Deltek Authorization Server Pro

ProPricer Authorization Server Pro Windows Deployment and Upgrade Guide

Contents

Introduction

Authorization Server Pro, an n-tier web application developed with ASP.NET Core, supports various deployment options. Authorization Server Pro provides centralized user management and licensing for ProPricer companion products including BOE Pro and Cost Volume Pro.

The focus of this guide is deployment on Microsoft Windows servers using Internet Information Services (IIS). It explains in detail how to install Authorization Server Pro on premises.

Authorization Server Pro

Minimum requirements

- Microsoft Windows Server 2016 (64 bits).
- Internet Information System 8.0.
- Microsoft SQL Server 2016 (13.x) or later.
- Valid SSL certificate(s) for all websites.
- DNS entry to resolve the fully qualified domain name matching the SSL certificate(s). For example, authorizationserverpro.mycompany.com.
- Optional: PFX certificate for encryption of authentication tokens.

Known limitations

- Bookmarking the URL of the Log In page can cause login issues for users. To quickly open Authorization Server Pro in a browser, users should bookmark the web application start URL instead.

Authorization Server Pro deployment packages

The platform-specific and framework-dependent packages include only the application and its dependencies. The .NET runtime is provided by ASP.NET Hosting Bundle. Deployment packages include a Windows x64 platform-specific executable.

Typically, all packages will be deployed in one and the same Windows server, but you can deploy each to different servers for scalability reasons.

The ZIP packages are available in the Deltek Software Manager (DSM).

Authorization Server Pro WebAPI

DeltekProPricerAuthorizationServerPro_WebAPI_[version]_win-x64.zip

Back-end Authorization Server Pro Web API that receives requests from the Authorization Server Pro web application, provides the WebSockets implementation for live collaboration updates, and communicates with the database to persist the information. This component requires .NET 8.

Authorization Server Pro WebApp

DeltekProPricerAuthorizationServerPro_WebApp_[version].zip

Front-end Authorization Server Pro web application. This component serves only static files (JS, HTML, CSS, etc.).

Authorization Server Pro Database Setup

DeltekProPricerAuthorizationServerPro_DatabaseSetup_[version]_win-x64.zip

Console application that creates and upgrades Authorization Server Pro databases. This component uses .NET 8 single .exe deployment (.NET 8 contained in the .exe).

Create an Authorization Server Pro database

The first thing you must do is create your Authorization Server Pro database. Authorization Server Pro Database Setup is a command line tool that allows database administrators to create and upgrade Authorization Server Pro databases.

1. Download and extract Authorization Server Pro Database Setup from the Deltek Software Manager (DSM).
2. Open a Command Prompt dialog and go to the folder where **AuthorizationServerProDatabaseSetup.exe** is extracted.
3. Run the tool with the create command depending on the SQL authentication method.
    - To use SQL authentication, run the tool with the create command and follow the prompts: `AuthorizationServerProDatabaseSetup create`
    - To use Windows authentication, run the tool with the create command, add the `-w` parameter, and follow the prompts: `AuthorizationServerProDatabaseSetup create -w`

See the Authorization Server Pro Database Setup reference section for further guidance.

Authorization Server Pro prerequisites

Authorization Server Pro is composed of two websites. You can install both Authorization Server Pro sites on a single Windows server or two different servers. Authorization Server Pro WebAPI and Authorization Server Pro WebApp must comply with the following requirements.

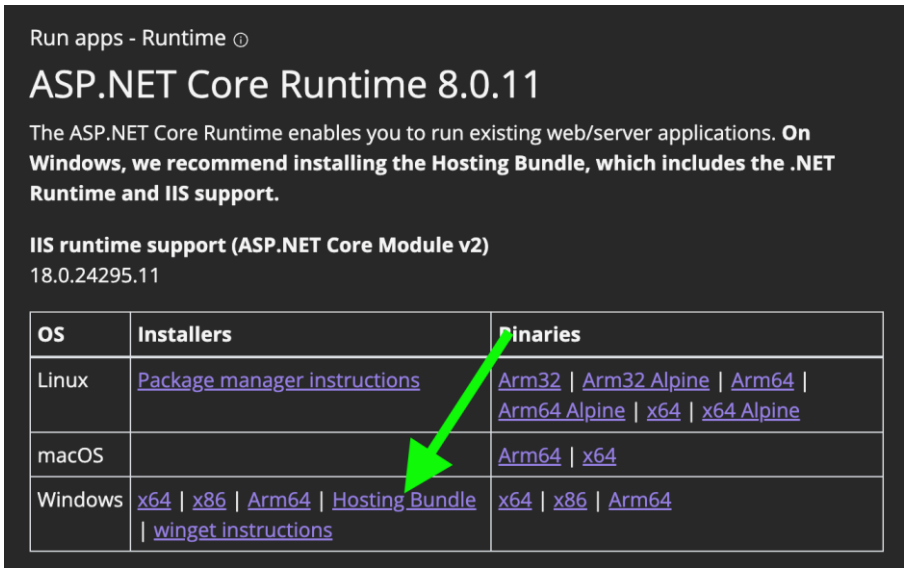
Enable Web Server (IIS)

1. Open **Add Roles and Features**.
2. Select **Web Server (IIS)**.
3. Select the following role services:
    - Common HTTP Features
        - **Default Document**
        - **Directory Browsing**
        - **HTTP Errors**
        - **Static Content**
    - Health and Diagnostics
        - **HTTP Logging**
    - Performance
        - **Static Content Compression**
    - Security
        - **Request Filtering**
    - Management Tools
        - **IIS Management Console**
    - Application Development
        - **WebSockets Protocol**

Install the .NET 8 Windows Server Hosting Bundle

Install the latest .NET 8 Hosting Bundle on the hosting system. The bundle installs the .NET 8 Runtime, .NET 8 Library, and the ASP.NET Core Module. The module allows ASP.NET Core apps to run behind IIS.

1. Go to the Download .NET 8.0 page.
2. Under **ASP.NET Core Runtime**, download the installer using the **Hosting Bundle** link.



3. Run the installer on the server.
4. Restart IIS or execute the following commands in a command shell:

```
net stop w3svc /y

net start w3svc
```

Install the URL rewrite 2.1 IIS extension

Download the extension at: https://www.iis.net/downloads/microsoft/url-rewrite

Install Authorization Server Pro WebAPI

Authorization Server Pro WebAPI connects to the Authorization Server Pro database. Once connected, it provides all required data to Authorization Server Pro WebApp and all other ProPricer companion products.

Prerequisites

Complete all prerequisites before continuing.

Configuration

- Authorization Server Pro WebApp URL.
- Authorization Server Pro Database connection information.
- Optional: Application registration's Application (client) ID, Directory (tenant) ID to enable Azure AD logins, and certificate private key (.pfx) for client certificate.

Installation

The zip packages for this procedure are available in the Deltek Software Manager (DSM).

1. Download and extract the Authorization Server Pro WebAPI package.
2. Move the extracted content to the desired site location. For example, **C:\inetpub\AuthorizationServerProWebAPI**.
3. Open Internet Information Service (IIS) Manager.
4. Create the Authorization Server Pro WebAPI application pool. To create a new pool:
    - Right-click **Application Pools**, then select **Add Application Pool**.
    - In the **Name** field, enter a valid name like Authorization Server **Pro WebAPI App Pool**.
    - In the **.NET CLR version** field, select **No Managed Code**.
    - In the **Managed pipeline mode** field, select **Integrated**.
    - Select **Start application pool immediately**.
    - Click **OK** to create the application pool.
    - Verify that the process model identity has the proper permissions. If the default identity of the application pool (**Process Model** > **Identity**) was changed from **ApplicationPoolIdentity** to something else, then the new identity needs permission to access the application's folder, database (when using Trusted Connection = true for the database connection string), and other required resources. For example, the pool needs Read and Write access to folders where the application reads and writes files.

5. Create the Authorization Server **Pro WebAPI** website.
    - Right-click the **Sites** folder, then select **Add Website**.
    - In the **Site name** field, enter **Authorization Server Pro WebAPI**.
    - In the **Application Pool** field, select the application pool created in previous steps.
    - In the **Physical path** field, select the path to the **Authorization Server Pro WebAPI** folder. For example, **C:\inetpub\AuthorizationServerProWebAPI**.
    - In the **Type** field of the **Binding** section, select **https**.
    - In the **IP address** field, select **All Unassigned**.
    - In the **Port** field, enter a valid port number.

When installing all Authorization Server Pro websites on the same server and under the same domain, use a different port for each website. 44350 is the recommended port for Authorization Server Pro WebAPI.

Ensure the port is accessible to the Authorization Server Pro users.

- In the **Host name** field, enter a domain. For example, **authorizationserverpro.mycompany.com**.
- Select the box **Require Server Name Indication.**
- Select a valid SSL certificate for the domain.
- Click **OK** to create the website.

Configuration

To configure Authorization Server Pro WebAPI, use the Manager tool, or edit **appsettings.json** in the **AuthorizationServerProWebAPI** folder.

The Manager tool is recommended.

1. Open a command prompt window with Administrator permissions.
2. Go to the folder where Authorization Server Pro WebAPI is installed.
3. At the command prompt, enter: `AuthorizationServerProWebAPIManager.exe config`
4. Enter the information requested.
5. To apply your changes, restart the application pool and the website.

You can also provide command line parameters with all the information.

Example

The following is an example configuration of Authorization Server Pro WebAPI for all sites running on the same server authorizationserverpro.mycompany.com under different ports and database connection using SQL authentication. Authorization Server is running in port 44350, and Authorization Server Pro WebApp is running in https default port 443:

```
AuthorizationServerProWebAPIManager.exe config --appurl
https://authorizationserverpro.mycompany.com --authurl
https://authorizationserverpro.mycompany.com:44350 --dbserver
sqlserver.mycompany.com --dbname AuthServerProDb --dbuser sa -dbpassword
MyStrongPassword --lifetime 720 --dbtrustcert true
```

Do not include the port in the URL when using default https port 443.

Use `--dbtrustcert true` to add `TrustServerCertificate=True;` to the DBConnection connection string if your SQL Server certificate is invalid.

To enable management of BOE Pro roles with Authorization Server Pro, use the **Authorization Server Pro WebAPI Manager tool** in the **AuthorizationServerProWebAPI** folder. The Manager tool is recommended.

1. Open a command prompt window with Administrator permissions.
2. Go to the folder where Authorization Server Pro WebAPI is installed.
3. At the command prompt, enter: `AuthorizationServerProWebAPIManager.exe config-boepro-webapi-url`
4. Enter the information requested.
5. To apply your changes, restart the application pool and the website.

To enable management of Cost Volume Pro roles with Authorization Server Pro, use the **Authorization Server Pro WebAPI Manager tool** in the **AuthorizationServerProWebAPI** folder. The Manager tool is recommended.

1. Open a command prompt window with Administrator permissions.
2. Go to the folder where Authorization Server Pro WebAPI is installed.
3. At the command prompt, enter: `AuthorizationServerProWebAPIManager.exe config-cvp-webapi-url`
4. Enter the information requested.
5. To apply your changes, restart the application pool and the website.

Windows authentication for database connection

If you want to use Windows authentication to connect to the Authorization Server Pro database, you must complete the following procedure:

1. Create the Authorization Server Pro database (using Windows or SQL Server authentication).
2. Create a Windows service account.
3. Configure the Windows service account as the WebAPI Application Pool's Identity.
4. Configure Authorization Server Pro WebAPI to use Windows authentication.

The Windows service account must be set as Identity for Authorization Server Pro Web API Application Pool at IIS. It must also be configured with the correct permissions before attempting to connect to the Authorization Server Pro database. To correctly configure the Windows service account:

1. Add the service account to the Authorization Server Pro database with the following permissions.
    - **Default schema**: dbo
    - **Database role membership**: db_owner
2. Add the service account to the Windows server where Authorization Server Pro is installed.
    - Set **Read & execute** permissions as minimum on the Authorization Server Pro Web API folder.
    - Set **Full Control** permissions on folder you want to generate log files.
    - Set **Read** permission to the certificate used to encrypt tokens.
    - Alternatively, you can add the service account to the local Administrators group. If the service account is not intended to be in the local Administrators group, make sure the service account permissions are properly configured for Authorization Server Pro. It must have at least the **Read & execute** permissions for the Authorization Server Pro WebAPI site folders.

If log files are stored in a different folder, the Full Control permission must be granted to the service account for that other folder.

To configure Authorization Server Pro WebAPI to use Windows authentication for database connection, use the Manager tool, or edit **appsettings.json** in the **AuthorizationServerProWebAPI** folder.

The Manager tool is recommended.

1. Open a command prompt window with Administrator permissions.
2. Go to the folder where Authorization Server Pro WebAPI is installed.
3. At the command prompt, enter: `AuthorizationServerProWebAPIManager.exe config -wa true`
4. Enter the information requested.
5. To apply your changes, restart the application pool and the website.

You can also provide command line parameters with all the information.

Example

The following is an example configuration of Authorization Server Pro WebAPI for all sites running on the same server authorizationserverpro.mycompany.com under different ports and database connection using Windows authentication. Authorization Server is running in port 44350, and Authorization Server Pro WebApp is running in https default port 443:

```
AuthorizationServerProWebAPIManager.exe config --appurl
https://authorizationserverpro.mycompany.com --authurl
https://authorizationserverpro.mycompany.com:44350 --dbserver
sqlserver.mycompany.com --dbname AuthServerProDb --dbwinauth true --lifetime 720
--dbtrustcert true
```

Do not include the port in the URL when using default https port 443.

Use `--dbtrustcert true` to add `TrustServerCertificate=True;` to the DBConnection connection string if your SQL Server certificate is invalid.

Certificate for access tokens (Optional)

Authorization Server Pro uses a certificate file to encrypt the access tokens. If no certificate is provided, Authorization Server Pro will generate the file **tempkey.jwk** in the same location as Authorization Server Pro WebAPI.

Ensure the Application Pool Identity user has full permissions in this folder.

Your own certificate is recommended. You can specify the use of a certificate file (.pfx), or a certificate from a Windows certificate store.

Configure certificate from file

To configure a certificate from a file in Authorization Server Pro, use the Manager tool, or edit **appsettings.json** in the **AuthorizationServerProWebAPI** folder.

The Manager tool is recommended.

1. Open a command prompt window with Administrator permissions.
2. Go to the folder where Authorization Server Pro WebAPI is installed.
3. At the command prompt, enter: `AuthorizationServerProWebAPIManager.exe config-token-cert-from-path`
4. Enter the information requested.
5. To apply your changes, restart the application pool and the website.

You can also provide command line parameters with all the information.

Example

The following is an example configuration of Authorization Server Pro using the certificate located in **c:\inetpub\certs\mycert.pfx** and the password **MyCertPassword**:

```
AuthorizationServerProWebAPIManager.exe config-token-cert-from-path --certfile
c:\inetpub\certs\mycert.pfx --cpassword MyCertPassword
```

Ensure the Application Pool Identity user has read permissions to this path.

Configure certificate from store

To configure a certificate from a store in Authorization Server Pro, use the Manager tool, or edit **appsettings.json** in the **AuthorizationServerProWebAPI** folder.

The Manager tool is recommended.

1. Open a command prompt window with Administrator permissions.
2. Go to the folder where Authorization Server Pro WebAPI is installed.
3. At the command prompt, enter: `AuthorizationServerProWebAPIManager.exe config-token-cert-from-store`
4. Enter the information requested. It is recommended that you use the `LocalMachine` location, and `My` as the store name.
5. To apply your changes, restart the application pool and the website.

You can also provide command line parameters with all the information.

Example

The following is an example configuration of Authorization Server Pro using the certificate in the personal store **My** in the **LocalMachine** location with thumbprint **1A1AA111111AAA1111A11AA1A1A11A1AA1A111AA**. The thumbprint is only an example and should be replaced with a valid thumbprint in your configuration:

```
AuthorizationServerProWebAPIManager.exe config-token-cert-from-store --
certstorelocation LocalMachine --cstorename My --cthumbprint
1A1AA111111AAA1111A11AA1A1A11A1AA1A111AA
```

Ensure the Application Pool Identity user has access permissions to the certificate's private key.

Enable Refresh Tokens

If you are upgrading Authorization Server Pro from version 3.6.100.0 or older to version 3.6.100.1 or newer, you must enable refresh tokens.

To enable refresh tokens in Authorization Server Pro and BOE Pro or Cost Volume Pro, use the Manager tool or edit **appsettings.json** in the **AuthorizationServerProWebAPI** folder.

The Manager tool is recommended.

1. Open a command prompt window with Administrator permissions.
2. Go to the folder where Authorization Server Pro WebAPI is installed.
3. At the command prompt, enter: `AuthorizationServerProWebAPIManager.exe enablerefreshtokens`
4. To apply your changes, restart the application pool and the website.

Azure Active Directory (Authorization Server Pro login support)

Authorization Server Pro can allow the use of Azure Active Directory (AD) as a login option. Completing the steps to configure the app registration is required. If you also want to communicate with ProPricer 9 using Azure AD, follow the steps for configuring the client certificate as well.

Register an application with the Microsoft Identity Platform

To enable Azure Active Directory logins, register an app in the Azure Portal so the Microsoft identity platform can provide authentication services for BOE Pro and Cost Volume Pro.

1. Sign into the Azure Portal.
2. If you have access to multiple tenants, use the **Directory +** subscription filter in the top menu to switch to the tenant in which you want to register the application.
3. Search for and select **Azure Active Directory**.
4. Under **Manage**, select **App registrations** > **New registration**.
5. Enter a display name for your application. For example, **Authorization Server Pro**.
6. Specify who can use the application, sometimes called its sign-in audience. **Accounts in this organizational directory only** is recommended.
7. In **Redirect URI**, select the **Web** platform, and enter **<your-authorization-server-pro-url>/signin-oidc**. For example, **https://authorizationserverpro.mycompany.com:44350/signin-oidc**
8. Click **Register**. Wait for the application registration creation.
9. Under **Manage**, select **Authentication**.
10. In **Front-channel logout URL**, enter **<your-authserverpro-webapi-server-url>/signout-oidc**. For example, **https://authorizationserverpro.mycompany.com:44350/signout-oidc**
11. Select **ID tokens (used for implicit and hybrid flows)**.
12. Click **Save**.
13. Under **Manage**, select **Token configuration**.
14. Click **Add optional claim.**
15. Under **Token type**, select **ID**.
16. Select **email**.
17. Click **Add**.
18. Go to **API registration** and click **Grant admin consent** for your tenant.
19. Under **Overview**, use **Application (client) ID** and **Directory (tenant) ID** in your Authorization Server configuration settings.

Reference: *https://docs.microsoft.com/en-us/azure/active-directory/develop/quickstart-register-app*

Add the Client/Desktop platform to the app registration

To enable Azure Active Directory logins when connecting to ProPricer 9, add the Client/Desktop platform to the app registration created in the previous procedure.

Authorization Server Pro requires a client certificate to obtain the token used to log into ProPricer 9 with Azure Active Directory. You must have a valid certificate private key (.pfx).

Ensure Authorization Server Pro and ProPricer 9 are configured to use the same app registration.

1. Select the app registration created for Authorization Server Pro from **App registrations**.
2. Under **Manage**, select **Authentication**.
3. Select **Add a platform**.
4. Select **Mobile and desktop applications**.
5. Select all redirect URI checkboxes, then click **Configure**.
6. Select **Access tokens (used for implicit flows)**.
7. Select **ID tokens (used for implicit and hybrid flows)** if not selected.
8. Click **Save**.
9. Under **Manage**, select **Certificates & secrets**.
10. Under **Certificates**, select **Upload certificate**.
11. Select a certificate file, then click **Add**.

Configure app registration

To configure the app registration to use in Authorization Server Pro, use the Manager tool, or edit **appsettings.json** in the **AuthorizationServerProWebAPI** folder.

The Manager tool is recommended.

1. Open a command prompt window with Administrator permissions.
2. Go to the folder where Authorization Server Pro WebAPI is installed.
3. At the command prompt, enter: `AuthorizationServerProWebAPIManager.exe config-azuread`
4. Enter the information requested.
5. To apply your changes, restart the application pool and the website.

You can also provide command line parameters with all the information.

Example

The following is an example configuration of Authorization Server Pro using the app registration with tenant id **1111a1a1-a11a-1111-a1a1-111a1a1a111a** and client id **111aaa11-aa11-1a11-1a11-1aa11a1111a1** on Azure Government cloud. The tenant and client IDs are only examples and should be replaced with valid IDs in your configuration:

```
AuthorizationServerProWebAPIManager.exe config-azuread --azureadinstance
https://login.microsoftonline.us/ --azureadtenantid 1111a1a1-a11a-1111-a1a1-
111a1a1a111a --azureadclientid 111aaa11-aa11-1a11-1a11-1aa11a1111a1
```

Azure AD client credentials (ProPricer 9 Azure AD login support)

Client credentials are required to support ProPricer 9 Azure AD logins from Authorization Server Pro. You can specify that client credentials use a certificate private key from Key Vault, a Windows certificate store, or a file (.pfx).

Configure client credentials certificate from Key Vault

To configure the client credentials to use a certificate from Key Vault in Authorization Server Pro, use the Manager tool, or edit **appsettings.json** in the **AuthorizationServerProWebAPI** folder.

The Manager tool is recommended.

1. Open a command prompt window with Administrator permissions.
2. Go to the folder where Authorization Server Pro WebAPI is installed.
3. At the command prompt, enter: `AuthorizationServerProWebAPIManager.exe config-azuread-client-cert-from-keyvault`
4. Enter the information requested.
5. To apply your changes, restart the application pool and the website.

You can also provide command line parameters with all the information.

Example

The following is an example configuration of Authorization Server Pro using the certificate **MyCostVolumeProCert** from Key Vault URL **https://mycompany.vault.azure.us**:

```
AuthorizationServerProWebAPIManager.exe config-azuread-client-cert-from-keyvault
--keyvaulturl https://mycompany.vault.azure.us --keyvaultcertname
MyCostVolumeProCert
```

Reference: https://github.com/AzureAD/microsoft-identity-web/wiki/Certificates

Configure client credentials certificate from file (.pfx)

To configure the client credentials to use a certificate from a .pfx file in Authorization Server Pro, use the Manager tool, or edit **appsettings.json** in the **AuthorizationServerProWebAPI** folder.

The Manager tool is recommended.

1. Open a command prompt window with Administrator permissions.
2. Go to the folder where Authorization Server Pro WebAPI is installed.
3. At the command prompt, enter: `AuthorizationServerProWebAPIManager.exe config-azuread-client-cert-from-path`
4. Enter the information requested.
5. To apply your changes, restart the application pool and the website.

You can also provide command line parameters with all the information.

Example

The following is an example configuration of Authorization Server Pro using the certificate file in **C:\inetpub\certs\myCostVolumeProCert.pfx** with password **MyCertPassword**:

```
AuthorizationServerProWebAPIManager.exe config-azuread-client-cert-from-path --certfile C:\inetpub\certs\myCostVolumeProCert.pfx --cpassword MyCertPassword
```

Reference: https://github.com/AzureAD/microsoft-identity-web/wiki/Certificates

Configure client credentials certificate from Windows store

To configure the client credentials to use a certificate from a Windows certificate store in **LocalMachine** in Authorization Server Pro, use the Manager tool, or edit **appsettings.json** in the **AuthorizationServerProWebAPI** folder.

The Manager tool is recommended.

1. Open a command prompt window with Administrator permissions.
2. Go to the folder where Authorization Server Pro WebAPI is installed.
3. At the command prompt, enter: `AuthorizationServerProWebAPIManager.exe config-azuread-client-cert-from-store-with-thumbprint`
4. Enter the information requested.
5. To apply your changes, restart the application pool and the website.

You can also provide command line parameters with all the information.

Example

The following is an example configuration of Authorization Server Pro using the certificate in the personal store **My** in the **LocalMachine** location with thumbprint **1A1AA111111AAA1111A11AA1A1A11A1AA1A111AA**. The thumbprint is only an example and should be replaced with a valid thumbprint in your configuration:

```
AuthorizationServerProWebAPIManager.exe config-azuread-client-cert-from-store-with-thumbprint --cstorename My --cthumbprint 1A1AA111111AAA1111A11AA1A1A11A1AA1A111AA
```

Reference: https://github.com/AzureAD/microsoft-identity-web/wiki/Certificates

Enable logging

Authorization Server Pro WebAPI uses the Serilog library to provide diagnostic logging. Logging to file is optional but recommended.

1. In the **AuthorizationServerProWebAPI** folder, edit **appsettings.json**.
2. In the **<Serilog>** section, find the **WriteTo** array section, and add one entry for **File** above the **Console** entry.
3. Specify the path where you want the log file to be created. Ensure the application pool's identity can write to this folder.
4. To apply your changes, restart the application pool and the website.

Example

The new **File** section is highlighted yellow:

```
"Serilog": {
    "Using": [ "Serilog.Sinks.Console" ],
    "MinimumLevel": { "Default": "Information" },
    "WriteTo": [
      {
        "Name": "File",
        "Args": {
          "path": "\\inetpub\\logs\\logfiles\\authserverpro-webapi.log",
          "rollOnFileSizeLimit": "true"
        }
      },
      {
        "Name": "Console",
        "Args": {
          "theme":
"Serilog.Sinks.SystemConsole.Themes.AnsiConsoleTheme::Literate,
Serilog.Sinks.Console"
        }
      }
    ],
```

References

- Learn more about Windows authentication and installing it for IIS. https://docs.microsoft.com/en-us/iis/configuration/system.webServer/security/authentication/windowsAuthentication/
- Configure Windows authentication in an ASP.NET Core app. https://learn.microsoft.com/en-us/aspnet/core/security/authentication/windowsauth?tabs=visual-studio&view=aspnetcore-8.0
- Host ASP.NET Core on Windows with IIS. https://learn.microsoft.com/en-us/aspnet/core/host-and-deploy/iis/?view=aspnetcore-8.0

Install Authorization Server Pro WebApp

Authorization Server Pro WebApp is the front-end user interface for Authorization Server Pro. This website serves static files only, and it is the URL administrators will use.

Prerequisites

Complete all prerequisites before continuing.

Configuration

- Authorization Server Pro WebAPI URL.

Installation

The zip packages for this procedure are available in the Deltek Software Manager (DSM).

1. Download and extract the Authorization Server Pro WebApp package.
2. Move the extracted content to the desired site location. For example, **C:\inetpub\AuthorizationServerProWebApp.**
3. Open **Internet Information Service (IIS) Manager**.
4. Create a new application pool. To create a new pool:
    - Right-click **Application Pools**, then select **Add Application Pool**.
    - In the **Name** field, enter **Authorization Server Pro WebApp App Pool**.
    - In the **.NET CLR version** field, select **No Managed Code**.
    - In the **Managed pipeline mode** field, select **Integrated**.
    - Select **Start application pool immediately**.
    - Click **OK** to create the application pool.

5. Create the Authorization Server Pro WebApp website.
   - Right-click the **Sites** folder, then select **Add Website**.
   - In the **Site name** field, enter **Authorization Server Pro WebApp**.
   - In the **Application pool** field, select the pool created in previous steps.
   - In the **Physical path** field, enter or select the path where the files will be located. For example, **C:\inetpub\AuthorizationServerProWebApp**.
   - In the **Type** field of the **Binding** section, select **https**.
   - In the **IP address** field, select **All Unassigned**.
   - In the **Port** field, enter a valid port number.

When installing all Authorization Server Pro websites on the same server and under the same domain, use a different port for each website. 443 is the recommended port for WebApp.

Ensure the ports are accessible to the Authorization Server Pro users.

   - In the **Host name** field, enter a domain.
   - Select **Require Server Name Indication**.
   - Select a valid SSL certificate for the domain.
   - Click **OK** to create the website.

Configuration

To configure Authorization Server Pro WebApp, use the Manager tool, or edit **config.js** in the target folder.

The Manager tool is recommended.

1. Open a **Command Prompt** window with Administrator permissions.
2. Go to the folder where Authorization Server Pro WebApp is installed.
3. At the command prompt, enter: `AuthorizationServerProWebAppManager.exe config`
4. Enter the information requested.
5. To apply your changes, restart the application pool and the website.

You can also provide command line parameters with all the information.

Example

The following is an example configuration of Authorization Server Pro WebApp when WebAPI site is running on the same server authorizationserverpro.mycompany.com under different port. Authorization Server Pro WebAPI is running in port 44350:

```
AuthorizationServerProWebAppManager.exe config --webapi
https://authorizationserverpro.mycompany.com:44350
```

Microsoft Clarity (Optional)

Microsoft Clarity is a user behavior analytics tool that helps you understand how users are interacting with your website through features such as session replays and heatmaps.

Enabling Microsoft Clarity is completely optional. Be aware that Microsoft collects all data and Authorization Server Pro and Deltek, Inc. have no access to the data from Clarity. Before enabling, ensure that this is not a security problem for your organization.

To enable Microsoft Clarity in Authorization Server Pro:

1. Sign into https://clarity.microsoft.com.
2. Create a new Clarity project for BOE Pro or Cost Volume Pro.
3. Go to **Settings** > **Overview** and copy the Project ID.
4. At the command prompt, enter: `AuthorizationServerProWebAppManager.exe EnableClarity`
5. When prompted, enter the Project ID.
6. To apply your changes, restart the application pool and the website.

Test your Authorization Server Pro installation

After Authorization Server Pro WebAPI and WebApp are installed, test your installation by launching a web browser and go to your Authorization Server Pro WebApp URL, for example https://authorizationserverpro.mycompany.com.

The WebApp will redirect to the login screen as this is the first time accessing it. The login screen is provided by the WebAPI. Enter the system administrator username and password you entered when the database was created.

If the username and password are correct, you should see the Deltek Authorization Server Pro main screen with the user created during database creation.

You can create more users, activate licenses, or install companion products now.

Upgrade Authorization Server Pro

To upgrade Authorization Server Pro, get the new version packages, upgrade the database, and upgrade both Authorization Server Pro websites.

Upgrade recommendations

- Plan for some downtime because Authorization Server Pro sites stop during the process. This affects all ProPricer companion products that use Authorization Server Pro, such as BOE Pro and Cost Volume Pro.
- Back up a copy of your sites before upgrading them.
- Back up your database before upgrading it.
- Go to the Deltek Software Manager (DSM) to download the zip packages for the new version you want to upgrade to.

Make sure you Download and install the latest .NET 8 Hosting Bundle.

1. Download the Authorization Server Pro packages.
   - **DeltekProPricerAuthorizationServerPro_WebAPI_[version]_win-x64.zip**
   - **DeltekProPricerAuthorizationServerPro_WebApp_[version].zip**
   - **DeltekProPricerAuthorizationServerPro_DatabaseSetup_[version]_win-x64.zip**

   The zip packages are available in the Deltek Software Manager (DSM).
2. Using IIS Manager stop all sites and application pools:
   - Stop the Authorization Server Pro WebAPI website.
   - Stop the Authorization Server Pro WebAPI application pool.
   - Stop the Authorization Server Pro WebApp website.
   - Stop the Authorization Server Pro WebApp application pool.
3. Upgrade your database
   - Extract **DeltekProPricerAuthorizationServerPro_DatabaseSetup_[version]_win-x64.zip**
   - Open a command prompt window.
   - Go to the folder where **AuthorizationServerProDatabaseSetup**.exe is extracted
   - At the command prompt, enter: `AuthorizationServerProDatabaseSetup.exe upgrade`
   - To connect using Windows authentication enter: `AuthorizationServerProDatabaseSetup.exe upgrade -w`
   - Enter the information requested.

4. Upgrade Authorization Server Pro WebAPI site.
   - Using Windows Explorer, go to the folder for the Authorization Server Pro WebAPI site.
   - Back up the **appsettings.json** file.
   - Remove everything in the folder.
   - Extract **DeltekProPricerAuthorizationServerPro_WebAPI_[version]_win-x64.zip**
   - Copy all extracted files to the WebAPI site's folder inside **C:\inetpub**. For example, **C:\inetpub\AuthorizationServerProWebAPI**.
   - Restore the **appsettings.json** file.
5. Verify configuration with the Manager tool.
   - Open a command prompt window with Administrator permissions.
   - Go to the folder where Authorization Server Pro Web API is installed.
   - At the command prompt, enter: `AuthorizationServerProWebAPIManager.exe config`
   - If you are using Windows authentication to connect to SQL Server, enter: `AuthorizationServerProWebAPIManager.exe config -wa true`
   - Press **Enter** key to accept the existing values.
6. Upgrade Authorization Server Pro WebApp site.
   - Using Windows Explorer, go to the folder for the Authorization Server Pro WebApp site.
   - Back up the **config.js** file.
   - Remove everything in the folder.
   - Extract **DeltekProPricerAuthorizationServerPro_WebApp_[version].zip**
   - Copy all extracted files to the WebApp site's folder inside **C:\inetpub**. For example, **C:\inetpub\AuthorizationServerProWebApp**.
   - Restore the **config.js** file.
7. Verify configuration with the Manager tool.
   - Open a command prompt window with Administrator permissions.
   - Go to the folder where Authorization Server Pro WebApp is installed.
   - At the command prompt, enter: `AuthorizationServerProWebAppManager.exe config`
   - Enter the information requested.
   - Press **Enter** to accept the existing values.
8. Using IIS Manager:
   - Start the Authorization Server Pro WebAPI application pool.
   - Start the Authorization Server Pro WebAPI website.
   - Start the Authorization Server Pro WebApp application pool.
   - Start the Authorization Server Pro WebApp website.

Authorization Server Pro Database Setup reference

Authorization Server Pro Database Setup is a command line tool that allows the database administrators to create and upgrade Authorization Server Pro databases.

Usage

```
AuthorizationServerProDatabaseSetup [options] [command]
```

Options

`-v|--version` Show version information.

`-?|-h|--help` Show help information.

Commands

`addsysadmin` Adds a system administrator account to an Authorization Server Pro database.

`create` Creates a new Authorization Server Pro database.

`upgrade` Upgrades a database for Authorization Server Pro to the current version.

create command

`create` Creates an Authorization Server Pro database.

Usage

```
AuthorizationServerProDatabaseSetup create [options]
```

Options

`-?|-h|--help` Show help information.

`-f|--scripttofile <fileName>` Output the script to a file.

`-s|--server <servername>` Hostname of the database server.

`-w|--windowsauth` Use Windows authentication.

`-d|--dbname <databasename>` Name of the database. Default value is `AuthServerProDb`.

`-al|--adminlogin <login>` Authorization Server Pro admin user email. Default value is `sysadmin@propricer.com`.

`-ap|--adminpass <password>` Authorization Server Pro admin user password. Default value is `sysadmin`.

`-an|--adminname <name>` Authorization Server Pro admin username. Default value is `System Administrator`.

`-u|--dbauser <dbalogin>` SQL Server authentication database user login.

`-p|--dbapass <dbapassword>` SQL Server authentication database user password.

`-e|--useexistingdb` Use an existing database (for example, when using Azure SQL Database).

Examples

Create an Authorization Server Pro database on an SQL Server on premises using Windows authentication:

```
AuthorizationServerProDatabaseSetup create -s sqlserver.mycompany.com -d
AuthServerProDb -al sysadmin@mycompany.com -ap MyPassword4AuthServerPro -an
"System Administrator" -w
```

Create an Authorization Server Pro database on an SQL Server on a VM using Windows authentication:

```
AuthorizationServerProDatabaseSetup create -s sqlserver.eastus.cloudapp.azure.com
-d AuthServerProDb -al sysadmin@mycompany.com -ap MyPassword4AuthServerPro -an
"System Administrator" -w
```

Create an Authorization Server Pro database on an Azure SQL database using SQL authentication:

```
AuthorizationServerProDatabaseSetup create -s mycompany.database.windows.net -u
myazureuser -p MyAzurePassword -d AuthServerProDb -al sysadmin@mycompany.com -ap
MyPassword4AuthServerPro -an "System Administrator" -e
```

upgrade command

`upgrade` Upgrades an existing Authorization Server Pro database.

Usage

```
AuthorizationServerProDatabaseSetup upgrade [options]
```

Options

`-?|-h|--help` Show help information.

`-s|--server <servername>` Hostname of the database server.

`-w|--windowsauth` Use Windows authentication.

`-u|--dbauser <dbalogin>` SQL Server authentication database user login.

`-p|--dbapass <dbapassword>` SQL Server authentication database user password.

`-d|--dbname <databasename>` Name of the database. Default value is `AuthServerProDb`.

`-f|--scripttofile <fileName>` Output the script to a file.

Examples

Upgrade a database named AuthServerProDb on an SQL Server on premises using Windows authentication:

```
AuthorizationServerProDatabaseSetup upgrade -s sqlserver.mycompany.com -d
AuthServerProDb -w
```

Upgrade a database named AuthServerProDb on an Azure SQL database using SQL authentication:

```
AuthorizationServerProDatabaseSetup upgrade -s mycompany.database.windows.net -u
myazureuser -p MyAzurePassword -d AuthServerProDb
```

addsysadmin command

`addsysadmin` Creates an administrator-type user in a previously created Authorization Server Pro database. The user will have the Administrator permission to access Authorization Sever Pro but cannot access any companion product.

Usage

```
AuthorizationServerProDatabaseSetup addsysadmin [options]
```

Options

`-?|-h|--help` Show help information.

`-s|--server <servername>` Hostname of the database server.

`-w|--windowsauth` Use Windows authentication.

`-d|--dbname <databasename>` Name of the database. Default value is `AuthServerProDb`.

`-al|--adminlogin <login>` Authorization Server Pro admin user email. Default value is `sysadmin@propricer.com`.

`-ap|--adminpass <password>` Authorization Server Pro admin user password. Default value is `sysadmin`.

`-an|--adminname <name>` Authorization Server Pro admin username. Default value is `System Administrator`.

`-u|--dbauser <dbalogin>` SQL Server authentication database user login.

`-p|--dbapass <dbapassword>` SQL Server authentication database user password.

Troubleshoot

Authorization Server Pro – Network error

You might receive the “Network Error.” message after entering the WebApp URL in your browser (for example, **https://costvolumepro.mycompany.com**).

Solution 1

Check your networking. Verify that your firewall is not blocking the Authorization Server Pro WebAPI port.

Solution 2

Restart the Authorization Server Pro WebAPI website.

Solution 3

1. In your browser, enter your authorization server webapi URL followed by **/.well-known/openid-configuration** (for example, **https://authorizationserverpro.mycompany.com/.well-known/openid-configuration**).
2. If you receive an error instead of the .json file, go to the Authorization Server Pro WebAPI folder, and find the file **appsettings.json**. Verify that it is a valid .json file containing the correct configuration.
3. Install or repair .NET Core hosting bundle and restart IIS service.

Solution 4

1. In this guide, follow the instructions for enabling logging in Authorization Server Pro .
2. Try logging into Authorization Server Pro again, and check the log for the error.

Solution 5

If there is no log file, check Windows Event Viewer.

Authorization Server Pro WebAPI – Common errors

HTTP Error 500.31 - Failed to load ASP.NET Core runtime

Install or repair .NET Core hosting bundle and restart IIS service.

500.119

The application pool identity cannot read **appsettings.json**. To fix this, assign permissions to the folder where the website files are located (for example, **C:\inetpub\CostVolumeProWebAPI**).

HTTP Error 503 – The service is unavailable

The service is unavailable when the application pool of the corresponding web application is stopped, disabled, or paused. It may also be unavailable when the given user identity of the application pool is invalid due to an expired password or is locked.

https://blogs.msdn.microsoft.com/webtopics/2010/02/17/a-not-so-common-root-cause-for-503-service-unavailable/

https://stackoverflow.com/questions/13322937/http-error-503-the-service-is-unavailable

Bad Request – Request header too long

“HTTP Error 400. The size of the request header is too long.”

While using Windows authentication, this error may appear instead of the BOE Pro or Cost Volume Pro application. It is the result of the user being a member of many Active Directory user groups.

To fix this:

1. Open **Windows Registry Editor** and go to:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\HTTP\Parameters
```

2. Increase the settings for the `MaxFieldLength` and the `MaxRequestBytes` registry entries on the server so that the user's request headers do not exceed these values. For example:

```
"MaxRequestBytes"=dword:01000000

"MaxFieldLength"=dword:0000fffe
```

https://docs.microsoft.com/en-us/troubleshoot/iis/http-bad-request-response-kerberos

https://www.grouppolicy.biz/2013/06/how-to-configure-iis-to-support-large-ad-token-with-group-policy