SAML SINGLE-SIGN-ON WITH LISTSERV

Last updated on Apr 22, 2025. This document is a complete guide to LISTSERV 17.5's SAML Single Sign-On feature.

All of L-Soft's manuals are available at the following URL: https://www.lsoft.com/manuals.html

TABLE OF CONTENTS

1 INTRODUCTION

SAML (Security Assertion Markup Language) is an XML-based markup language for exchanging authentication and authorization information between an Identity Provider (IdP) and a Service Provider (SP).

LISTSERV's SAML Single-Sign-On support allows site administrators to add external identity provider(s) to LISTSERV for user authentication on LISTSERV's web interface, thus enhancing security and user experience. The SAML feature is provided by a separate package (i.e., not bundled with LISTSERV kits) that needs to be installed and configured independently. This document has detailed instructions for the SAML component (version 1.0.0) installation and configuration.

2 PREREQUISITE

In order to use SAML with LISTSERV, the hosting server and the relevant identity provider(s) must meet the criteria mentioned below.

The hosting server:

- Must have LISTSERV 17.5 (or a later version).
- Must have the TCPGUI port (default is 2306) open to the SAML application.
- Should run LISTERV and SAML on the same local network.
- Should have set up HTTPS for outgoing traffic.

The identity provider:

- Must support SAML protocol with HTTP Redirect binding.
- Must specify user's email address in the Subject's `NameID` field of the SAML response/assertion.
- Must sign SAML responses.

3 INSTALLATION

LISTSERV's SAML feature is distributed as a separate add-on package. The installation procedure, as detailed below, involves downloading the SAML package and setting up your web server for the SAML application.

In order to host and run the SAML application, you need to install .NET Core. The procedure is different for Windows and Linux.

3.1 WINDOWS

On Windows, the SAML application runs along with IIS through the ASP.NET Core Module from the .NET Core Hosting Bundle.

3.1.1 .NET CORE 8 HOSTING BUNDLE AND APPLICATION INITIALIZATION FEATURE

First, download and install .NET Core 8 hosting bundle: .NET Core Hosting Bundle 8.0.12. For troubleshooting regarding the hosting bundle, please refer to Hosting Bundle | Microsoft Learn.

> When the manual was written, the latest .NET Core Hosting Bundle version was 8.0.12. You may also use any other .NET Core 8.0 *Hosting Bundle* listed under:
>
> https://dotnet.microsoft.com/en-us/download/dotnet/8.0

Then, enable IIS's "Application Initialization" feature. On Windows Server, this is usually done via Server Manager's "Add Roles and Features Wizard", and the feature can be found in "Server Roles" > "Web Server (IIS)" > "Web Server" > "Application Development" > "Application Initialization".

3.1.2 DOWNLOAD AND UNZIP SAML PACKAGE

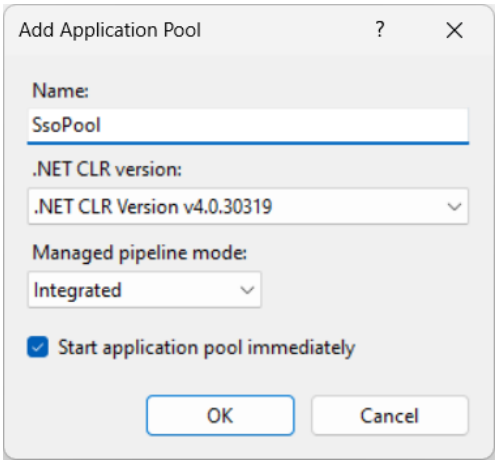
Next, download the SAML package for Windows with the link below and unzip it to a desired location on the server where LISTSERV is running on, for example, `<drive>:\LISTSERV\SAML`.

https://dropbox.lsoft.com/download/listserv-saml-net-win-x64.zip

Make sure that IIS_IUSRS has *read, write, modify* and *execute* permissions on the SAML folder.

3.1.3 CREATE A NEW APPLICATION POOL IN IIS

In IIS, right click on "Application Pools" to "Add Application Pool..." and create an app pool as follows:



In the “Actions” panel on the right, under “Edit Application Pool”, click on “Advanced Settings...”, set the following options:

- General > Start Mode: AlwaysRunning
- Process Model > Idle Time-out (minutes): 0
- Process Model > Load User Profile: True

3.1.4 CREATE IIS SITE APPLICATION

In IIS, right-click on the LISTSERV website to “Add Application...”:

- Set the application alias to “SSO” (or something that you prefer)
- Set the physical path to where the SAML package is located, e.g., “<drive>:\LISTSERV\SAML”
- For Application pool, choose “SsoPool”

Choose the new SSO application, click on “Advanced Settings...” from the “Actions” panel, set the following option:

- Preload Enabled: True

After all the setup is complete, SAML endpoints will be accessible via the specified alias, e.g., https://*listserv.example.com*/sso. This URL is referred to as “ServiceUrl” below.

Make a note of this URL. You will need to use this URL later in the config file samlsettings.json.

3.2 LINUX

For Linux environment, we provide both self-contained and platform-dependent packages ending with `-net` and `-nonet` respectively:

- https://dropbox.lsoft.com/download/listserv-saml-net-linux-x64.tar.gz (self-contained)
- https://dropbox.lsoft.com/download/listserv-saml-nonet-linux-x64.tar.gz (platform-dependent)

The installation process may vary depending on the package you use. In most cases, the self-contained package works out-of-the-box: it suffices to expand the package and grant necessary permissions.

If you wish to use the platform-dependent package which requires you to install a separate .NET runtime, please refer to Section 9.2 to install necessary dependencies. The rest of the procedure is identical for self-contained and platform-dependent packages.

After finishing extracting the tarball into a desired location, you will need to configure a reverse proxy (see Section 3.2.2) to bridge the SAML application (usually listening on the `localhost`) and the external network.

3.2.1 GENERAL INSTALLATION PROCEDURE

The recommended approach is to extract the tarball into `/home/listserv/saml` and change the ownership of the entire directory recursively to `listserv:listserv`. Please ensure that the user `listserv` has RWX permission to the SAML executable and all directories as well as RW access to all files under `/home/listserv/saml`.

Once the package has been installed, you may run

```
./SAML
```

at the installation directory to verify the initial setup.

An example systemd unit file `listserv-saml.service` can be found under the `etc/` directory and should be copied into `/etc/systemd/system/` . Parameters may need to be adjusted accordingly if the SAML application is installed in a location other than `/home/listserv/saml` or is run as a user other than `listserv`.

For suggestions to run SAML inside a container, please refer to Section 9.1.

3.2.2 SETTING UP REVERSE PROXY

The following is an example of the reverse proxy setup for LISTSERV SAML extension in an existing virtual host on Apache HTTP Server.

httpd.conf

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
...
<VirtualHost *:443>
  ServerName listserv.example.com
  ...
  ProxyPass "/sso/" "http://localhost:5000/"
  ProxyPassReverse "/sso/" "http://localhost:5000/"
</VirtualHost>
```

Note that both `proxy` and `proxy_http` modules should be loaded.

The configuration above only serves as a reference, and the actual configuration may vary based on distros or the choice of web servers. By default, the SAML application listens on localhost:5000. If you wish the application to be listening to a different address, you may customize the `LocalBinding` property listed in Section 7.2 *Optional configuration variables*.

In the rest of the documentation, we will assume that the reverse proxy has been setup to proxy the application at *https://listserv.example.com/sso*. Note that web servers on Linux tend to be case-sensitive by default, so it is important to use a consistent casing convention through the setup.

4 TEST SETUP

To test if the environment has been configured properly, you can copy the example `samlsettings.json` from the `etc/` subdirectory into the root directory of the SAML application.

> In the rest of the documentation, we will use *listserv.example.com* as the hostname. Readers should substitute the hostname accordingly.

Note: Files under the `etc/` subdirectory are **not** read by the application and should be used for reference purposes only. When you want to make configuration changes, make sure to edit the files in the SAML root directory, not the one in `etc`/.

4.1 START SAML APPLICATION

On Windows, the application can be started from IIS with "Application Pools > SsoPool > Start".

On Linux, it is recommended to `cd` into the SAML directory and run the SAML executable directly in the console during the setup phase. When the configuration is ready for production, one should consider using the systemd unit file included under `etc/` to manage service start/restart automatically.

4.2 QUICK TEST

Once the application has been started, you should be able to access https://*listserv.example.com*/sso/setup.

The page will display some configuration information in JSON format that will help you with the rest of the setup procedure.

> **Tip for Linux installation:** With the presence of reverse proxy, it can be sometime tricky to isolate SAML application misconfiguration from the reverse proxy. One can validate the SAML configuration separately by making a GET request to the local address directly, e.g.,
>
> ```
> curl localhost:5000/setup
> ```

5 INITIALIZATION AND CONFIGURATION

The SAML add-on is configured via file "`samlsettings.json`" (the one in the SAML root directory). It is a standard JSON file. There're some pre-configured settings in the config file. See Section 7 for details about each configuration variable.

Initialization requires multiple steps that are detailed below.

5.1 ADJUST CONFIGURATIONS IN SAMLSETTINGS.JSON

In `samlsettings.json`, fill in "`WaUrl`" with the URL to LISTSERV web interface and "`ServiceUrl`" with the location where SAML endpoint is hosted/reverse proxied. You can leave the `IdpProfiles` field empty for now.

Here are examples for `samlsettings.json` for Windows and Linux respectively.

samlsettings.json (Windows example)

```
{
  "WaUrl": "https://listserv.example.com/scripts/WA.EXE",
  "ServiceUrl": "https://listserv.example.com/sso",
  "IdpProfiles": [],
  "EnableSetupPage": true
}
```

samlsettings.json (Linux example)

```
{
  "WaUrl": "https://listserv.example.com/cgi-bin/wa",
  "ServiceUrl": "https://listserv.example.com/sso",
  "IdpProfiles": [],
  "EnableSetupPage": true
}
```

Restart the SAML service/application.

5.2 CONFIGURE YOUR IDENTITY PROVIDER

The procedure and nomenclature may be different with different identity providers. In general, you start by creating an SAML application within the identity provider portal. You may use information at

https://*listserv.example.com*/sso/setup

or the SP metadata at

https://*listserv.example.com*/sso/saml/metadata

to help with the initial setup of IdP.

Some IdPs require that you manually fill in those settings, while some IdP accepts SP metadata upload. You should refer to the identity provider's documentation to set up.

> **Note**: Make sure to use EmailAddress for Name ID format. Different IdPs may have a different nomenclature for this setting.

After you have obtained your IdP's Metadata URL, you can proceed to the next section and provide this URL to the SAML application.

5.3 ADD YOUR IDENTITY PROVIDER INFORMATION TO SAMLSETTINGS.JSON

SAML identity providers are configured via the `IdpProfiles` variable in `samlsettings.json`. This is where to use the IdP metadata.

Here's an example that defines two IdP profiles, OKTA and Entra ID (where Entra is disabled):

samlsettings.json (partial view)

```
"IdpProfiles": [
  {
    "Name": "OKTA",
    "Id": "okta",
    "MetadataUrl": "https://dev-xxx.okta.com/app/xxx/sso/saml/metadata"
  },
  {
    "Name": "Entra ID",
    "Id": "entraid",
    "MetadataUrl": "https://login.microsoftonline.com/xx-xxx/federationmetadata..",
    "SLO": false,
    "Enabled": false
  }
]
```

- `Name`: A friendly name for this identity provider. This will be the display name of the login entry point on LISTSERV's login page, as part of the login button's text label.
- `Id`: A unique alphanumerical ID to distinguish this IdP from other IdPs. This ID will be used to construct logon URLs and will be invisible to end users.
- `MetadataUrl`: The IdP's SAML metadata URL provided by your identity provider. The SMAL application will load the IdP information from there.
- (Optional) `SLO`: A Boolean value of whether to enable Single Logout. The default value is false.
- (Optional) `Enabled`: A Boolean value of whether to enable this IdP in LISTSERV SAML. The default value is true.

Although usually you only need one SAML identity provider, it's possible to configure multiple IdPs to be used by LISTSERV for Single Sign-On, as shown in the example above. To add an additional identity provider, one can add a profile with relevant settings to the list, separated by comma.

> **NOTE**: Any change to `samlsettings.json` require an application restart to be applied.

5.4 FINISH IDP CONFIGURATION IF NECESSARY

Now that the IdP Metadata has been provided, the SAML application will generate an updated version of setup information:

https://*listserv.example.com*/sso/setup

If the IdP requires additional information from SP, or if you want to enable SLO (as detailed in 5.8), you will need the updated setup page and/or Metadata to finish the configuration on the identity provider side.

5.5 INITIALIZE SAML AND LISTSERV THROUGH SAML.MERGE.JSON

> **Note**: Before running init or update, we highly recommend backing up the current template file `site.wwwtpl` in LISTSERV's `MAIN` (for Windows) or `home` (for Linux) directory.

After the `samlsettings.json` is in place, LISTSERV SAML uses a one-time file `saml.merge.json` to perform initialization and synchronize changes to LISTSERV.

In the SAML directory, create a file called `saml.merge.json` that contains the text below.

saml.merge.json

```
{
    "Email": "pm@example.com",
    "Password": "password",
    "Action": "init"
}
```

Use your LISTSERV Postmaster account for "`Email`" and "`Password`". Leave the "`Action`" field unchanged.

> **Note**: If your LISTSERV instance is configured with LDAP authentication, the "`init`" action will not apply to you. Please follow Section 10 instead to use "`adduser`" and "`update_sso`" actions.

This file will be loaded automatically when the SAML application (re)started, and it will be removed on success. The "`init`" action performs a series initialization including creating an SSO user, set up relevant LISTSERV site variables to prepare for SAML, and populate necessary web templates. The postmaster credential is used to authorize those privileged changes within LISTSERV, and LISTSERV will be restarted automatically (unless you have "`AutoRestartLsv`" set to false, which means you will have to manually restart LISTSERV at a proper time in order for the changes to take effect).

> **TIP**: During the installation stage, we recommend you make a copy of this file, as it tends to require a few modifications when configurating the SAML application for the first time.

Once the initialization has been done, you may use the method in 4.2 to validate again.

5.6 UPDATE LISTSERV'S LOGIN PAGE USING SAML.MERGE.JSON

We recommend performing a synchronization each time you modify the `samlsettings.json` for any possible changes to be reflected on the web interface. This can be done in similar way to the initialization procedure by creating a `saml.merge.json` file with the `Action` property set to "`update_sso`" instead.

saml.merge.json

```
{
    "Email": "pm@example.com",
    "Password": "password",
    "Action": "update_sso"
}
```

5.7 (OPTIONAL) HIDE LISTSERV'S NATIVE PASSWORD LOGIN

It's possible to hide LISTSERV's native login form as well as password registration form from the web interface. To do this, add a line in samlsettings.json with the "NoLsvLogin" variable.

samlsettings.json (partial view)

```
{
  …other settings…
  "NoLsvLogin": true,
}
```

And once again, use saml.merge.json with "update_sso" or "update" action to trigger an update for LISTSERV's web template.

> Note: Hiding the login forms doesn't mean that registration and login functionalities are disabled completely.

To put the email address and password form back, set "NoLsvLogin" to false or simply remove that line from samlsettings.json. The rest of the procedure will be the same.

5.8 (OPTIONAL) ENABLE SINGLE LOGOUT

SAML Single Logout (SLO) allows users to log out from applications they previously logged into using SSO. There are two scenarios where SLO is involved:

- User logs out from LISTSERV where they previously logged in via SSO. In this case, SLO will log the user out from other SSO-logged-in applications automatically.
- User logs out from a different application where they previously logged in via SSO. When the user accesses a LISTSERV web page afterwards, they will be automatically logged out.

To enable SLO with a certain IdP, simply set the "`SLO`" variable to `true` for that IdP profile as shown below:

samlsettings.json (partial view)

```
 "IdpProfiles": [
{
     "Name": "Entra ID ",
     "Id": "entraid",
     "MetadataUrl": "https://login.microsoftonline.com/xx-xxx/federationmetadata..",
     "SLO": true,
     "Enabled": false
   }
 ]
```

IdP usually requires SLO requests/responses to be signed by standard, which would require you to upload the certificate to verify those messages. The certificate can be obtained by copying the data under the **X509Certificate** element in:

https://*listserv.example.com*/sso/saml/metadata

and wrap it in:

```
-----BEGIN CERTIFICATE-----
<Certificate data goes in here>
-----END CERTIFICATE-----
```

6 MISCELLANEOUS NOTES

6.1 LOGGING

The SAML add-on has its own log file and is rotated on a daily basis. SAML log files, called "`listserv-saml-yyyymmdd.log`", can be found in the /logs directory of where the SAML application is installed.

Generally, log events will be one of the following levels: Debug, Information, Warning, Error.

6.2 LIMITATIONS

6.2.1 LDAP

In SAML v1.0.1, we introduced support for LISTSERV instances configured with LDAP authentication. However, additional procedure is needed during the initialization phase. Details can be found in Section 10.

6.2.2 TOMCAT

The SAML application, which is written in ASP.NET Core, is NOT natively supported by Maestro's Tomcat. If your LISTSERV web interface is hosted by Maestro's Tomcat instead of a general-purpose web server such as IIS or Apache, you could deploy a web server to host the SAML app and put the rest behind this web server via reverse proxy or use LISTSERV Maestro's built-in SSO functionality instead.

6.2.3 HIDEOPTXY

LISTSERV 17.5's new feature to hide X and Y tokens from URLs is incompatible with the SAML application. This feature is disabled by default, so this will generally not cause any issue. However, you should avoid turning HIDEOPTXY on whenever you have SSO/SAML enabled with LISTSERV.

6.2.4 REVERT JS-GLOBAL WEB TEMPLATE

LISTSERV SAML appends necessary template modification to the `JS-GLOBAL` web template. When using the "`REVERT`" feature provided by `saml.merge.json`, the SAML application *only* deletes the part that was previously added by itself instead of reverting the template to the system default. If you wish to revert the template to the system default after a LISTSERV upgrade, you need to revert it manually.

7 APPENDIX – CONFIGURATION SETTINGS IN SAMLSETTINGS.JSON

> **Note**: Configuration variable names are case-insensitive.

7.1 REQUIRED CONFIGURATION VARIABLES

Name	Type	Description
ServiceUrl	String	The full URL to the SAML application. E.g., "https://*listserv.example.com*/sso"
WaUrl	String	The full URL to LISTSERV's CGI script. E.g., "https://*listserv.example.com*/scripts/wa.exe" (Windows) or "https://*listserv.example.com*/cgi-bin/wa" (Linux)
IdpProfiles	List of `IdpProfile`	A list of IdP profiles to be used by LISTSERV. See 7.1.1.

7.1.1 IDPPROFILE CONFIGURATION VARIABLES

"`IdpProfiles`" is a list of objects of "`IdpProfile`" type. Each `IdpProfile` has the following properties:

Name	Type	Description
Name	String	A friendly name for this identity provider.
Id	String	A unique identifier made up of alphanumerical characters
MetadataUrl	String	Identity provider's metadata URL.
SLO	Bool	(Optional) Whether to enable Single Logout. Default is false.
Enabled	Bool	(Optional) Whether this IdP shall be enabled. Default is true.
BindingMethod	String	(Optional) Override the method for submitting AuthnRequests. "Redirect" for HTTP-Redirect, "Post" for HTTP-POST binding. By default, HTTP-Redirect is preferred.
WantAuthnRequestsSigned	Bool	(Optional) Override whether AuthnRequests will be signed when the IdP Metadata doesn't conform to the standard. By default, the WantAuthnRequestsSigned attribute in IdP's metadata is respected.

7.2 OPTIONAL CONFIGURATION VARIABLES

Name	Type	Description
AcsRoute	String	The route for SAML assertion consumer service. Default is "/saml/post".
AutoRestartLsv	Bool	Whether or not to allow SAML initialization process to restart LISTSERV automatically after changes to system variables and templates are made. If disabled, a manual restart of LISTSERV will be needed. Default is true.
DEBUG_FLAGS	Enum	A collection of debug options. See 7.3.
EnableSetupPage	Bool	Enables the page at https://<hostname>/sso/setup to facilitate SAML setup on the identity provider side. Default is false.
KeepSamlMerge	Bool	Whether or not to keep the saml.merge.json file on disk. Default is false, i.e., the file is automatically deleted every time after it's loaded.
LocalBinding	String	(*Reverse Proxy Only*) The address and port for the SAML application to listen to. This is where the **ServiceUrl** should be mapped into. Default is "127.0.0.1:5000".
MetadataRoute	String	The route for SAML SP Metadata. Default is "/saml/metadata".
NoLsvLogin	Bool	Whether or not to hide LISTSERV's native (email address and password) login. Default is false.
TcpguiHost	String	The hostname for TCPGUI connections. Default is "localhost".
TcpguiPort	Int	The port for TCPGUI connections. Default is 2306.
ReadXForward	Bool	(*Reverse Proxy Only*) Whether to respect the X-Forward header. By default, this value is false on Windows, but true on Linux.

7.3 DEBUG FLAGS

Debug flags are toggled with the "DEBUG_FLAGS" variable in samlsettings.json. There's a connection of debug flags that you can turn on/off:

Name	Description
LOG_ALL_CONFIGS	Print loaded SAML configurations on start.
LOG_TCPGUI	Display TCPGUI transactions with LISTSERV, including commands sent to LISTSERV and responses received.
LOG_SAML_PROTOCOL	Display information between SAML SP and IdP transactions.
LOG_TCPGUI_PERF	Show performance details about TCPGUI.

By default, no debug flag is enabled, which is equivalent to:

```
"DEBUG_FLAGS": "none"
```

To enable one or more debug flags, simply set the value to your desired flag or a comma separated list of flags, e.g.,

```
"DEBUG_FLAGS": "LOG_ALL_CONFIGS, LOG_TCPGUI"
```

Below is a syntax sugar to enable all debug flags without explicitly specifying all flags:

```
"DEBUG_FLAGS": "all"
```

> **Note**: Enabling debug flags could make log files grow large very quickly. It's not recommended to enable all debug flags in a production environment all the time.

8 APPENDIX – SAML.MERGE.JSON

The purpose of file "`saml.merge.json`" is to synchronize SAML settings with LISTSERV's. The file is generally used in the following scenarios:

8.1 CASE 1: SAML INITIALIZATION

When you are setting up LISTSERV's SAML for the first time, there are a few configuration variables and templates in LISTSERV that need to be initialized. To do this, you need to put your LISTSERV Postmaster credential into the file and set "`Action`" to "`init`". For example:

saml.merge.json

```
{
    "Email": "pm@example.com",
    "Password": "password",
    "Action": "init"
}
```

Your Postmaster account is used to authorize this configuration change in LISTSERV.

8.2 CASE 2: SAML IDP PROFILES UPDATED

In cases where you make any changes to the `IdpProfiles` setting in `samlsettings.json` (e.g., switching from one IdP to another, or adding/removing an IdP), some LISTSERV web templates must be updated for those changes to be reflected on LISTSERV's login page.

This is very similar to initialization. The only difference is that you put "`update_sso`" (as opposed to "`init`") in the "`Action`" field. Here's an example:

saml.merge.json

```
{
    "Email": "pm@example.com",
    "Password": "password",
    "Action": "update_sso"
}
```

8.3 CASE 3: REVERT SAML CHANGES

If you want to revert LISTSERV template changes made by the SAML application, use the same `saml.merge.json` file and set "`Action`" to "`revert`". The rest is the same.

saml.merge.json

```
{
    "Email": "pm@example.com",
    "Password": "password",
    "Action": "revert"
}
```

8.4 CASE 4: REGISTER THE SPECIAL SSO USER ACCOUNT

The service account ssouser-xxxxxx@hostname is usually registered automatically during the “init” process. However, there may be situations where you would want to re-register this account or do this separately.

In such cases, use the “adduser” action. Note that after the “adduser” action is complete, LISTSERV will need to be restarted either automatically (which is the default behavior) or manually.

saml.merge.json

```
{
    "Email": "pm@example.com",
    "Password": "password",
    "Action": "adduser"
}
```

9 APPENDIX – ADDITIONAL REFERENCE FOR SAML INSTALLATION ON LINUX

9.1 DOCKER

This section is intended for system administrators with experience in Docker containers. L-Soft support generally is unable to assist with troubleshooting problems encountered when using Docker.

Below is an example Dockerfile to create an image of the SAML application (you can also find it under the `etc/` directory in the package):

Dockerfile

```
FROM mcr.microsoft.com/dotnet/aspnet:8.0-noble-chiseled
USER $APP_UID
WORKDIR /
ADD --chown=$APP_UID:$APP_UID saml-nonet.tar.gz /app
EXPOSE 5000
VOLUME [ "/app/data" ]

WORKDIR /app
ENTRYPOINT ["./SAML"]
```

You should consider binding the following file and folders and granting them necessary permission to provide the configuration file to the container and persist logs and data:

- **samlsettings.json:/app/samlsettings.json**
- **logs/:/app/logs/**
- **data/:/app/data/**

We recommend copying the `saml.merge.json` (5.5) only when needed and restarting the container for initialization/update.

9.2 DEPENDENCIES FOR THE PLATFORM-DEPENDEPNT PACKAGE

This section only applies if you use the platform-dependent (`listserv-saml-nonet-linux-x64`) package.

In case that the bundled .NET framework is not compatible with your environment, you may use the platform-dependent package. To use this, you need to ensure that both .NET 8 and ASP.NET runtimes are present in your environment. You may refer to your package manager or the official installation guide https://learn.microsoft.com/en-us/dotnet/core/install/linux for assistance.

We have included the following are installation commands on common distros for your reference.

Ubuntu	RHEL
`apt install aspnetcore-runtime-8.0`	`dnf install aspnetcore-runtime-8.0`

10 APPENDIX – INITIALIZE SAML WITH LDAP CREDENTIAL

> This requires SAML v1.0.1 (or later).

During SAML initialization, the SAML program automatically creates a special user in LISTSERV to delegate authentication. However, when LDAP authentication is enabled in LISTSERV, i.e., `LDAP_PW_SERVERS` is defined, it will NOT be able to create this special user due to LDAP restrictions. In this case, you must manually provide an existing LDAP user account for this purpose to SAML instead. This user will be used to login to LISTSERV on behalf of other users.

> Do not confuse this user account with the Postmaster user used in saml.merge.json. The latter must be a privileged LISTSERV user, while the former doesn't need to be a Postmaster, although you can use the same account for both.

Here's the procedure to initialize SAML with your existing LDAP user:

1. Open `samlsettings.json`, add "`SsoUser`" and "`SsoUserAuth`" with the credentials of your LDAP user as following and save the file:

samlsettings.json

```
{
  ...
  "SsoUser": "<your_ldap_user_email>",
  "SsoUserAuth": "<your_ldap_user_password>",
}
```

2. Open or create `saml.merge.json`, use your LISTSERV Postmaster credential here and set "`Action`" to "`ADDUSER`".

saml.merge.json

```
{
    "Email": "pm@example.com",
    "Password": "password",
    "Action": "ADDUSER"
}
```

3. Restart the SAML program. (If on Windows, restart the application pool.)
4. Repeat step 2, but with "Action" set to "UPDATE". This step will trigger web template changes in LISTSERV, including adding a SSO link to the LISTSERV login page.

saml.merge.json

```
{
    "Email": "pm@example.com",
    "Password": "password",
    "Action": "UPDATE"
}
```

Once this is completed, you can go to Section 5.8 to proceed. At this point, you can remove "`SsoUser`" and "`SsoUserAuth`" from `samlsettings.json`.