# ELO Sync

Installation

# Table of contents

# Manual installation

## Requirements

> **Information**
>
> We recommend installing ELO Sync using the ELO Server Setup.

Make sure that all requirements are met before installing ELO Sync.

OAuth must have been set up.

### Setting up OAuth

To set up OAuth, refer to the following sections:

- Configuration of ELO Auth (ELO 23.6 or lower)
- Configuration of ELO Modern Authentication (Auth2) (ELO 23.6 or higher)

### Windows

The following requirements must be met to install ELO Sync on Microsoft Windows:

- Microsoft Windows 10 version 1607 or higher
- Min. 1 GB RAM
- x64 processor with at least two kernels
- 1 GB hard drive space
- Access to a database server with one of the supported DBMS
- Access to an ELO repository. The account must have administrator rights.

See also:

- .NET 8 - Supported OS versions

### Linux

The following requirements must be met to install ELO Sync on Linux:

- glibc 2.17+ or musl 1.2.2+
- OpenSSL 1.x or 3.x
- Min. 1 GB RAM
- x64 processor with at least two kernels
- 1 GB hard drive space
- Access to a database server with one of the supported DBMS
- Access to an ELO repository. The account must have administrator rights.

See also:

- .NET 8 - Supported OS versions
- .NET Support and Compatibility for Linux Distributions

> **Information**
>
> The requirements only indicate what ELO Sync itself requires.
>
> The requirements for the operating system and all other services running on the same system also have to be aligned.

# Method

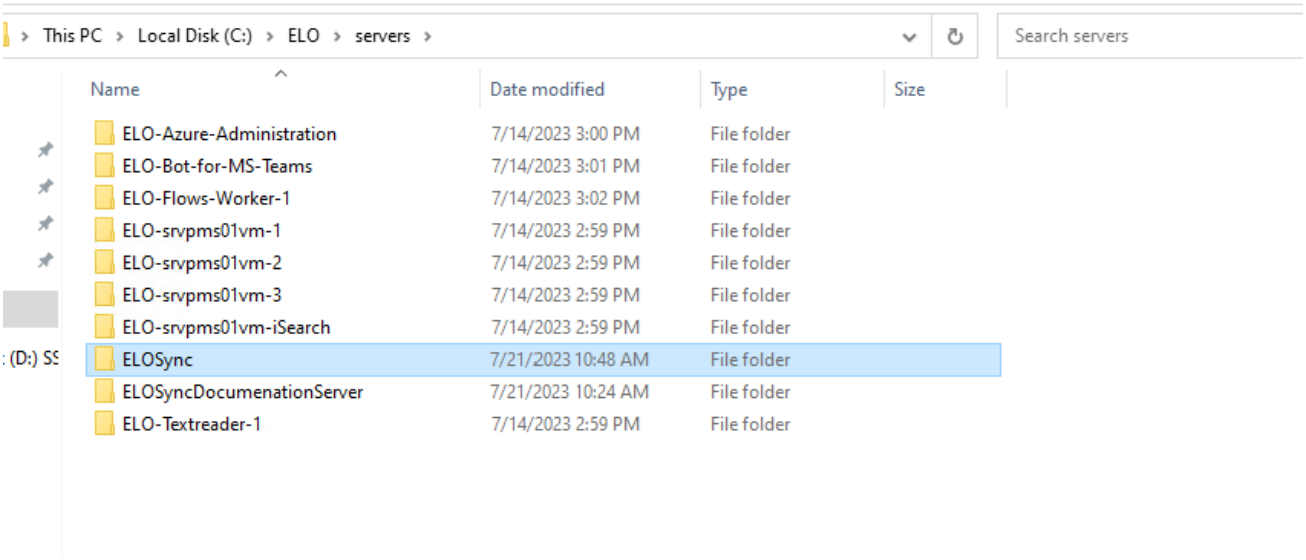
> We recommend you use the ELO Server Setup to install ELO Sync, rather than performing a manual installation.

Make sure that all requirements are met before installing ELO Sync.

ELO Sync can be installed on the following operating systems:

- Windows
- Linux

## Windows




1. Create the target directory on your system and extract the files from ELO Sync.
2. Follow the guide on Registering ELO Sync in Azure and note the application information.
3. Copy the *appsettings.json* file as *appsettings.Production.json*.

   This ensures that your configuration is not overwritten by future updates.
4. Open the configuration file in a text editor of your choice.
   1. Use the information from step 2 to change the following configuration section.

```
"AzureAd": {
 "Instance": "https://login.microsoftonline.com/",
 "Domain": "example.onmicrosoft.com",
 "ClientId": "00000000-0000-0000-0000-000000000000",
 "TenantId": "11111111-2222-3333-4444-555555555555",
```

```
 "ClientSecret": "TheClientSecretFromAzurePortal",
 "CallbackPath": "/signin-oidc-custom"
},
```

Some of these terms have changed over time. Below is a short list of alternative names for each setting:

- ClientId: AppID, Application client ID
- TenantId: Directory ID

1. Enter the PublicUrl path if ELO Sync should be called from a domain other than the internal one, e.g. via a proxy.

```
"PublicUrl": "https://domain:port/path/to/elosync",
```

1. Change the logon information for the service user that will be used to connected to the ELO repository:

```
"ServiceUser": {
"UserName": "ELO Service",
"Password": "ThePasswordForTheServiceUser"
}
```

1. Configure the ELO repositories to be accessed via ELO Sync:

```
"Repositories": [
{
  "name": "Display Name for Repository",
    "key": "TechnicalKeyForRepository",
    "url": "https://elo-example-server.com:9093/ix-Repository/ix",
    "webclienturl": "https://elo-example-server.com:9093/ix-Repository/plugin/de.elo.
    "oauthcallbackurl": "https://elo-example-server.com:9093/ix-Repository"
}
]
```

The technical key for the repository can be any character except a space (' '). We recommend using the repository name unless this name is not unique on all servers.

1. If necessary, change the profile key for the ELOauth plug-in. This is used if the user logs on to the ELO Sync web interface.

```
"OAuth": {
"ConfigId": "elo_sync_oauth"
}
```

1. Enter the database type and the connection string.

```
"Database": "Postgres",
"ConnectionStrings": {
"Postgres": "User ID=dbuser;Password=dbpassword;Server=dbserver;Port=5432;Database=elo
"MsSql": "Server=dbserver,1433;Database=elosyncdb;User Id=dbuser;Password=dbpassword;
},
```

1. Configure the web server endpoints:

```
"Kestrel": {
  "Endpoints": {
    "Https": {
    "Url": "https://elo-sync-server",
    "Certificate": {
    "Path": "C:\\Path\\To\\Certificate.pfx",
    "Password": "PasswordForCertificate"
    }
    }
  },
},
```

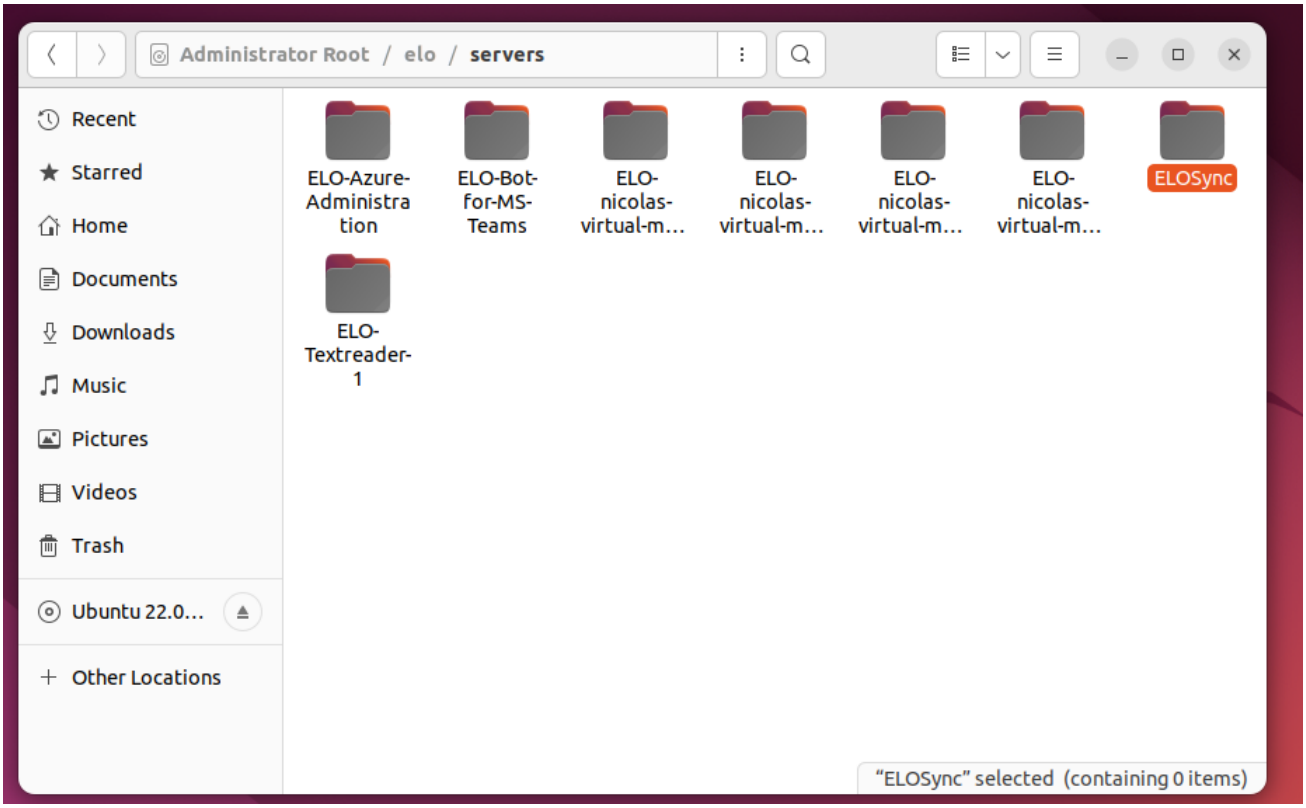
ELO Sync is now installed and fully configured.

To test ELO Sync, open a terminal in the ELO Sync installation directory and execute the following command:

```
.\Elo.Sync.Main.exe
```

You can find more details about configuration in the *appsettings.json* (or *appsettings.Production.json*) file in the Configuration section.

## Linux




1. Create the target directory on your system and extract the files from ELO Sync.
2. Follow the guide on Registering ELO Sync in Azure and note the application information.
3. Copy the *appsettings.json* file as *appsettings.Production.json*.

   This ensures that your configuration is not overwritten by future updates.
4. Open the configuration file in a text editor of your choice.
   1. Use the information from step 2 to change the following configuration section.

```
"AzureAd": {
 "Instance": "https://login.microsoftonline.com/",
 "Domain": "example.onmicrosoft.com",
 "ClientId": "00000000-0000-0000-0000-000000000000",
 "TenantId": "11111111-2222-3333-4444-555555555555",
 "ClientSecret": "TheClientSecretFromAzurePortal",
 "CallbackPath": "/signin-oidc-custom"
},
```

   Some of these terms have changed over time. Below is a short list of alternative names for each setting:

   ◦

ClientId: AppID, Application client ID

- TenantId: Directory ID

1. Enter the PublicUrl path if ELO Sync should be called from a domain other than the internal one, e.g. via a proxy.

```
"PublicUrl": "https://domain:port/path/to/elosync",
```

1. Change the logon information for the service user that will be used to connected to the ELO repository:

```
"ServiceUser": {
"UserName": "ELO Service",
"Password": "ThePasswordForTheServiceUser"
}
```

1. Configure the ELO repositories to be accessed via ELO Sync:

```
"Repositories": [
{
  "name": "Display Name for Repository",
    "key": "TechnicalKeyForRepository",
    "url": "https://elo-example-server.com:9093/ix-Repository/ix",
    "webclienturl": "https://elo-example-server.com:9093/ix-Repository/plugin/de.elo.
    "oauthcallbackurl": "https://elo-example-server.com:9093/ix-Repository"
}
]
```

The technical key for the repository can be any character except a space (' '). We recommend using the repository name unless this name is not unique on all servers.

1. If necessary, change the profile key for the ELOauth plug-in. This is used if the user logs on to the ELO Sync web interface.

```
"OAuth": {
"ConfigId": "elo_sync_oauth"
}
```

1. Enter the database type and the connection string.

```
"Database": "Postgres",
"ConnectionStrings": {
"Sqlite": "Data Source=elosync.db",
"Postgres": "User ID=dbuser;Password=dbpassword;Server=dbserver;Port=5432;Database=el
"Oracle": "Data Source=elosyncdb;User Id=dbuser;Password=dbpassword;Integrated Securi
```

```
"MsSql": "Server=dbserver,1433;Database=elosyncdb;User Id=dbuser;Password=dbpassword;"
},
```

1. Configure the web server endpoints:

```
"Kestrel": {
  "Endpoints": {
    "Https": {
    "Url": "https://elo-sync-server",
    "Certificate": {
        "Path": "/path/to/certificate.pem",
        "Password": "PasswordForCertificate"
    }
    }
},
},
```

ELO Sync is now installed and fully configured.

To test ELO Sync, open a terminal in the ELO Sync installation directory and execute the following command:

```
./Elo.Sync.Main
```

You can find more details about configuration in the *appsettings.json* (or *appsettings.Production.json*) file in the Configuration section.

## Configuration of ELO Auth (ELO 23.6 or lower)

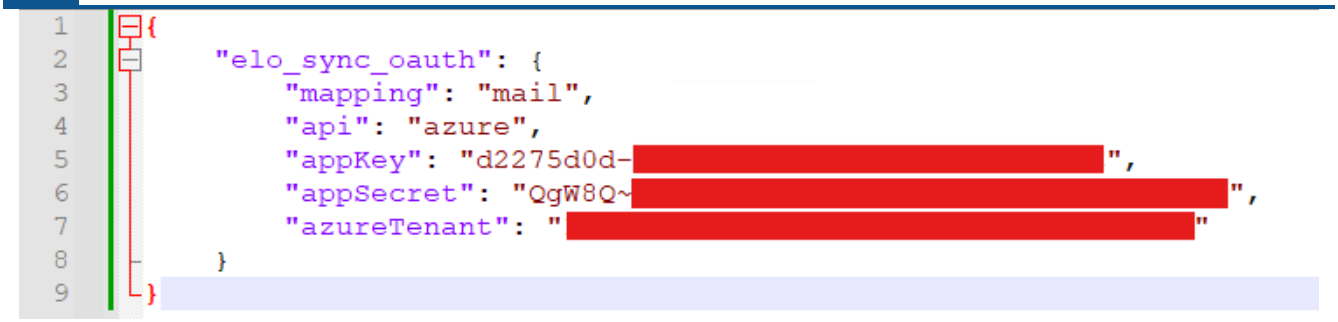
To use ELO Auth (Auth 1), an OAuth profile must be created. *The appsettings.json* file must be adjusted as follows: The `"loginmode"` variable must be incorporated into the "Repositories" node, if it does not already exist. The value `"ELOauth"` must be entered for ELO Auth.

```
36   "Repositories": [
37     {
38       "name": "Repository Name",
39       "key": "",
40       "url": "",
41       "webclienturl": "",
42       "oauthcallbackurl": "",
43       "loginmode": "ELOauth"
44     }
45   ],
```

If you already have an OAuth profile, an additional configuration node *"elo_sync_oauth"* must be added to the file *de.elo.ix.plugin.auth.json*. This file is located under `<InstallDir>\<ELOInstallFolder>\config\ix-<repository name>\ELO-<server name>-1\de.elo.ix.plugin.auth.json`.

```
1  {
2      "azuread": {
3          "mapping": "mail",
4          "api": "azure",
5          "appKey": "cb359113-[redacted]",
6          "appSecret": "f3f8Q[redacted]",
7          "azureTenant": "[redacted]"
8      },
9      "elo_sync_oauth": {
10         "mapping": "mail",
11         "api": "azure",
12         "appKey": "d2275d0d-[redacted]",
13         "appSecret": "QgW8Q~[redacted]",
14         "azureTenant": "[redacted]"
15     }
16 }
```

If you do not have an OAuth profile, you must create the file *de.elo.ix.plugin.auth.json* and insert the configuration node *"elo_sync_oauth"*.




**Please note**

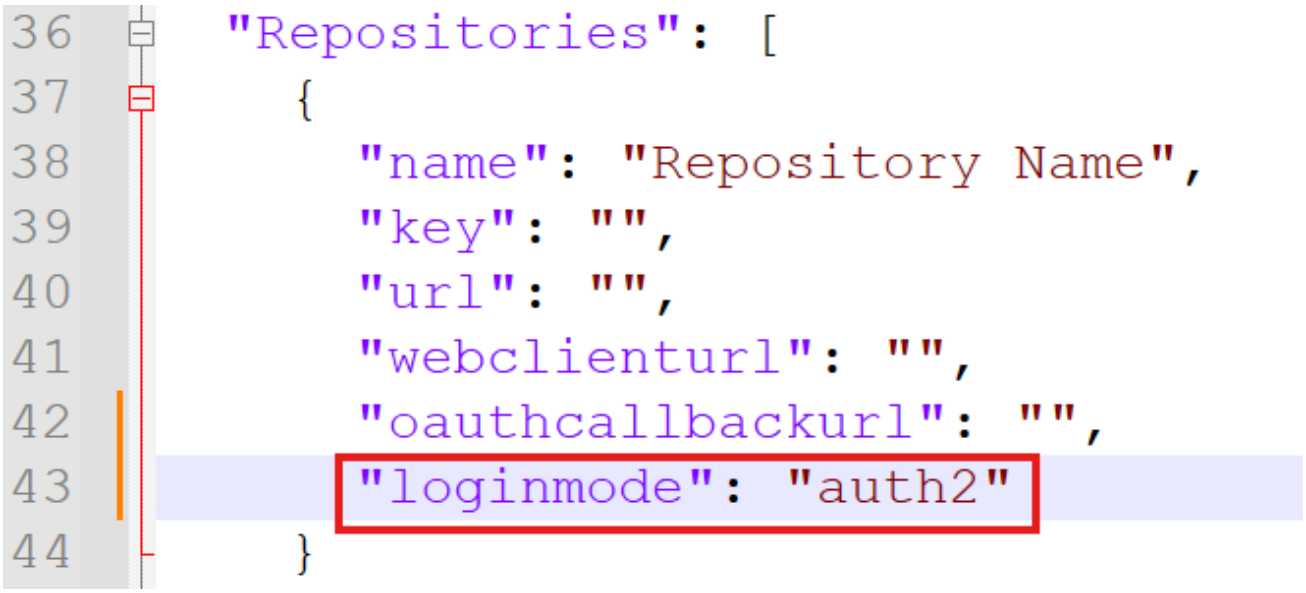
The following data must be taken from the ELO Sync *appsettings.json* (*ClientId*, *TenantId* and *ClientSecret*):

- "ClientId" -> "appKey"
- "TenantId" -> "azureTenant"
- "ClientSecret" -> "appSecret"

The *ConfigID* name in the ELO Sync *appsettings.json* under the *OAuth* node must be identical to the node name in the configuration file for OAuth (Example: "elo_sync_oauth").

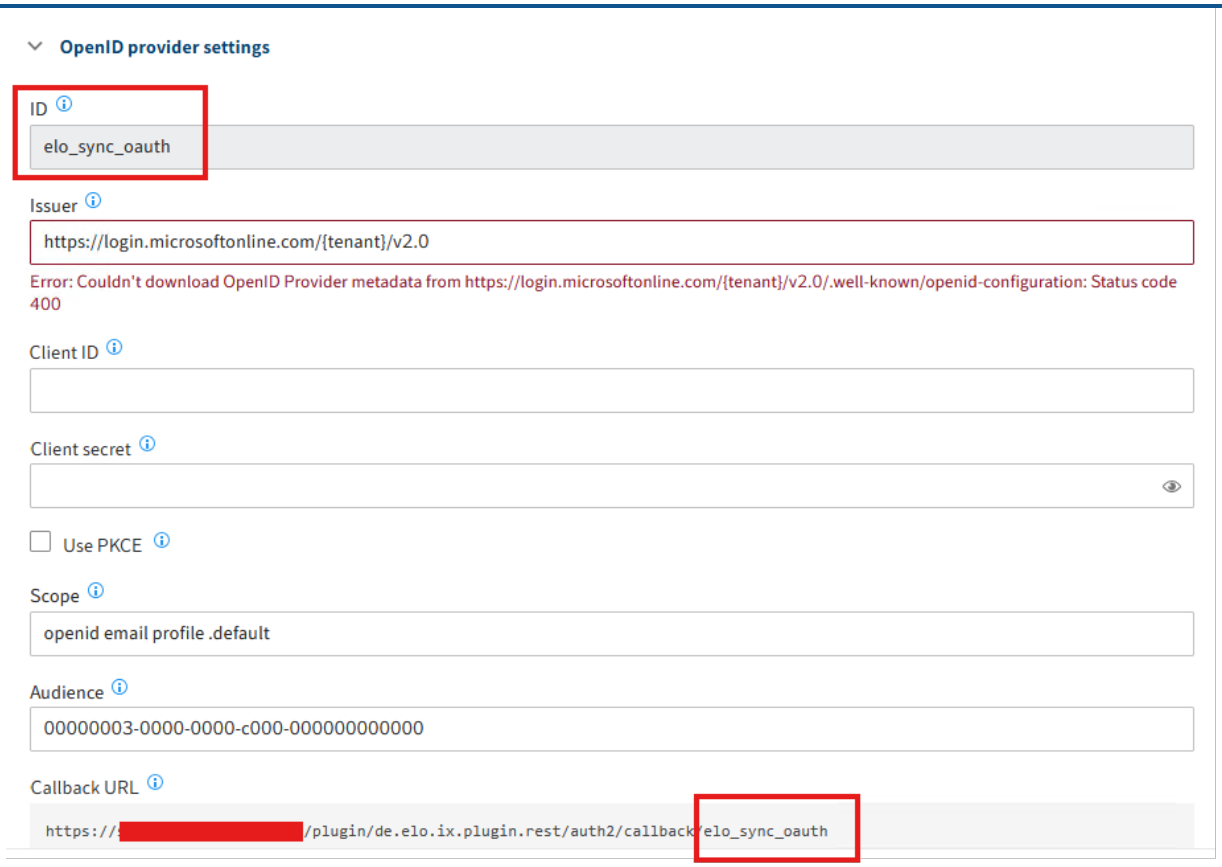
# Configuration of ELO Modern Authentication (Auth2) (ELO 23.6 or higher)

To use ELO Modern Authentication (Auth 2), an OAuth profile must be created. *The appsettings.json* file must be adjusted as follows: The `"loginmode"` variable must be incorporated into the "Repositories" node, if it does not already exist. The value `"auth2"` must be entered for ELO Modern Authentication (Auth2).




A new OpenID provider must be added for ELO Sync. Refer to the Add OpenID provider documentation for more information.

1. Select *Microsoft* from the drop-down menu.
2. Enter the name **elo_sync_oauth*.




The interface may look different depending on the version, but you can find the ConfigID using the callback URL. In our example, it is *elo_sync_oauth*.

3. Enter the values from the ELO Sync *appsettings.json* in this dialog box:
   - "ClientId" -> Client ID
   - "TenantId" -> {Tenant}
   - "ClientSecret" -> Client secret
4. In the *Audience* field, enter the value `00000003-0000-0000-c000-000000000000` (Information: This value is the GUID for the Graph API).

# Service registration

After you have installed ELO Sync, you can register it as a system service, so that it starts automatically with the operating system.

## Windows

To register ELO Sync as a Windows service, execute the following command in PowerShell as an administrator:

```
New-Service -Name ELOSync -DisplayName "ELO Sync" -Description "ELO Sync provides synchroni
```

> **Information**
>
> This command runs the ELO Sync service under the SYSTEM account of the Windows installation.

## Linux

Create an *elosync.service* systemd unit file for the ELO Sync service with the following content:

```
[Unit]
Description=ELO Sync provides synchronization between ELO repositories and third-party syst

[Service]
WorkingDirectory=/path/to/elosync/
ExecStart=/path/to/elosync/Elo.Sync.Main

# Always restart if the service exits
Restart=always
RestartSec=5

KillSignal=SIGINT
Environment=ASPNETCORE_ENVIRONMENT=Production
```

> **Information**
>
> The ELO Sync service is then executed as *root* with these instructions.
>
> This is not recommended, but setting up and configuring a service user account is not part of this guide.
>
> You can find information on this in the documentation for your Linux distribution.

Open a terminal in the directory where you created the *elosync.service* file and execute the following commands:

```
sudo cp elosync.service /etc/systemd/system/elosync.service
sudo systemctl daemon-reload
sudo systemctl start elosync.service

# If you want to enable auto start execute the following
sudo systemctl enable elosync.service
```

# Update guide

## Manual update of ELO Sync without the ELO Server Setup

To update ELO Sync manually, first the ELO Sync service has to be stopped.

After this, the files have to be extracted from the ELO Sync ZIP file to the ELO Sync installation directory. Existing files should be deleted to avoid inconsistencies. Only the *appsettings.json* file with the configuration data may be kept.

After it has been ensured that the data in the *appsettings.json* is correct and e.g. the specified certificate exists at the corresponding path, the service can be started again. The manual update is then completed.

> **Information**
>
> We recommend renaming the existing installation directory and copying the ELO Sync files into a new folder with the name of the old installation. As a result, both versions are retained temporarily, and configuration files such as the *appsettings.json* can easily be switched out. This is recommended especially for major version upgrades, as the structure of the configuration files may have changed or new properties may have been added. The old and new *appsettings.json* files can then easily be compared and populated accordingly.

## Update from 23.6 to 25.0

### High-impact changes

#### Conflict types added

- The conflict types have been simplified for use with the REST API.

  The new properties 'FirstSystem' and 'SecondSystem' were added, which make it possible to identify the respective system of the items directly in a conflict. These fields may not be populated yet for older conflicts. Running the synchronization job again populates these additional conflict fields.

#### Conflict resolutions simplified

- Conflict resolution has been simplified. Now, only the item ID has to be entered without the system prefix.

#### OracleDB no longer supported

- OracleDB is no longer supported in 25.0 due to low demand.

## Update from 23.4 to 23.5

### High-impact changes

#### Minimum database versions increased

- The minimum required version of Microsoft SQL Server was increased to SQL Server 2016.

  This is due to a change by Microsoft in a dependency used by ELO Sync. You will find more information under Microsoft Docs.

- The minimum required version of Oracle Database was increased to 19c

  Oracle has discontinued support for older database versions with the introduction of support for .NET 8. You will find the detailed system requirements in the Oracle Developer Guide under *ODP.NET Core*.

- The minimum required version of PostgreSQL was increased to 12.20

  This is in line with the standard support policy for PostgreSQL. See the Versioning Policy for supported PostgreSQL versions.

#### ConnectionString change required for SQL Server when using self-signed certificates

If ELO Sync is unable to connect to the SQL Server database, it may be necessary to update the ConnectionString in the `appsettings.json` file.

If the ELO Sync log file contains the following error (or a similar message), the trust settings for the server certificate have changed.

```
<details>
<summary>Microsoft.Data.SqlClient.SqlException</summary>

A connection was successfully established with the server, but then an error occurred durin

</details>
```

This error occurs due to a change in the default behavior when connecting to SQL Server. The server certificate now has to be signed by a trusted certificate authority.

Self-signed certificates as created by the standard SQL server setup are no longer accepted.

As a workaround, you can add `TrustServerCertificate=true;` to the ConnectionString to re-enable trust in self-signed certificates.