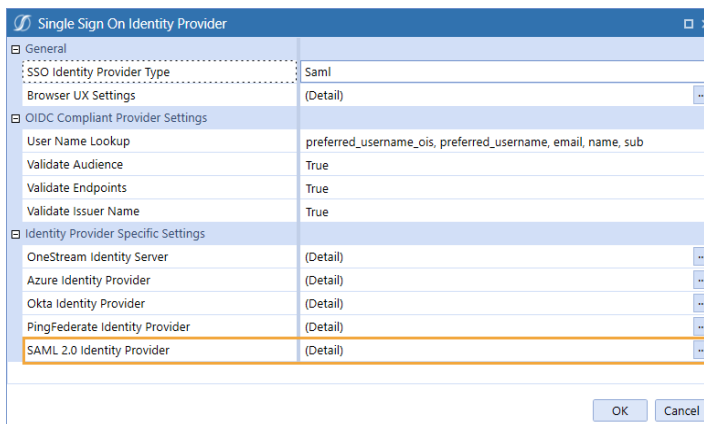1. Open the **OneStream Server Configuration Utility** application.

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

   > **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Web Server Configuration Settings** section, click the ellipsis to the right of **Single Sign On Identity Provider**.

4. Click the ellipsis to the right of **SAML 2.0 Identity Provider**.

5. In the **SAML 2.0 Identity Provider** dialog box, in the **General** and **Windows Desktop Client Settings** sections, complete the following fields:

   - **Application Server Pre-Shared Key**: Enter the same value from the External Provider Web SSO Secret Key field in the Application Server Configuration.

   - **ACS URL for Windows Application**: Enter the assertion consumer service (ACS) URL from PingFederate. See Set Up the Application in PingFederate step 10. Use this format:
     https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx/

     > **IMPORTANT:** Follow the format listed for the ACS URL exactly when you enter it in the Web Server Configuration File. It must be an identical match.

   - **Metadata Content For Native Application**: Enter the metadata URL from PingFederate. See Set Up the Application in PingFederate step 14.

   - **IdP Entity ID for Native Applications**: Enter the entity ID from the metadata. See Set Up the Application in PingFederate step 14.

   - **IdP Single Sign-On URL for Native ApplicationsACS**: Enter the single sign-on from the metadata. See Set Up the Application in PingFederate step 14.

6. Click the **OK** button.

7. Save changes and reset IIS.

> **NOTE:** Reset IIS after you save any changes to the Application Server
> Configuration or Web Server Configuration.

Set Up OneStream Login with PingFederate

1. In the OneStream Desktop application, go to **System** > **Security** > **Users** > **<user>**.

2. In the **Authentication** properties, configure the user to authenticate through PingFederate.

    - **External Authentication Provider**: In the drop-down menu, select the
      PingFederate configuration.

    - **External Provider User Name**: Enter the username configured in PingFederate.
      This name must match the username set in PingFederate and be used by only one
      user.

⊟ Authentication	
External Authentication Provider	PingFed SAML
External Provider User Name	Username
Internal Provider Password	••••••

3. Click the **Save** icon.

Log into OneStream with PingFederate

1. On the OneStream Desktop Application Logon screen, for **Server Address**, specify the
   URL or a client connection.

2. Click the **Connect** button.

3. Click the **External Provider Sign In** button.

4. Enter your PingFederate login credentials.

5. On the OneStream Desktop application Logon screen, select an application from the drop-
   down menu.

6. Click the **Open Application** button.

> **NOTE:** To log off, on any screen, click the **Logoff** icon and then click the **End
> Session**button.

SAML 2.0 Configuration with ADFS

To enable single sign-on with ADFS for the OneStream Desktop application, which includes the Windows Client application and the Excel Add-In, using SAML protocol, follow these steps:

1. Set Up for Single Sign-on with an External Identity Provider or Set Up for Native Authentication and Single Sign-on with an External Identity Provider.

2. Set Up the Application Server Configuration in OneStream.

3. Set Up the Application in ADFS.

4. Set Up the Web Server Configuration in OneStream.

5. Set Up OneStream Login with ADFS.

6. Log into OneStream with ADFS.

Set Up the Application Server Configuration in OneStream

1. Open the **OneStream Server Configuration Utility** application.

2. Go to **File** > **New Application Server Configuration File**.

   > **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Security** section, click the ellipsis to the right of **External Authentication Providers**.

4. Click the **+** icon to add an item.

5. In the **General** and **External Provider Single Sign On** sections, complete the following fields:

- **Name**: Enter the name of the identity provider. This name will display when configuring users to their external SSO.

- **Authentication Provider Type**: Select **SamISSO** in the drop-down menu.

- **External Provider Web SSO Secret Key**: Enter a unique value. This key is used in the OneStream Application Server Configuration and the OneStream Web Server Configuration. It enables your application server to communicate with the web server.

6. Click the **OK** button.

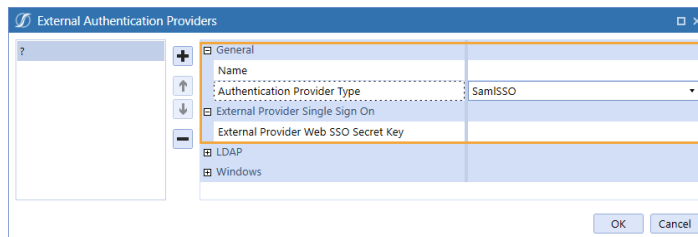7. Save changes and reset IIS.

> **NOTE:** Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.

Set Up the Application in ADFS

To set up the application in ADFS, you must copy the ACS URL and metadata URL from ADFS and paste them into the Web Server Configuration in OneStream. Also, enter the relying party SAML 2.0 SSO service URL and relying party trust identifier in ADFS in this format: https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx/

> **IMPORTANT:** Follow the format listed for the relying party SAML 2.0 SSO service URL and relying party trust identifier exactly when you enter them in ADFS. They must be an identical match.

1. Log into your ADFS account.

2. On the left of the screen, open the **AD FS** folder, and then open the **Relying Party Trusts** folder.

3. In the **Actions** pane on the right, in the **Relying Party Trusts** menu, select **Add Relying Party Trust**.

4. In the **Add Relying Party Trust Wizard** dialog box, on the **Welcome** screen, select **Claims Aware** and click the **Start** button.

5. On the **Select Data Source** screen, select **Enter data about the relying party manually** and click the **Next** button.

6. On the **Specify Display Name** screen, enter a name for the application and click the **Next** button.

7. On the **Configure Certificate** screen, click the **Next** button.

8. On the **Configure URL** screen:

   a. Select **Enable support for the SAML 2.0 WebSSO protocol**.

   b. Enter the relying party SAML 2.0 SSO service URL in this format:
      https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx/

      > **IMPORTANT:** Follow the format listed for the relying party SAML 2.0 SSO service URL exactly when you enter it in ADFS. It must be an identical match.

   c. Click the **Next** button.

9. On the **Configure Identifiers** screen:

   a. Enter the relying party trust identifier in this format:
      https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx/

      > **IMPORTANT:** Follow the format listed for the relying party trust identifier exactly when you enter it in ADFS. It must be identical match.

   b. Click the **Next** button.

10. On the **Choose Access Control Policy** screen, select an option and click the **Next** button.

11. On the **Ready to Add Trust** screen, click the **Next** button.

12. On the **Finish** screen, click the **Close** button.

13. In the **Relying Party Trusts** list, right-click on the item you created, and click **Edit Claim Issuance Policy**.

14. In the **Edit Claims Issuance Policy** dialog box, click the **Add Rule** button.

15. In the **Add Transform Claim Rule Wizard** dialog box, on the **Select Rule Template** screen, in the **Claim rule template** drop-down menu, select **Send LDAP Attributes as Claims** and click the **Next** button.

16. On the **Configure Rule** screen, in the Claim rule name field, type **Attributes**. In the **Attribute store** drop-down menu, select **Active Directory**. In the Mapping of LDAP attributes to outgoing claim types table of drop-down menus, select these options:

    a. **LDAP Attribute**: **Given-Name** and **Outgoing Claim Type**: **Given Name**

    b. **LDAP Attribute**: **Surname** and **Outgoing Claim Type**: **Surname**

    c. **LDAP Attribute**: **E-Mail-Addresses** and **Outgoing Claim Type**: **E-Mail Address**

17. Click the **Finish** button.

18. In the **Edit Claims Issuance Policy** dialog box, click the **Add Rule** button.

19. In the **Add Transform Claim Rule Wizard** dialog box, on the **Select Rule Template** screen, in the **Claim rule template** drop-down menu, select **Send LDAP Attributes as Claims** and click the **Next** button.

20. On the **Configure Rule** screen, in the Claim rule name field, type **Windows Account Name**. In the **Attribute store** drop-down menu, select **Active Directory**. In the Mapping of LDAP attributes to outgoing claim types table of drop-down menus, select these options:

    a. **LDAP Attribute**: **SAM-Account-Name** and **Outgoing Claim Type**: **Name ID**

21. Click the **Finish** button.

22. In the **Edit Claims Issuance Policy** dialog box, click the **Add Rule** button.

23. In the **Add Transform Claim Rule Wizard** dialog box, on the **Select Rule Template** screen, in the **Claim rule template** drop-down menu, select **Send LDAP Attributes as Claims** and click the **Next** button.
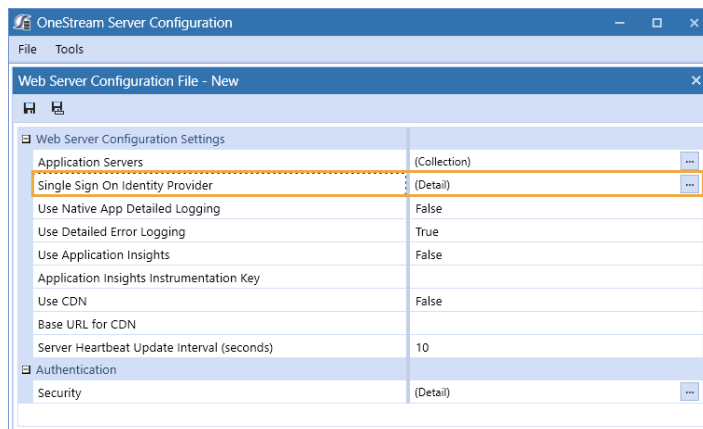
24. On the **Configure Rule** screen, in the Claim rule name field, type **UpnToAppId**. In the **Attribute store** drop-down menu, select **Active Directory**. In the Mapping of LDAP attributes to outgoing claim types table of drop-down menus, select these options:

    a. **LDAP Attribute**: **User-Principal-Name** and **Outgoing Claim Type**: **Application Identifier**

25. Click the **Finish** button.

26. In the **Edit Claims Issuance Policy** dialog box, click the **OK** button.

Set Up the Web Server Configuration in OneStream

1. Open the **OneStream Server Configuration Utility** application.

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

   > **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Web Server Configuration Settings** section, click the ellipsis to the right of **Single Sign On Identity Provider**.

4. Click the ellipsis to the right of **SAML 2.0 Identity Provider**.

5. In the **SAML 2.0 Identity Provider** dialog box, in the **General** and **Windows Desktop Client Settings** sections, complete the following fields :

- **Application Server Pre-Shared Key**: Enter the same value from the External Provider Web SSO Secret Key field in the Application Server Configuration.

- **ACS URL for Windows Application**: Enter the relying party SAML 2.0 SSO service URL from ADFS. See Set Up the Application in ADFS step 9. Use this format: https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx/

> **IMPORTANT:** Follow the format listed for the relying party SAML 2.0 SSO service URL exactly when you enter it in the Web Server Configuration File. It must be an identical match.

- **Metadata Content For Native Application**: Enter the metadata URL from ADFS. To find the metadata URL, in ADFS, go to **AD FS** > **Service** > **Endpoints** > **Metadata**. Copy the URL path for the Federation Metadata. Paste that URL path to end of the URL to the ADFS VM.

6. Click the **OK** button.

7. Save changes and reset IIS.

> **NOTE:** Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.

Set Up OneStream Login with ADFS

1. In the OneStream Desktop application, go to **System** > **Security** > **Users** > **<user>**.

2. In the **Authentication** properties, configure the user to authenticate through ADFS.

   - **External Authentication Provider**: In the drop-down menu, select the ADFS configuration.

   - **External Provider User Name**: Enter the username configured in ADFS. This name must match the username set up in ADFS and be used by only one user.

3. Click the **Save** icon.

Log into OneStream with ADFS

1. On the OneStream Desktop Application Logon screen, for **Server Address**, specify the URL or a client connection.

2. Click the **Connect** button.

3. Click the **External Provider Sign In** button.

4. Enter your ADFS login credentials.

5. On the OneStream Desktop application Logon screen, select an application from the drop-down menu.

6. Click the **Open Application** button.

> **NOTE:** To log off, on any screen, click the **Logoff** icon and then click the **End Session** button.

SAML 2.0 Configuration with Salesforce

To enable single sign-on with Salesforce for the OneStream Desktop application, which includes the Windows Client application and the Excel Add-In, using SAML protocol, follow these steps:

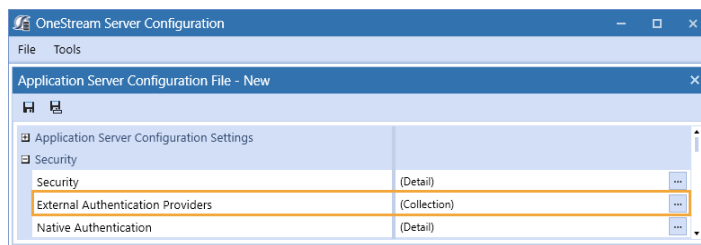1. [Set Up for Single Sign-on with an External Identity Provider](#) or [Set Up for Native Authentication and Single Sign-on with an External Identity Provider](#).

2. [Set Up the Application Server Configuration in OneStream](#).

3. [Set Up the Application in Salesforce](#).

4. [Set Up the Web Server Configuration in OneStream](#).

5. [Set Up OneStream Login with Salesforce](#).

6. [Log into OneStream with Salesforce](#).

Set Up the Application Server Configuration in OneStream

1. Open the **OneStream Server Configuration Utility** application.

2. Go to **File** > **New Application Server Configuration File**.

   > **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Security** section, click the ellipsis to the right of **External Authentication Providers**.

4. Click the **+** icon to add an item.

5. In the **General** and **External Provider Single Sign On** sections, complete the following fields:

   - **Name**: Enter the name of the identity provider. This name will display when configuring users to their external SSO.

   - **Authentication Provider Type**: Select **SamlSSO** in the drop-down menu.

   - **External Provider Web SSO Secret Key**: Enter a unique value. This key is used in the OneStream Application Server Configuration and the OneStream Web Server Configuration. It enables your application server to communicate with the web server.

6. Click the **OK** button.

7. Save changes and reset IIS.

> **NOTE:** Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.

Set Up the Application in Salesforce

To set up the application in Salesforce, you must copy the ACS URL and metadata discovery endpoint from Salesforce and paste them into the Web Server Configuration in OneStream. Also, enter the Entity Id and ACS URL in Salesforce in this format: https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx/

> **IMPORTANT:** Follow the format listed for the Entity Id and ACS URL exactly when you enter them in Salesforce. They must be an identical match.

1. Log into your Salesforce account.

2. On the left of the screen, under **PLATFORM TOOLS**, in the **Apps** list, select **App Manager**.

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

4. On the **New Connected App** page, in the **Basic Information** and **Web App Settings** sections, complete the following fields:

   - **Connected App Name**: Enter a name for the application.

   - **API Name**: Enter an API name.

   - **Contact Email**: Enter an email address.

   - **Enable SAML**: Select this option.

   - **Entity Id**: Enter Entity Id in this format:
     https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx/

     > **IMPORTANT:** Follow the format listed for the Entity Id exactly when you enter it in Salesforce. It must be an identical match.

- **ACS URL**: Enter the ACS URL in this format:
  https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx/

> **IMPORTANT:** Follow the format listed for the ACS URL exactly when you enter it in Salesforce. It must be an identical match.

5. Click the **Save** button.

> **NOTE:** On the left of the screen, go to **PLATFORM TOOLS** > **Apps** > **Connected Apps** > **Manage Connected Apps** to view the Entity ID and ACS URL.

Set Up the Web Server Configuration in OneStream

1. Open the **OneStream Server Configuration Utility** application.

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

> **NOTE:** Alternatively, you can open an existing file to edit it.

3. In the **Web Server Configuration Settings** section, click the ellipsis to the right of **Single Sign On Identity Provider**.

4. Click the ellipsis to the right of **SAML 2.0 Identity Provider**.

5. In the **SAML 2.0 Identity Provider** dialog box, in the **General** and **Windows Desktop Client Settings** sections, complete the following fields :

- **Application Server Pre-Shared Key**: Enter the same value from the External Provider Web SSO Secret Key field in the Application Server Configuration.

- **ACS URL for Windows Application**: Enter the ACS URL from Salesforce. See Set Up the Application in Salesforce step 4. Use this format: https://<domainname>/OneStreamWeb/OneStreamLogonCallback.aspx/

  > **IMPORTANT:** Follow the format listed for the ACS URL exactly when you enter it in the Web Server Configuration File. It must be an identical match.
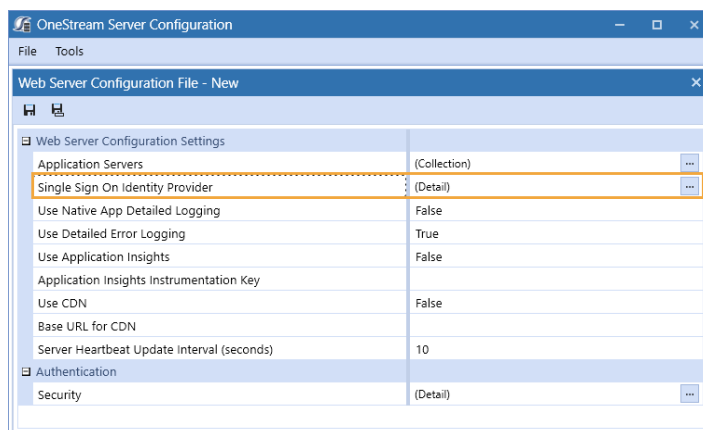
  > **TIP:** In Salesforce, on the left of the screen, go to **PLATFORM TOOLS** > **Apps** > **Connected Apps** > **Manage Connected Apps** to view the ACS URL.

- **Metadata Content For Native Application**: Enter the metadata discovery endpoint from Salesforce.

  > **TIP:** In Salesforce, go to **PLATFORM TOOLS** > **Apps** > **Connected Apps** > **Manage Connected Apps** to view the metadata discovery endpoint.

6. Click the **OK** button.

7. Save changes and reset IIS.

> **NOTE:** Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.

Set Up OneStream Login with Salesforce

1. In the OneStream Desktop application, go to **System** > **Security** > **Users** > **<user>**.

2. In the **Authentication** properties, configure the user to authenticate through Salesforce.

   - **External Authentication Provider**: In the drop-down menu, select the Salesforce configuration.

   - **External Provider User Name**: Enter the username configured in Salesforce. This name must match the username set up in Salesforce and be used by only one user.

3. Click the **Save** icon.

Log into OneStream with Salesforce

1. On the OneStream Desktop Application Logon screen, for **Server Address**, specify the URL or a client connection.

2. Click the **Connect** button.

3. Click the **External Provider Sign In** button.

4. Enter your Salesforce login credentials.

5. On the OneStream Desktop application Logon screen, select an application from the drop-down menu.

6. Click the **Open Application** button.

> **NOTE:** To log off, on any screen, click the **Logoff** icon and then click the **End Session** button.

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

Password: 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
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:
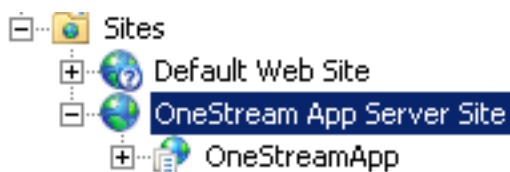
2. Click the OneStream App Server Site and choose **Advanced Settings** on the right side. In the pop-up window, confirm the Physical Path is correct and, if not, update it accordingly.

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

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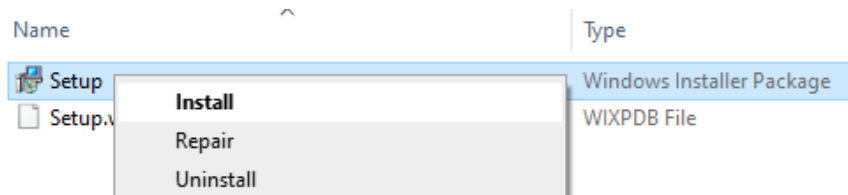
2. Click **Next** to continue the installation operation.
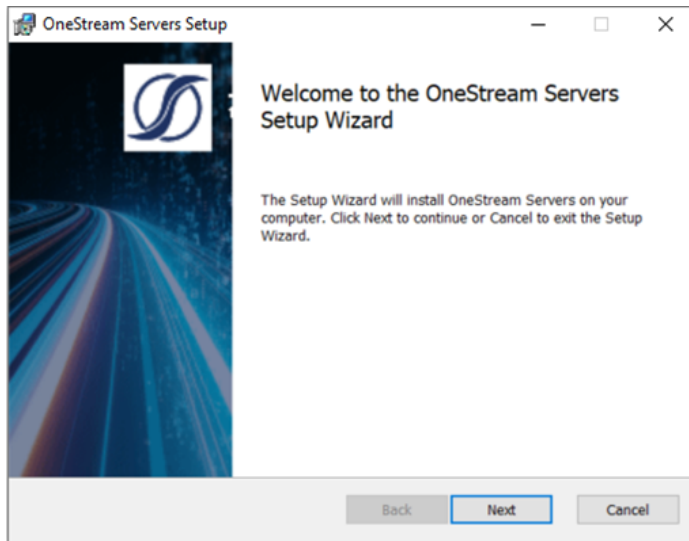
3. Accept the terms of the license agreement and click **Next**.
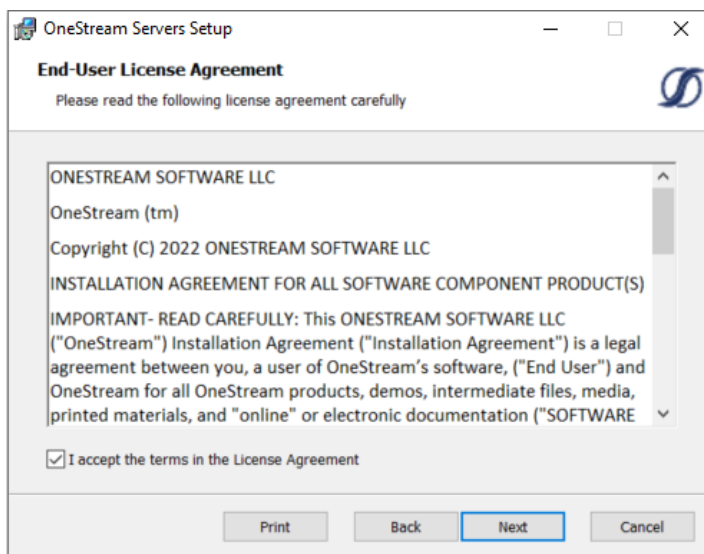
4. Select the **Custom** setup type and click **Next**.
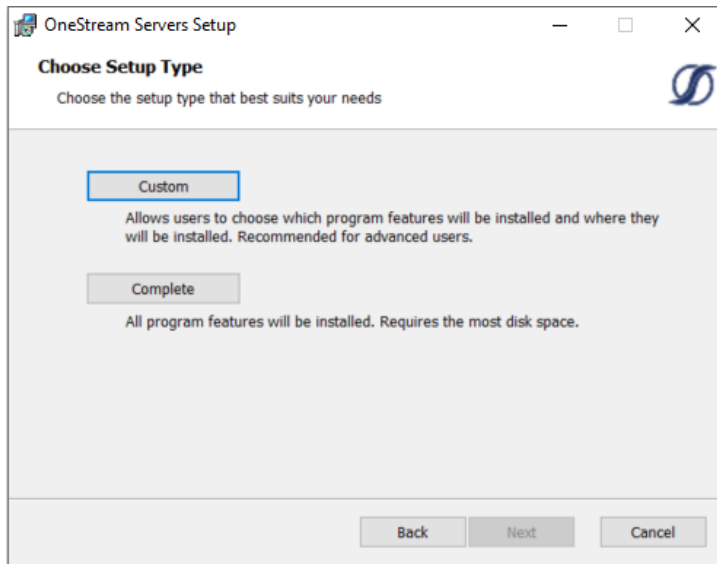
5. Install the Application Server only, click **Application Server** and choose Will be installed on local hard drive. Repeat for Server Configuration Utility. Click **Next**.

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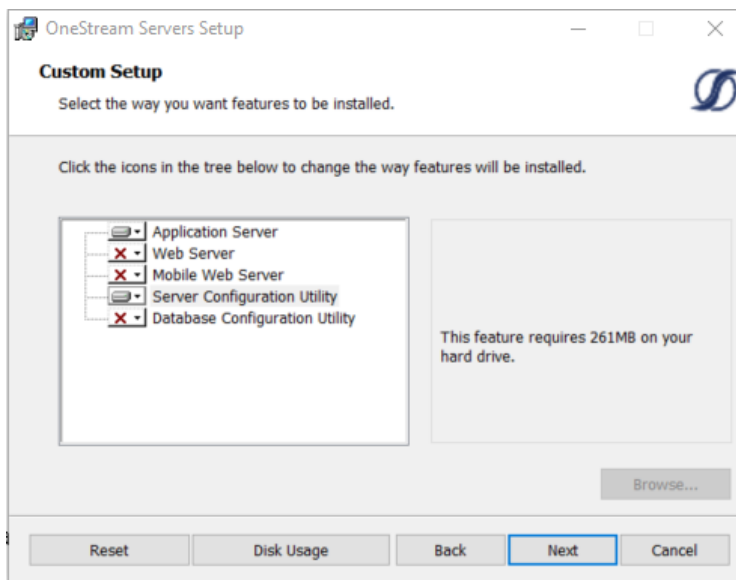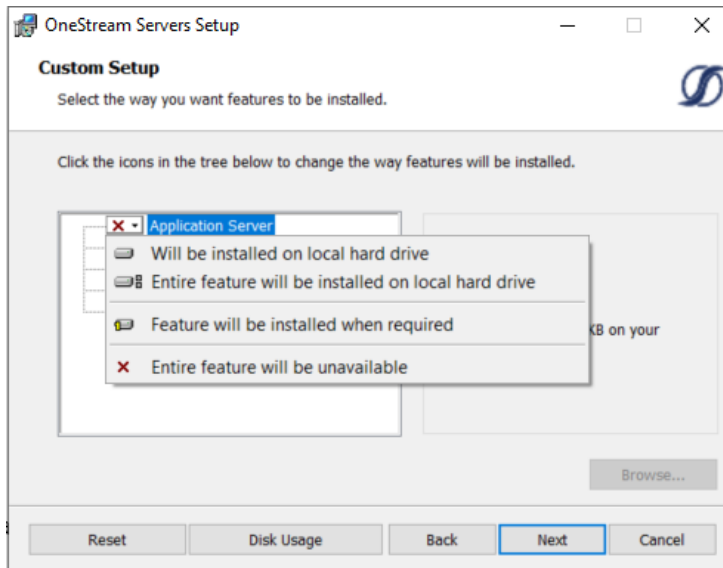
6. If needed, change the folder path and click Next.
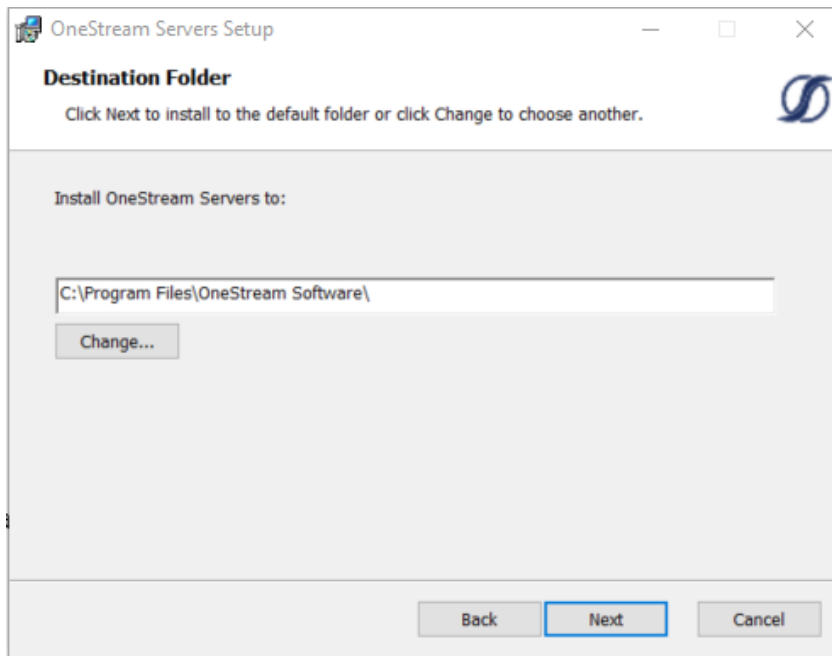
7. Click Install to begin the installation.
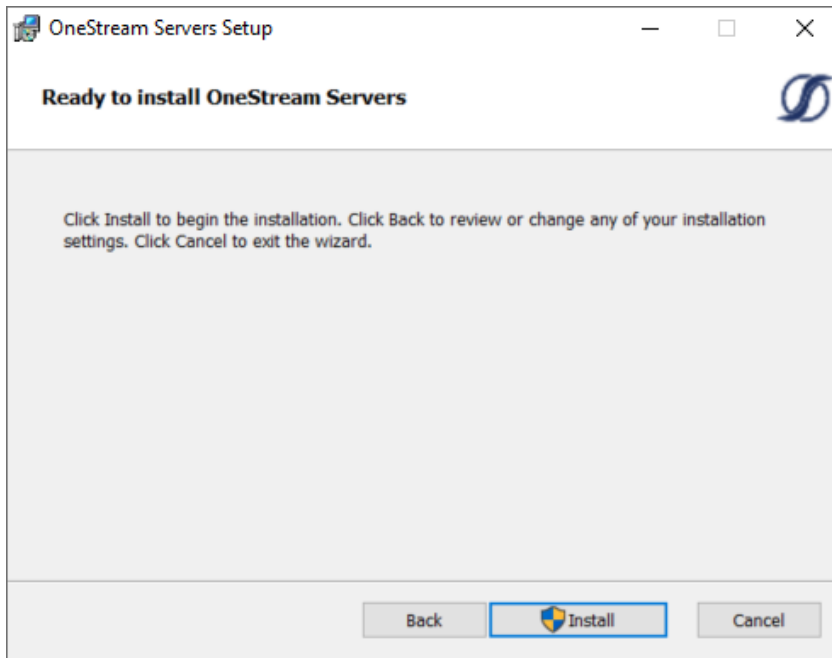
8. Click **Finish** to complete the installation.
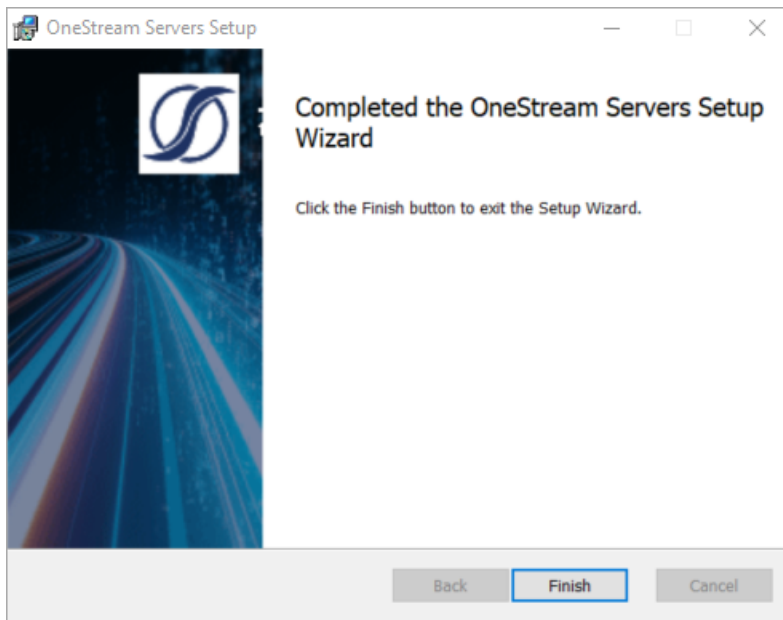
Configuring the Application Server

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

> **TIP:** Create a central location, C:\OneStreamShare\Config for example, accessible for all server configurations. Save the **XFAppServerConfig.xml** and **XFWebServerConfig.xml** there.

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**.
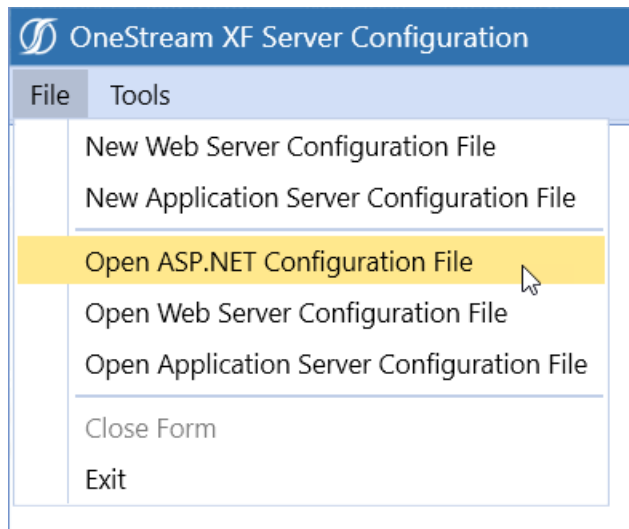
    b. Browse to the following location: <installdrive>\Program Files\OneStream Software\OneStreamAppRoot\OneStreamApp, select the **appsettings.json** file and click **Open**
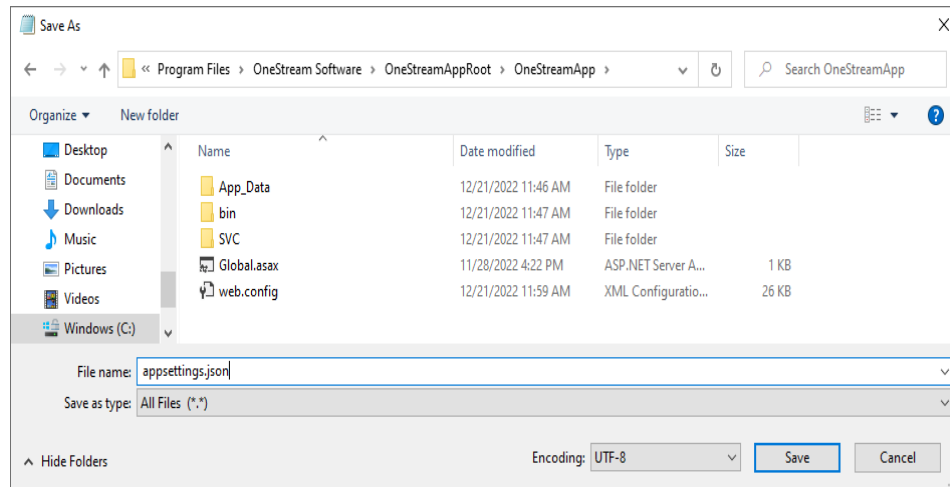
c. Update the path to the file location of the shared OneStream Application Server Configuration File (XFAppServerConfig.xml). For example, C:\OneStreamShare\Config.
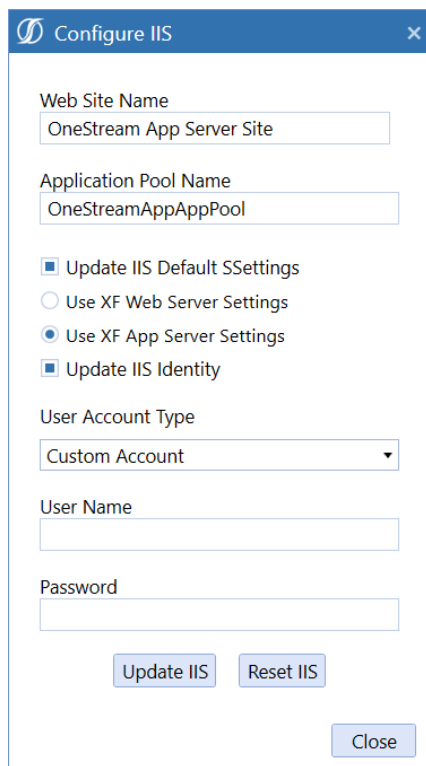
> **TIP:**
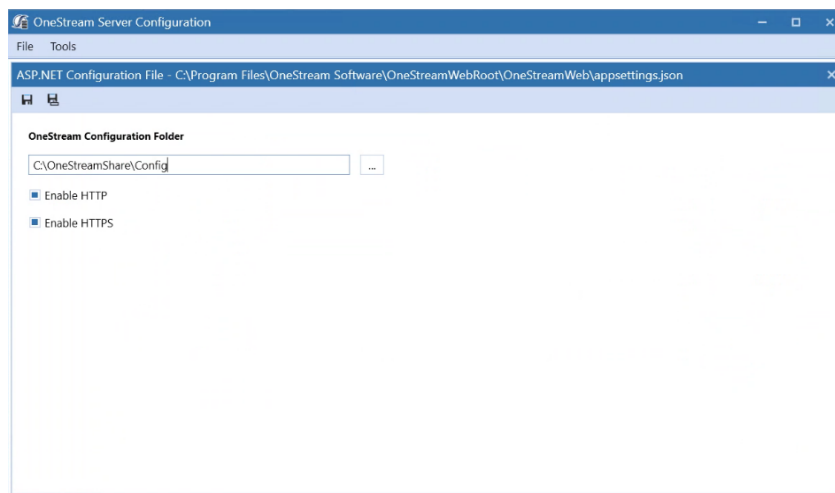
d. Enable HTTP and/or HTTPS as needed

e. Save the configuration file.

Update the Application Server IIS Settings using Configure IIS Tool

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

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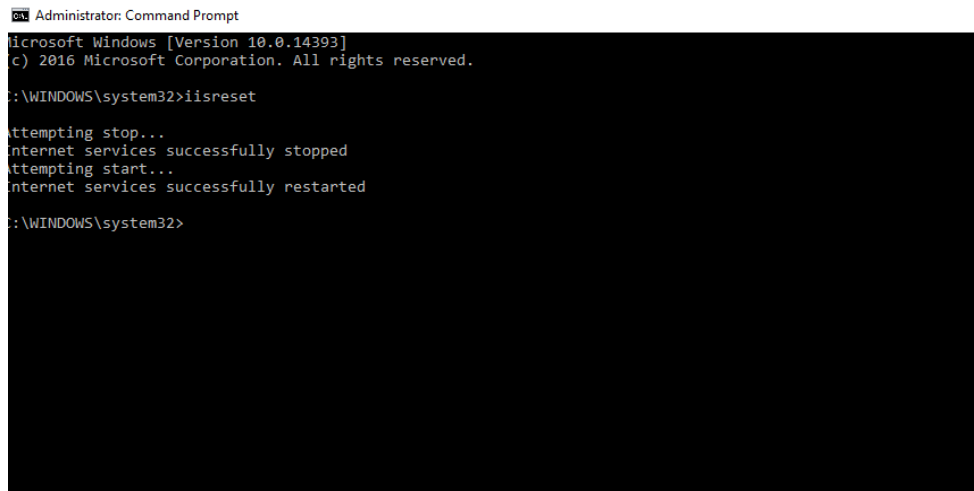
   - Web Site Name: OneStream App Server Site

   - 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
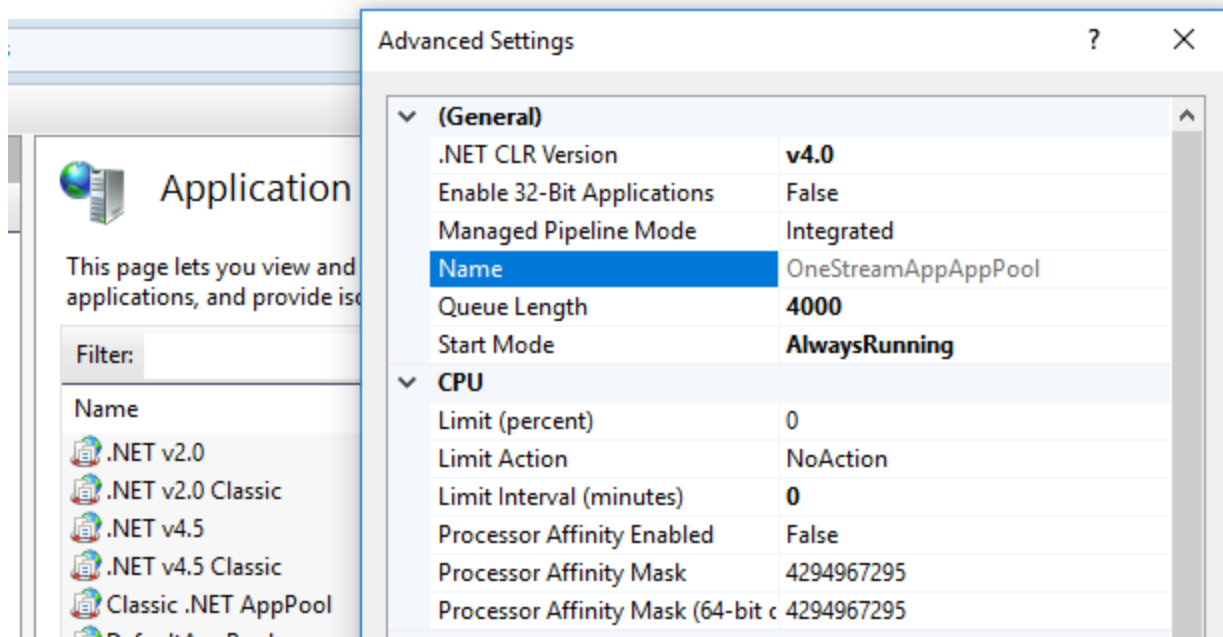
PreLoad Enabled: True on the OneStream App Server Site
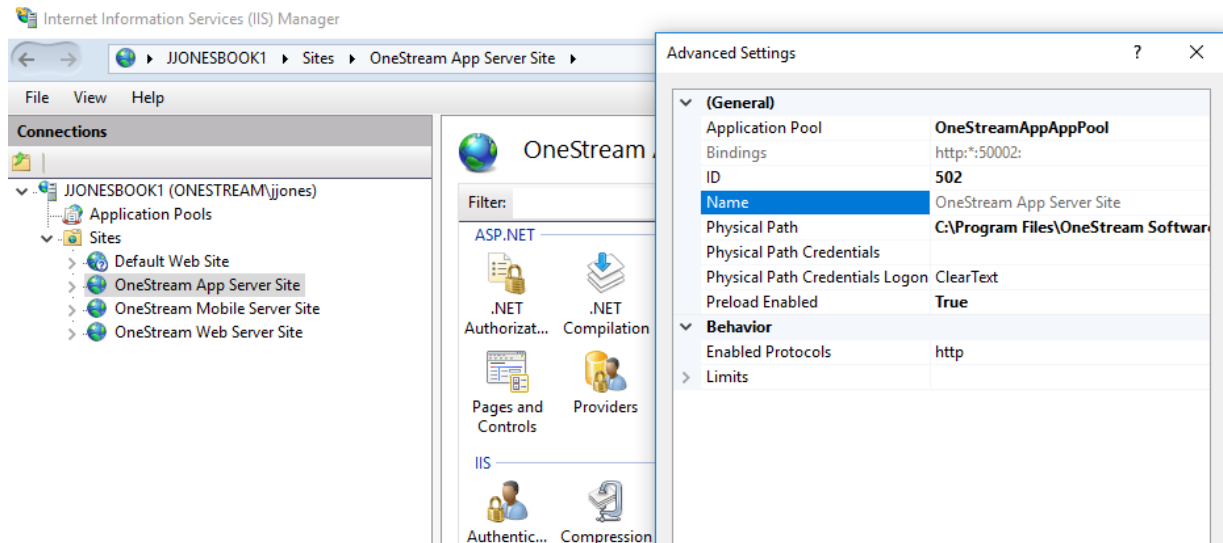
Preload Enabled True on the OneStreamApp registration:
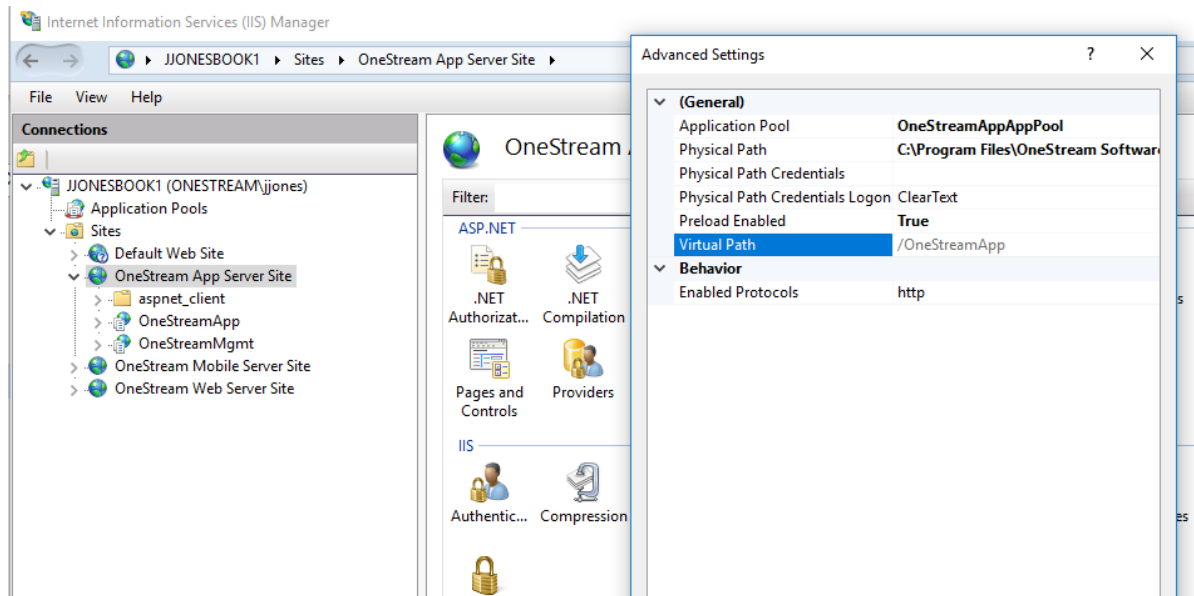
Installing the Web Server

> **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.
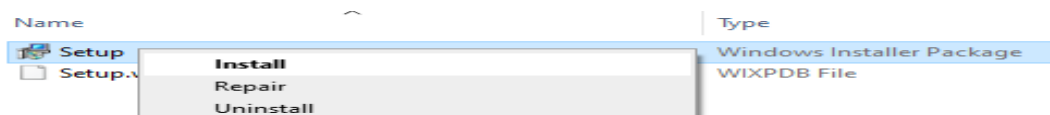
2. Click **Next** to continue with the installation operation.
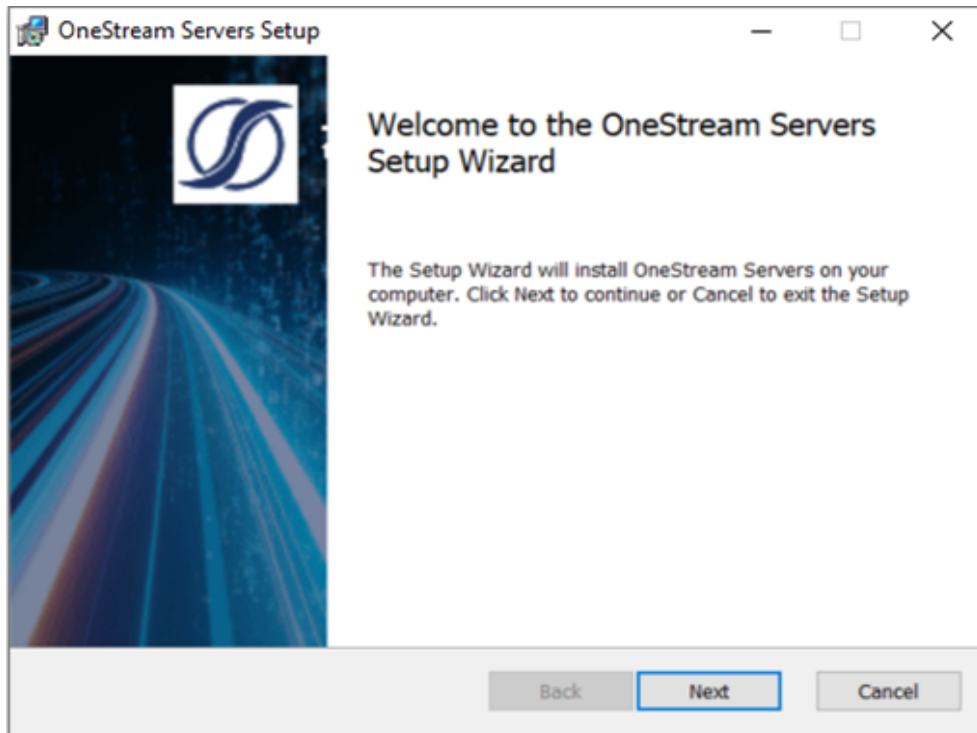
3. Accept the terms of the license agreement and click **Next**.
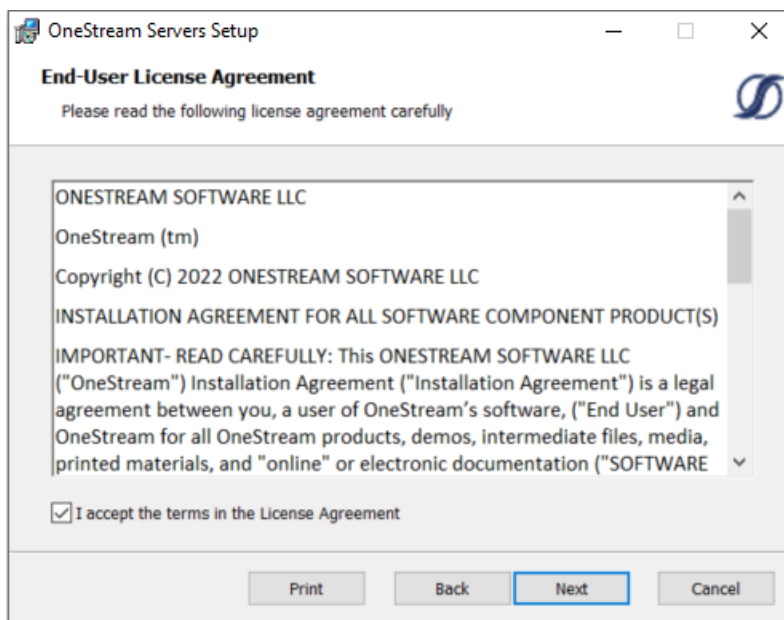
4. Select **Custom** as the setup type.
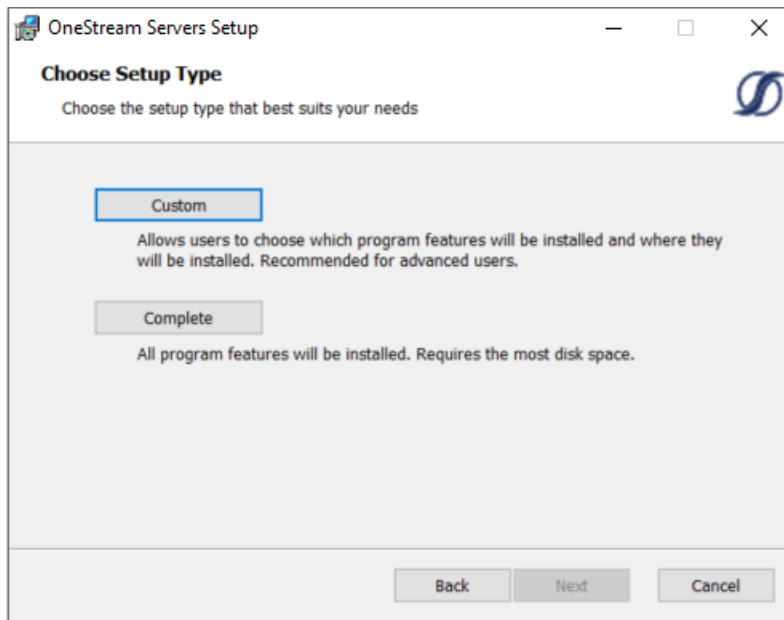
5. To install the Web Server only, click **Web Server** and choose **Will be installed on local hard drive**. Repeat for **Server Configuration Utility**. Click **Next**.
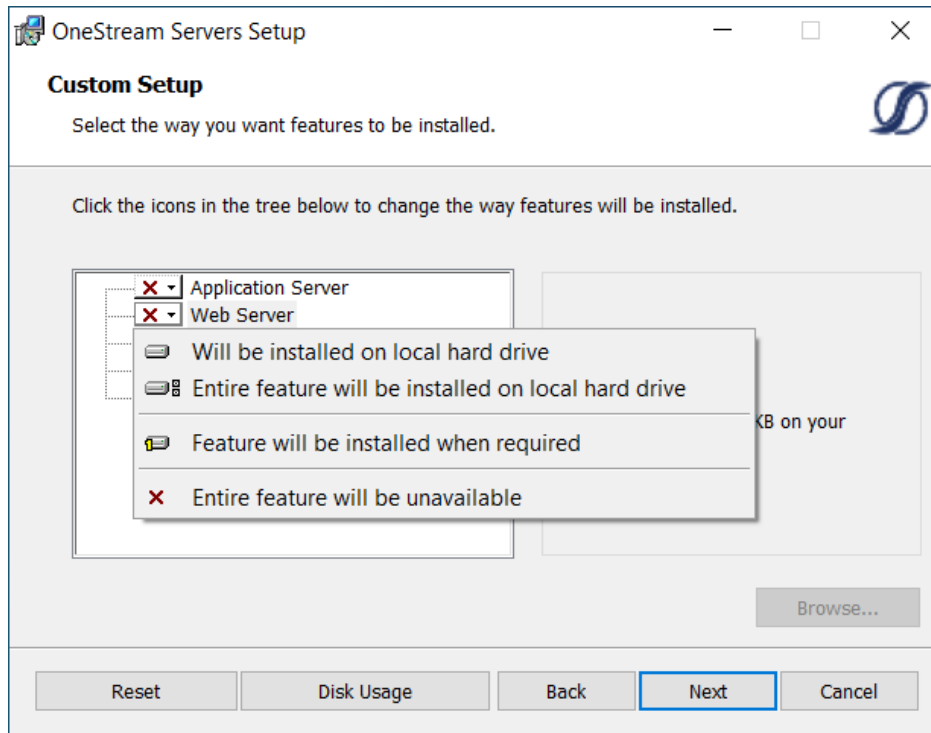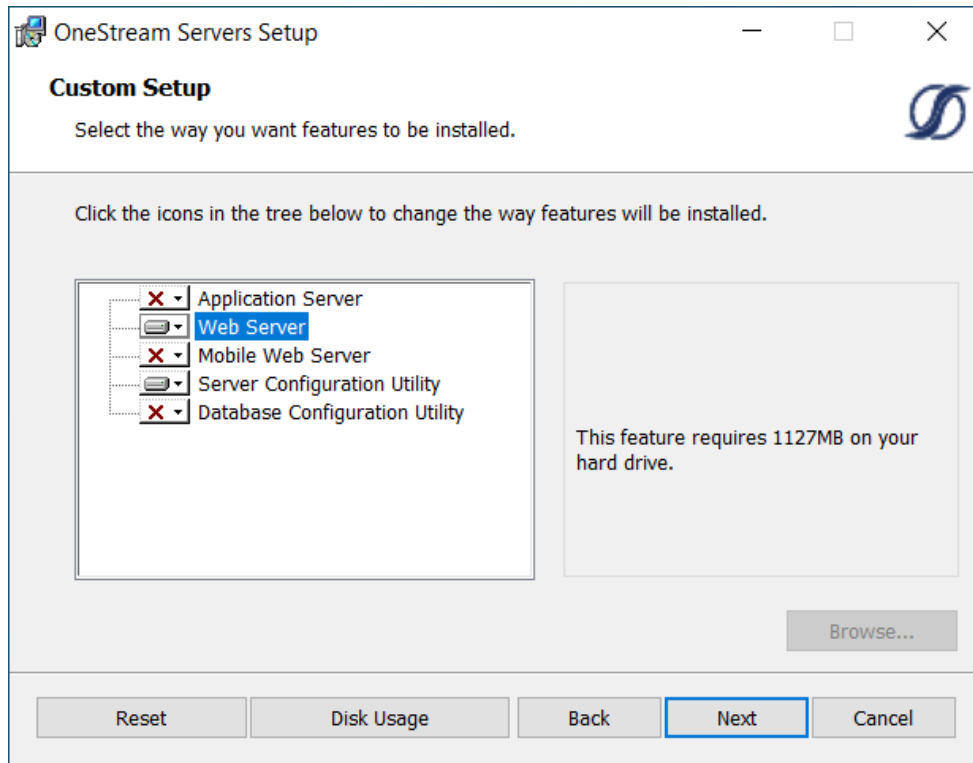
6. Change the folder path, if needed, and click **Next**.

> **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.
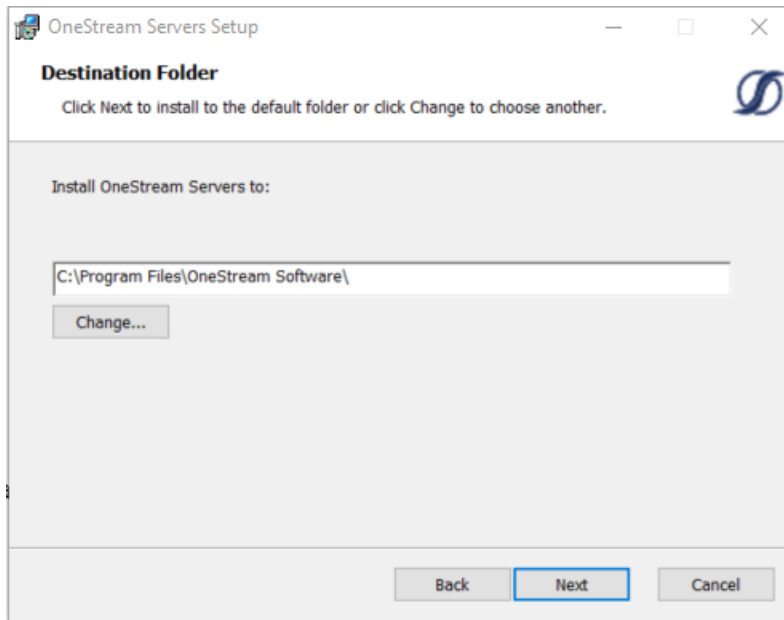
7. Click **Install** to start.
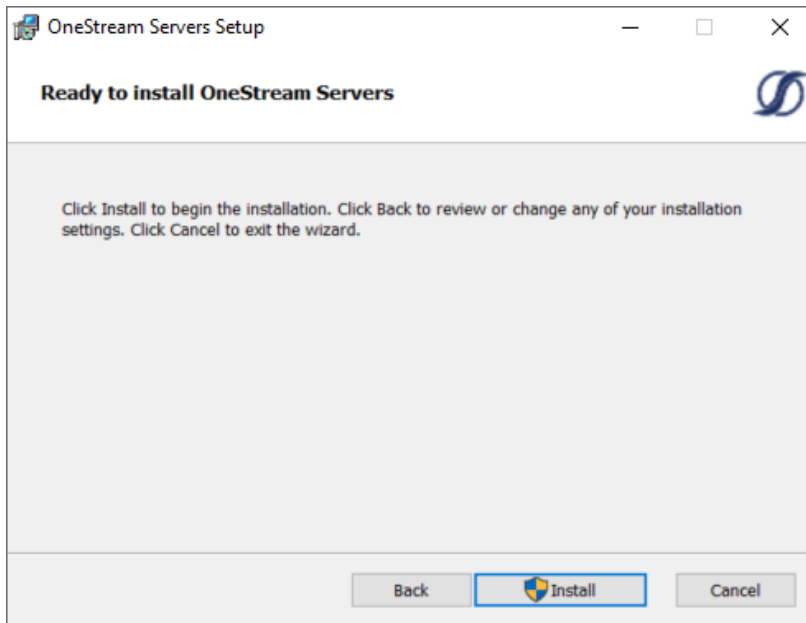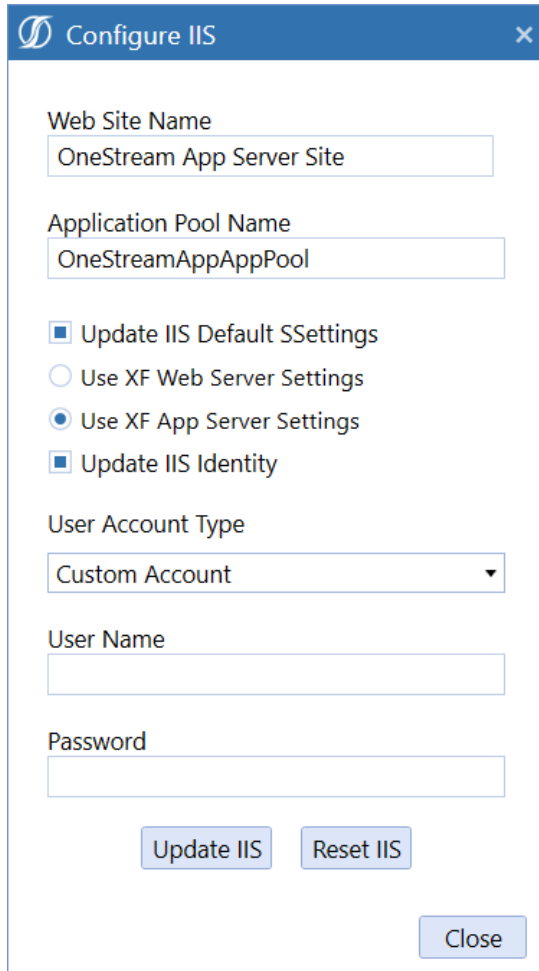
8. Click **Finish** to complete the installation.
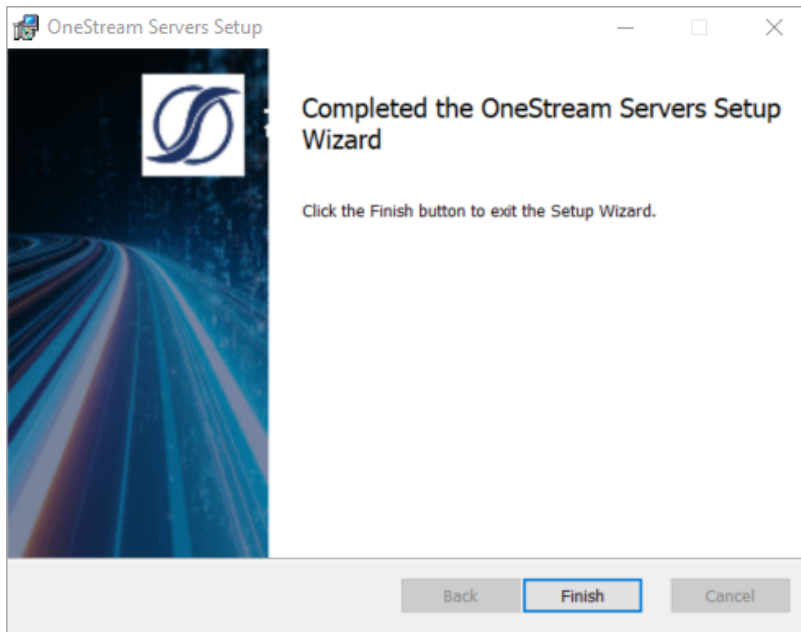
Configuring the OneStream Web Server

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.

Client Options and Installation Guide

This section provides an overview of OneStream's client options and the required installation process, while giving guidance in choosing the proper software configuration and requirements for deployment.

Overview

The Client Installation and Configuration guide provides an overview of the OneStream client applications and the installation process. It also provides guidance in choosing the proper software configuration along with requirements for deployment. Information technology professionals who are responsible for installing, maintaining, and supporting the OneStream platform will find this guide especially useful. You can find supporting documentation on the MarketPlace.

Client Software

You can install OneStream client applications on Windows PCs. After the installation is complete, you can log into the application with your related credentials.

OneStream for Desktop

OneStream for Desktop is a standalone browser-less application for administrators and end-users to access OneStream. The application includes a spreadsheet feature with the potential to eliminate the need for an Excel Add-in which requires administrative rights to install on your desktop.

OneStream Excel Add-In

The OneStream Excel Add-In provides ad hoc querying and reporting, data analysis, data entry, and formatted reports inside of Microsoft Excel. The Excel Add-In lets you perform Excel-based analysis on OneStream data.

Planning the Installation

This section describes requirements and configurations for the client workstations.

Hardware and Software Requirements

The following table presents a list of hardware and software requirements.

Application Server

Software	Requirements
Operating Systems	Windows Server 2022, v10.0 Windows Server 2019, v7.0
Internet Information Server (IIS)	7.0 or higher
.NET Framework	Latest (6.0.14 minimum)

Database Server

Software	Requirements
Operating Systems	Windows Server 2022 (Recommended) Windows Server 2019
Additional Software	SQL Server 2017 SQL Server 2019 SQL Server 2022

Client Workstation

Software	Requirements
Operating Systems	Windows 11, 22H2+, 64-bit Windows 10, 20H2+, 64 bit
.NET Framework	Latest (6.0.14 minimum)
MS Office	Office 365 64-bit Office 2019 64-bit
Web Browser	Microsoft Edge v11.4+ Google Chrome v11.4+ Apple Safari v16.5+

Hardware	Requirements
Screen Resolution	1920 x 1080 (Recommended minimum)

Display Settings

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

Additionally, you should adjust the Windows System Display text setting to 100%. Do not apply any Custom Scaling options.

Installation Packages

The Client Software zip package that contains client installers can be downloaded from the OneStream MarketPlace. Select your platform version, download the Client Software package, and unzip the files to your desired directory using a zip file extraction program.

OneStream for Desktop

There are different considerations to think about when installing the OneStream Desktop application. This section will cover those considerations as well as installation and deployment procedures, upgrading, and uninstalling.

Considerations

You can deploy OneStream for Desktop using two different methods: ClickOnce and traditional installation using an Installer file. Both methods are outlined in the table below. Organizations can choose their preferred method to distribute the application to end-users and their machines.

	ClickOnce	Installer
End-User Deployment	Web page	Manual distribution of the installer file or remote installation by an admin.
Type of install	Temporary installation. The applicationdoes not appear in the Start menu or Add or Remove programs, but you can create a web shortcut on the desktop for easy access.	Traditional installation. The application appears in Start menu and Add or Remove programs.
Package size	~220MB with 325 files	~220MB with 325 files
Requires local administrator	No	No for per-user installs. Yes for per-machine installs.
Upgrades	Auto-upgrades on launch.	Manual

	ClickOnce	Installer
		upgrade with installer file.
Use multiple versions on the same machine	No; possible only if the other version is a traditionalinstall.	Yes
Connect to different servers from a single installed instance	No	Yes

Deployment using ClickOnce

With ClickOnce deployment, you can download and start the OneStream Desktop application from a web page. The client is deployed to the AppData folder located in your profile C:/Users/<username>/AppData/Local/Apps/2.0/<Windows Assigned GUID>. The location is controlled by Windows and .NET. Deploying to the local profile lets users who are not local administrators download and start the application.

> **IMPORTANT:** Every client machine running ClickOnce must have .NET 6 Desktop Runtime Installed.

When you deploy through ClickOnce, the client automatically updates to match the version of the OneStream server software.

Because of the ease of deployment and automatic upgrades, this is the preferred deployment method for customers and users who:

- Can use the ClickOnce technology in their organization or industry.

- Would like to limit IT's involvement in deploying and upgrading the application.

- Do not have to connect to multiple versions of the application at the same time.

> **NOTE:** ClickOnce does not work with Citrix. The workaround for this is a traditional installation . Contact your IT support or add OneStream's URLs as trusted sites in Cisco Umbrella, Z Scaler, or proxy.

Run the ClickOnce Desktop Application

1. Open the Microsoft Edge browser.

2. Navigate to your company's specific OneStream URL to open the OneStream Desktop Application.

   > **Example:**
   > https://<customer>.onestreamcloud.com/OneStreamWeb/One StreamWindowsApp.aspx
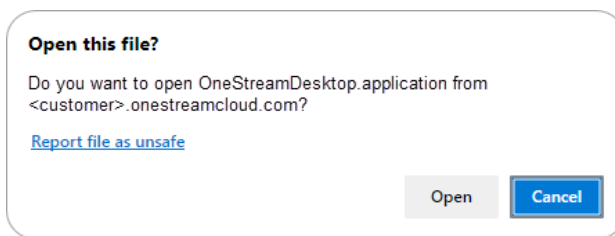
3. Click **Open**. You are automatically redirected to the URL of the launching server which opens up the OneStream Desktop Application.

4. Click the **Create Windows Shortcut** button to save a shortcut to the application on your desktop. The shortcut launches the application and eliminates the need to go to the website every time.
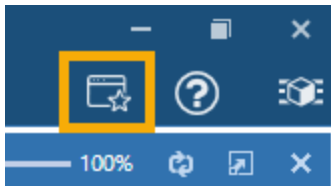
Installation Using the Installer

OneStream for Desktop is offered as two different downloadable installers: EXE and MSI.

	EXE Installer	MSI Installer
User without	Yes, a per-user	Yes, a per-user installation.

	EXE Installer	**MSI Installer**
local admin rights	installation	
User with local admin rights	Yes, a per-machine or per-user installation	Yes, when using elevated Command Prompt or PowerShell. Installs a per-user installation if the MSI file is opened directly.
Install multiple versions	Yes	Yes, when using Command Prompt or PowerShell.

A per-user installation allows only the user who installs it to run OneStream Desktop on the computer on which it is installed. A per-machine installation allows any user to run the application.

The default installation location for a per-user installation is C:\Users\<userid>\AppData\Roaming\Apps\OneStream Software. The default installation location for a per-user installation is C:\Program Files (x86)\OneStream Software.

This deployment strategy is preferred by customers and users who:

- Cannot use the ClickOnce technology in their organization or industry.

- Have to connect to multiple versions of the application at the same time.

- Frequently have to connect to different environments of the same version.

Install OneStream Desktop Using the Install Wizard

1. Double-click the OneStream Desktop MSI or EXE installer file to launch the wizard.

2. Click **Next**.

3. Accept the terms of the license agreement and click **Next**.

4. If necessary, change the folder path, then click **Next**.

5. Click **Install**.

Install Multiple Desktop Versions

A common scenario that requires more than a single version of OneStream Desktop installed on the same computer is when your company has set up a testing environment with a new version of the application prior to upgrading production servers. You might need to install an additional instance of OneStream Desktop on your computer to test the new version and still maintain access to the current version.

You can install up to eight instances of the desktop application on a single computer. Each installation instance has a specific named instance 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. But with a per-machine installation, you must use a named instance that is unique among all users on a machine. The installation attempt will fail if all eight instances are already claimed.

Install Additional Desktop Application Instances

1. Double-click the OneStream Desktop EXE installer file to launch the wizard.

2. Click **Next**.

3. Select one of the following:

   - **Install for all users**: installs the application for all users on the computer

   - **Install for just me**: installs the application only for the current user

4. Click **Next**.

5. Click **Next** again to begin the installation. The instance number is indicated in the title.

6. Accept the terms of the license agreement and click **Next**.

7. If necessary, change the folder path, then click **Next**.

8. Click **Install**.

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

Upgrade OneStream for Desktop

To upgrade to a new version of the application:

1. Double-click the newer version of the OneStream Desktop installer EXE file.

2. Click **Next**. The installer recognizes that a prior version already exists and informs you that an upgrade will be performed.

3. Click **Upgrade**.

Uninstall OneStream for Desktop

If you are not a local administrator, you can only uninstall your own per-user installation. If you are a local administrator, you can uninstall per-machine installations.

To uninstall the application:

1. Close the OneStream Desktop application.

2. Open **Add or Remove Programs**.

3. From the list of installed applications and features, select **OneStream Desktop**.

4. Click **Uninstall** and follow the onscreen instructions.

Use the Command Line

If you are an IT administrator, you can install, upgrade, or uninstall the application using the installer file via the command line.

To run the MSI file via the command-line, use the following steps as a guide:

1. Locate the MSI installer file.

2. Press and hold SHIFT, right-click on the file, and select copy to copy its path.

3. From the Start menu, right-click on the command prompt or Windows PowerShell and select **Run as Administrator**.

4. In the command prompt or Windows PowerShell, use standard MSI command line parameters, along with the file path, to customize the installation as needed.

   The following table shows examples of the most useful command line options.

Purpose	Syntax Example
Normal installation or upgrade process using the full install wizard	msiexec /i "C:\Users\UserName\Downloads\OneStream_Client_7.0.1\OneStreamDesktop\en-us\Setup.msi"
Perform a silent install or upgrade, no user interaction required	msiexec /i "C:\Users\UserName\Downloads\OneStream_Client_7.0.1\OneStreamDesktop\en-us\Setup.msi" /quiet
Perform an unattended install or upgrade, the installation only shows a progress bar	msiexec /i "C:\Users\UserName\Downloads\OneStream_Client_7.0.1\OneStreamDesktop\en-us\Setup.msi" /passive
Uninstall the package	msiexec /x "C:\Users\UserName\Downloads\OneStream_Client_7.0.1\OneStreamDesktop\en-us\Setup.msi"

Excel Add-In

The OneStream application is integrated with Microsoft Excel, which can be used for ad hoc querying/reporting, analysis, data entry, and formatted reports. Excel can also be used with Cube Views. See About Excel Add-In

Considerations

Refer to the following table for guidance to consider when installing the Excel Add-In.

	Installer Deploment
Operating system	Windows
Type of install	Traditional installation. The application appears in the Start menu and Add or Remove programs.
End-User Deployment	Manual distribution of the installer file or remote installation by an administrator.
Requires local administrator	Yes
Version upgrade	Manual upgrade with installer file, or via OneStream Desktops Client Updater.
Use multiple version on the same machine	No
Connect to different servers from single installed instance	Yes

Changes are required to the Windows registry to properly install the OneStream Excel Add-In. You must have permissions to update the registry when performing installations. Changes that are made to the Windows registry are made only for the person who is performing the installation.

For example, if an IT professional is logged into your machine to perform the installation, you will not be able to access the OneStream Excel Add-In. The registration process of the Excel Add-In requires certain Microsoft .NET Framework rights to execute a program in the C:\Windows\Microsoft.NET\Framework folder (or Framework64 if you are running the 64-bit version of Excel).

If you have Excel 2003 or another prior version installed before Excel 2010 or greater, and you have uninstalled the older version of Excel, the newer version of Excel and the OneStream Excel Add-In will need to be uninstalled and then reinstalled.

Other Office Add-Ins may conflict with the Excel Add-In. Therefore, discuss this and other installation issues with OneStream support.

Install the Excel Add-In

Install the Excel Add-In using the standard install wizard:

1. Double-click the Excel Add-In installer file.

2. Click **Next**.

3. Accept the terms of the license agreement and click **Next**.

4. If necessary, change the folder path, then click **Next**.

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

5. Click **Install**.

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

Upgrade the Excel Add-In

You can upgrade the Excel Add-In client in either of two methods: using the installer wizard or using the Desktop Client Updater.

Installer Wizard

Upgrade to a new version using the standard install wizard:

1. Double-click the newer version of the Excel Add-In installer MSI file.

2. Click **Next**.

   The installer recognizes that a prior version already exists and informs you that an upgrade will be performed.

3. Click **Upgrade**.

OneStream for Desktop Client Updater

You can use the Client Updater, located on the Administrator tab of the OneStream application, to upgrade the Excel Add-In. It retrieves updated software from the OneStream server when versions do not match the current version of OneStream found on the server.

There are a few prerequisites necessary prior to upgrading:

- You need write access to the installation folder.

- The Client Updater functionality must be enabled in the application server. Set the following to True: **Application Server Configuration** > **OneStream Environment** > **Can Use Client Updater**.

- You need to be assigned to the ClientUpdaterPage security role.

Upgrade the Excel Add-In:

1. Close all instances of Excel.

2. Launch OneStream for Desktop and log in.

3. On the Application tab, click **Client Updater**.

   If the server version is different from the currently installed Excel Add-In, you will see a message that an update is available.

4. Click **Update**.

   When the server version and the Excel Add-In version match, you will see a message that your Excel Add-In is up to date.

   > **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.

Troubleshooting

If you are attempting to update the Excel Add-In, you might receive the following error message:

"The Client Updater has been disabled by your System Administrator. Please use OneStream's full client installation program, or see your System Administrator."

This indicates that the Client Updater is disabled. The system administrator must enable the Client Updater, or you must use the Excel Add-In MSI installer file instead.

Uninstall the Excel Add-In

> **NOTE:** Only local administrators can uninstall the Excel Add-In.

1. Save any open workbooks and close Excel.

2. Open **Add or Remove Programs**.

3. In the list of Apps and Features, select **OneStream ExcelAddIn**.

4. Click **Uninstall** and follow the onscreen instructions.

Use the Command Line

If you are an IT administrator, you can install, upgrade, or uninstall the application using the installer file via the command line.

To run the MSI file via the command-line, use the following steps as a guide:

1. Locate the MSI installer file.

2. Press and hold SHIFT, right-click on the file, and select copy to copy its path.

3. From the Start menu, right-click on the command prompt or Windows PowerShell and select **Run as Administrator**.

4. In the command prompt or Windows PowerShell, use standard MSI command line parameters, along with the file path, to customize the installation as needed.

   The following table shows examples of the most useful command line options.

Purpose	Syntax Example
Normal installation or upgrade process using the full install wizard	msiexec /i "C:\Users\UserName\Downloads\OneStream_Client_ 7.0.1\OneStreamDesktop\en-us\Setup.msi"
Perform a silent install or upgrade, no user interaction required	msiexec /i "C:\Users\UserName\Downloads\OneStream_Client_ 7.0.1\OneStreamDesktop\en-us\Setup.msi" /quiet
Perform an unattended install or upgrade, the installation only shows a progress bar	msiexec /i "C:\Users\UserName\Downloads\OneStream_Client_ 7.0.1\OneStreamDesktop\en-us\Setup.msi" /passive
Uninstall the package	msiexec /x "C:\Users\UserName\Downloads\OneStream_Client_ 7.0.1\OneStreamDesktop\en-us\Setup.msi"

Side by Side Install

With OneStream Desktop 7.0.0, you can install the application alongside previous versions of the software.

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 verify migration. To test this new version and maintain access to the current version, you need to install another instance of the application.

The application can only communicate with the back-end servers of the same 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.

The new MSI based installer does not recognize or interact with the previous InstallShield based installer. All versions of OneStream Desktop prior to 7.0.0 must be handled independently. OneStreamDesktop 7.0.0 may be installed while previous versions remain on the machine.

Installation Scope

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

Named Instances

The application installer, setup.msi, can be installed up to eight times on a single machine. Each installation instance has a specific name embedded in the installer. These names are:

- 0

- I2

- I3

- I4

- I5

- I6

- I7

- I8

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

> **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

Installing multiple instances of OneStream Desktop on a machine requires understanding both the impact of the different installation scopes, and a knowledge of what has already been installed on the machine, including what per user instances have been installed by other users. To simplify the process, there is a PowerShell script provided (`setup.ps1`) that can handle gathering all of the necessary information and execute the installer with the correct configuration.

To execute a per machine installation, you must open an elevated (admin) PowerShell. Executing the script in a non-elevated PowerShell will execute a per user installation.
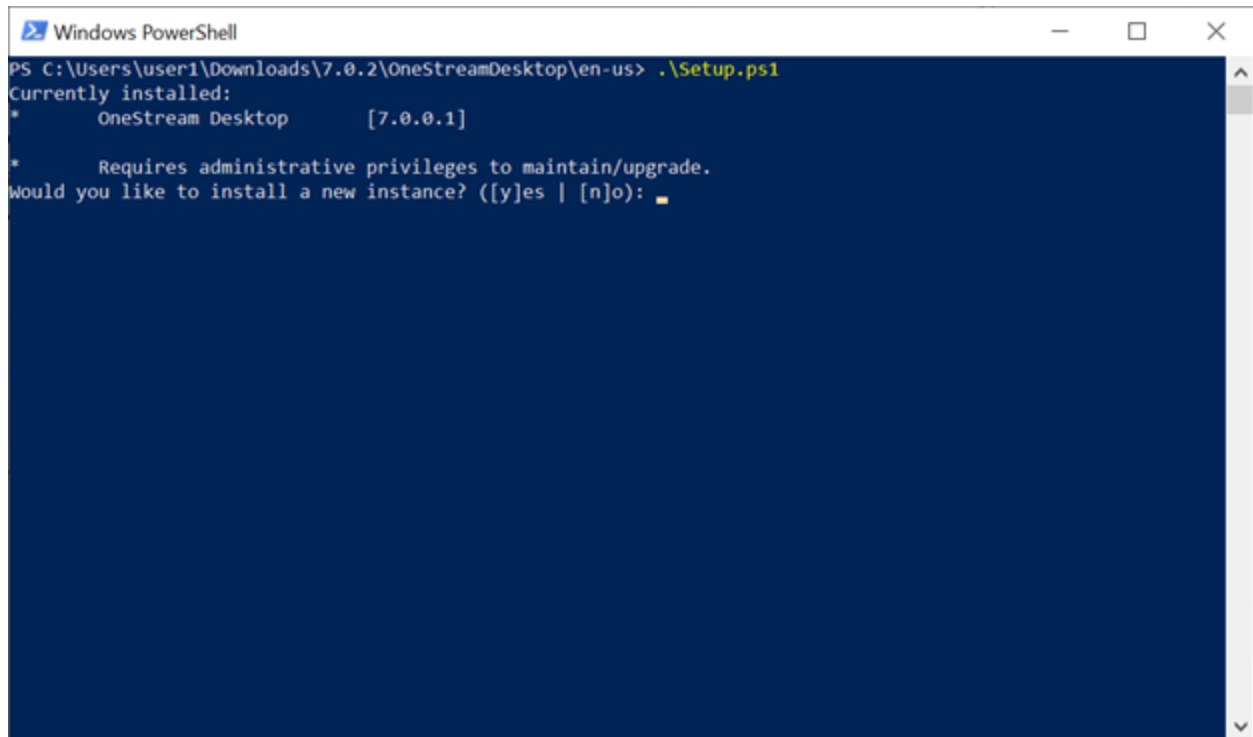
If there are no existing installations when you execute the script, no message is displayed and the installer is immediately launched with the default named instance. If there are one or more installations already on the machine, the script provides a series of prompts to guide you through the installation process.

The following examples provide further information to help understand how to use the script in different scenarios.

Non-Elevated, Per Machine Only

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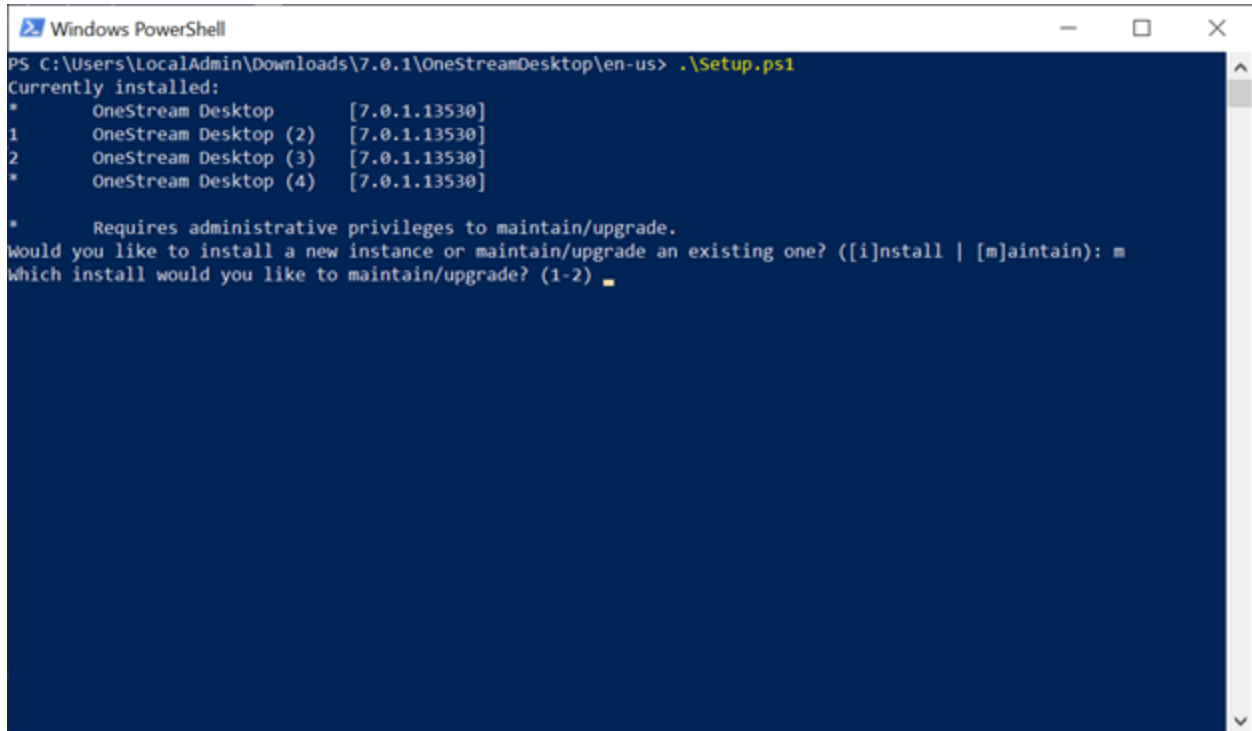
First, all existing instances are listed. The list of known installations is the combination of all per machine installs, plus all per user installs for this user. The asterisk (*) signifies that the installation is a per machine install, and that it cannot be modified because that would require running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

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

Non-Elevated, Complex

In this example, the script is running in a non-elevated PowerShell. There are a combination of both per machine and per user installs already on the machine.
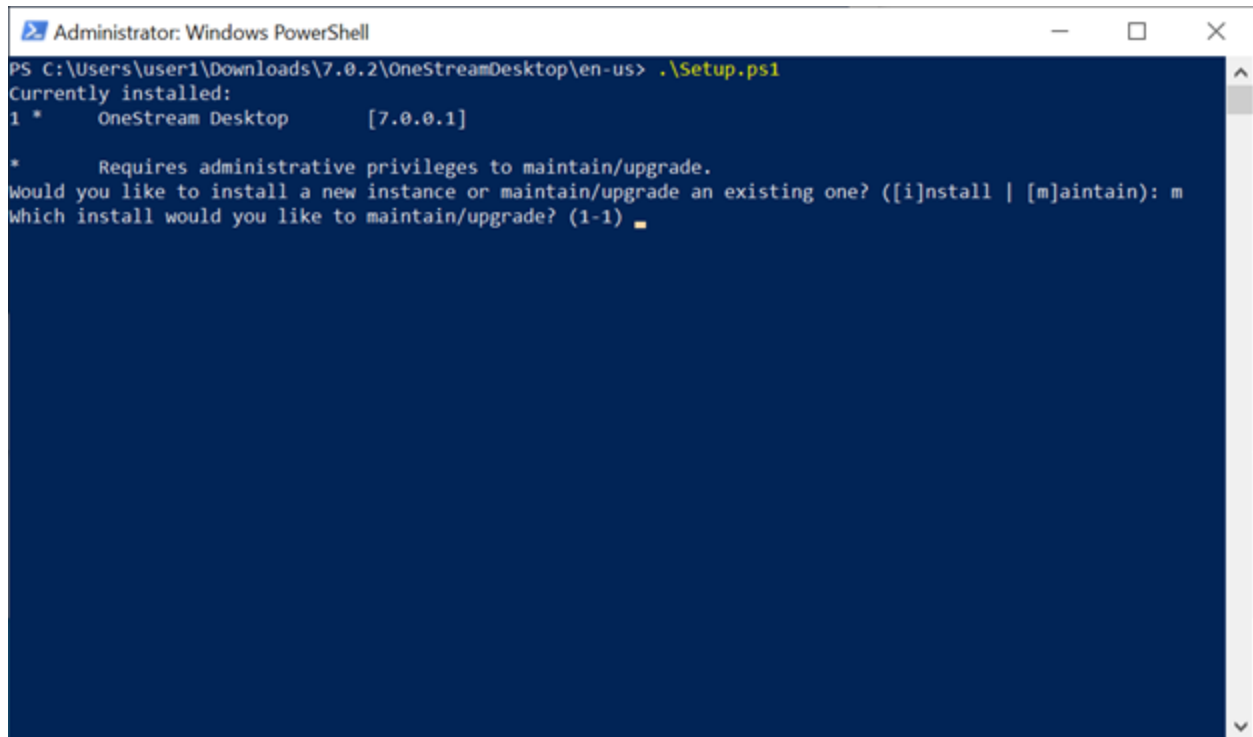
There are four existing installations of OneStream Desktop. The first and fourth installs are per machine, and therefore cannot be modified in a non-elevated PowerShell. The second (I2) and third (I3) instances are per user, and can be modified in the current PowerShell. Typing 'm' and pressing **Enter** to maintain an instance will prompt you to select which installation to modify. Only modifiable instances have a number at the beginning of the list to choose. Typing '1' and pressing **Enter** will 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.

Elevated, Per Machine Only

This example is the same as the Non-Elevated, Per Machine Only example from an elevated PowerShell.
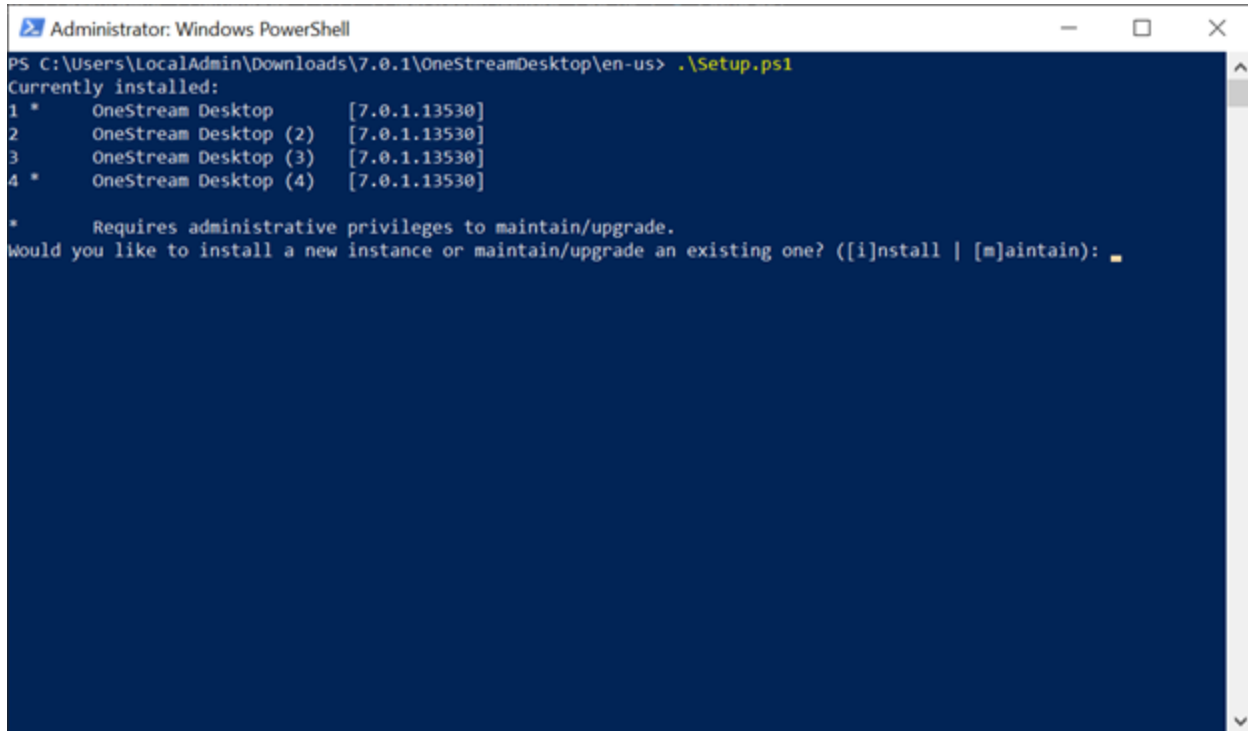
Because the PowerShell is elevated, the existing install can be maintained. Selecting 'i' from the first prompt will install a new per machine instance, while selecting 'm' will prompt you for which instance to maintain, which will then launch the installer for that instance.

Elevated, Complex

This is the same as the Non-Elevated, Complex example.

All instances are numbered, meaning in this elevated PowerShell, any of the installs can be modified. The per machine installs are still 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 autocomplete 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.

Silent Install

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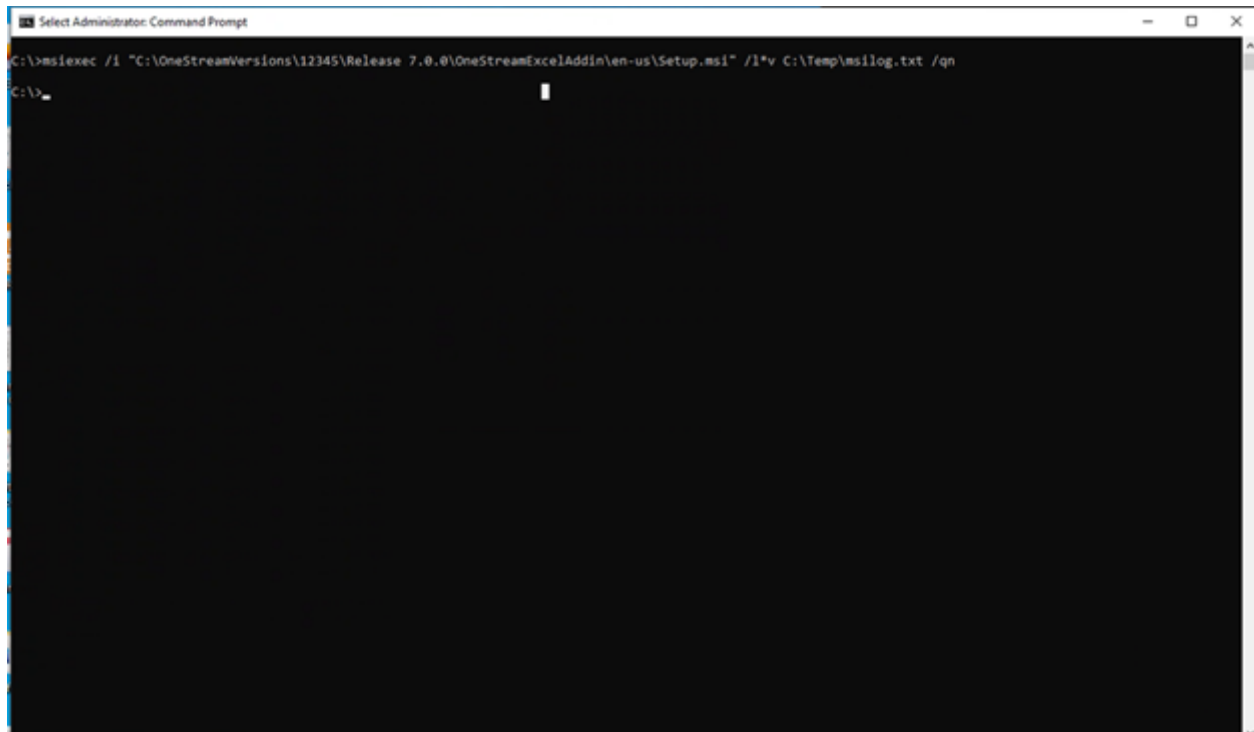
For Excel Add-In enter:

```
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:
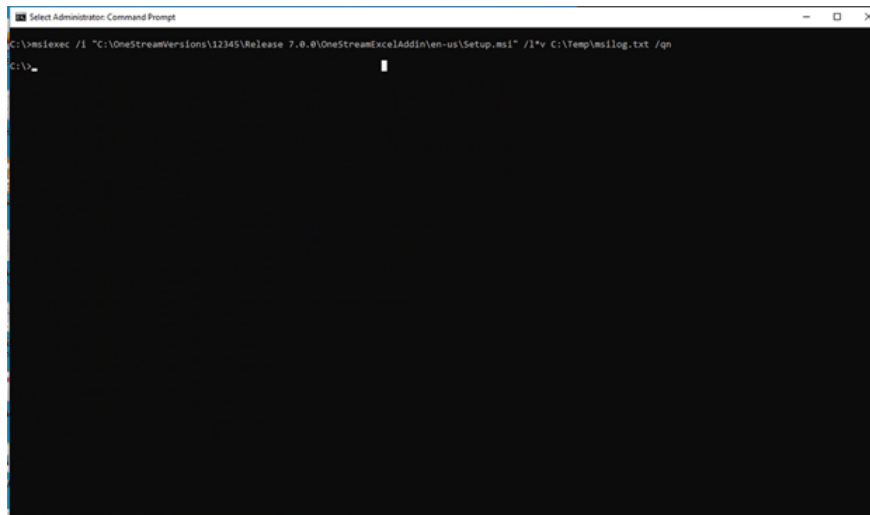

```
C:\>msiexec /x "C:\OneStreamVersions\12345\Release 7.0.0\OneStreamExcelAddin\en-us\Setup.msi" /qn
```

Appendix: 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: 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
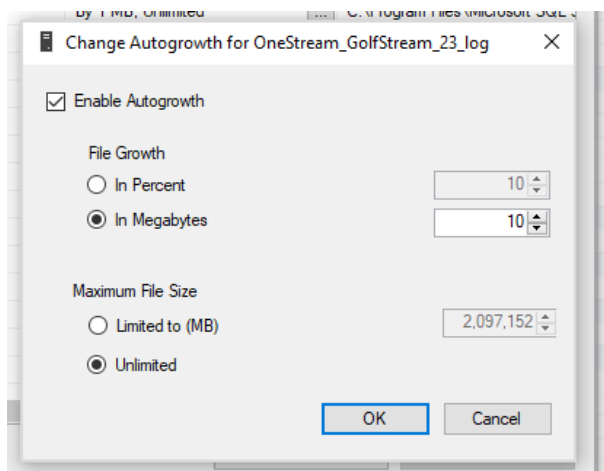
Database Instance Tuning

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: 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:
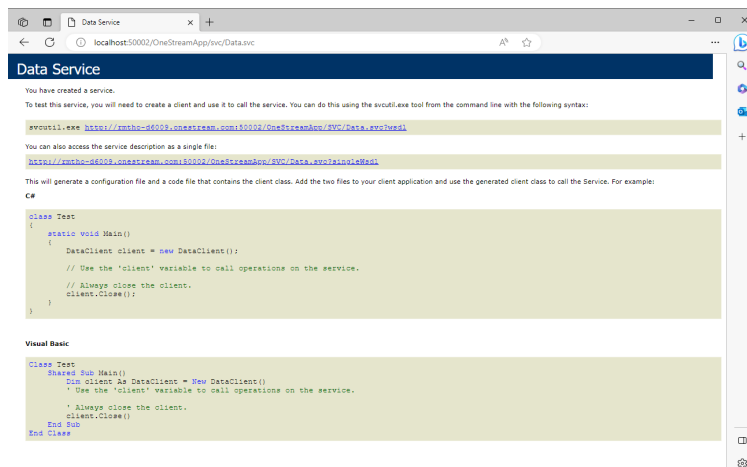
Difficulties Registering the OneStream Excel Add-In in Excel

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.

Browser Issues

If you are working with a 64-bit version of Windows operating system, we recommend that you use the 64-bit version of Edge or another browser.

- If you use the 32-bit version of a browser, you may experience an out of memory situation after several hours of extended use. Restart the web browser to resolve this issue.

- If you run both the client and server on a Virtual Machine (VM) and use the 64-bit version of Windows and the 32-bit version of a browser, browser errors may occur. Microsoft also recommends disabling the PC Tablet Input Service or running the browser outside of the VM when running in this mode.

Appendix: 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

```
XScpcDzo3an7EWAAp6Wvz1IZWE3Q+Fo8DSx0OpacS0Yp5X4sHXtiQGdAoNmNGU48xOa
3xD3C68yAl7B8aXUbv02lj4921ErgN2R+E1qlLG24p98O8a62X0n/q4PS70xzo
sL7R9HzKWT0txtfbpJMUPYrIhCDz6Ubd52/buVABQyUxf2c0Y1BmgIKE/
/6eOcMCv0D5abX6oKbEmZLlms7vXyuicD+KaYheBNX/vbtkA4=
```

Encrypted SAP String Example

```
MF/HrDU0zQupeiyGGSUZ431S1guSfOCDoss4T7JYmiNm8BPTw7inI97W5en
ORZfrVN1Z8ADUHkavsRXKnomFKqBLmddbamOIt5s9bO3jxfXiLI
9B26SDQDKwJer1e6Jc
```

5. Open the OneStream Application Server Configuration tool.

6. Open **Database Server Collections**.

7. Add a database connection.
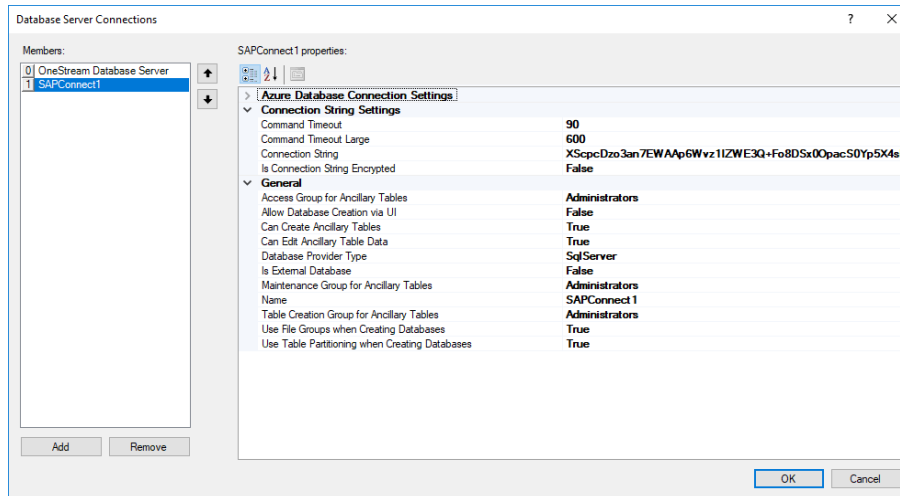
8. (Email Connection Only) In the OneStream Business Rule using the email connection, add the code to call the mail function:

```
'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: 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 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 broweser'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: 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

- Class Name: com.pingidentity.pf.access.token.management.plugins. JwtBearerAccessTokenManagementPlugin

- Type: JSON Web Tokens

- 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://docs.pingidentity.com/r/en-us/pingfederate-111/pf_config_end_user_browser

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: Reserve URL for Native Application Authentication

This section describes how to configure client PCs running pre -5.0 versions of your native applications.

Disregard this section if you 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 browser warnings. Take note of the certificate's thumbprint (ex: d7a045f8xxxxxxxxxb9702066b88bbecf)

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= d7a045f8xxxxxxxxxb9702066b88bbecf 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= d7a045f8xxxxxxxxxb9702066b88bbecf 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: 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.