Recast

Application Workspace Access Manager

Last Modified on 04.22.26

The Access Manager add-on is a completely integrated single sign-on solution that works with Application Workspace to provide simple, contextual, immediate access to the hundreds of applications.

OAuth2 Identity Provider for SSO

OAuth 2.0 / OpenID Connect are industry standards for exchanging identification and authorization data between trusted parties. Application Workspace implements an OAuth authorization server that also supports the OpenID Connect standards.

OIDC limitations

- Application Workspace supports OpenID Connect but does not implement the OIDC discovery endpoint.
- Administrators must configure relying-party components with the JWKS endpoint provided by Zone.
- Token validation assumptions based on discovery metadata do not apply, because there is no discovery endpoint.
- The typ numeric claim on the Token is Application Workspace's own discriminator, not the standard OIDC typ claim.

Settings screen

Allow requesting metadata – If enabled, external parties can receive the metadata of this identity provider. In most cases, it's recommended to allow this so that federations can be created quick and easily. This can always be disabled after setting up a federation.
Metadata URL – The URL where the metadata is published. Note: even if metadata queries are not allowed, an administrator can always download the metadata by using the download button. (.well-known/openid-configuration is also supported)

Clients screen

Clients must be defined before they can make use of this authorization server.

Create client dialog box

Name – The name of the client, visible for users when they log out.
Client ID – The OAuth 2.0 Client ID used to identify the Client with the Authorization Server.
Support OpenID Connect – If enabled, the *OpenID Connect* scope is supported by default for this client. You do not have to add it in the **Supported scopes** tab of the **Edit client dialog box**.

Edit client dialog box

Secrets tab

Client can use a secret for trusted requests to the authorization server. Keep this value safe as only the Client and Authorization Server knows it.
If the **Require secret for all token endpoint requests** checkbox is selected,
then every grant type (AuthorizationCode, RefreshToken and Password) on the token endpoint should provide a client secret.

Redirect urls tab

After a client requested an authorization code, the Application Workspace Server needs to send back the user. The redirect URLs option ensures that authorization codes are not shared with unwanted parties.

Example: `https://oauth-client.liquit.com/callback`

Supported scopes tab

A list of scopes that the client has access to.

Logout tab

Most OAuth 2.0 implementation do not support logout. You can still logout of a client by specifying a normal logout web page.

Logout URL – Insert the client logout URL. If the logout page supports redirection after logging out, the client logout URL should incorporate the variable "${slo.return.url". This variable will generate a URL that directs back to the Application Workspace, enabling the logout status to be reported to the logout page.
Logout reporting – If enabled, the logout page will wait for the status report. Otherwise the logout page assumes a successful logout after a few seconds.
Embedded logout supported – Enable this option if the client supports being logged out within an HTML iframe. If there are problems with logging out, we recommend you disable this option.

Access conditions tab

With access control you can limit which users can make use of this client. When the access is denied for a user, the user will be redirected to the requesting web application where an access denied error message will be displayed.

The filters available are the same as those for contexts.

Advanced tab

Access token section
Access token lifetime – The length of time an access token is valid. The default value is 1 hour.
Allow implicit Access Token – If enabled, the client can request an access token from the authorization endpoint and skip the authorization code part of the flow.

Refresh token section
Refresh token lifetime – The length of time an access token is valid. The default value is 90 days.

ID token section
ID token lifetime – The length of time an ID token is valid. The default value is 10 hours.
Allow implicit ID Token – If enabled, the client can request an ID token from the authorization endpoint and skip the authorization code part of the flow.

Code section
Code lifetime – The length of time an authorization code is valid. The default value is 1 minute.
PKCE field – PKCE (RFC 7636) is an extension to the Authorization Code flow to prevent certain attacks and to be able to securely perform the OAuth exchange from public clients. The following options are available:

- *Never require PKCE* – A client never has to provide a code challenge.
- *Always require PKCE, allow plain challenge* – A client always has to provide a code challenge which doesn't need to be hashed.
- *Always require PKCE, require hashed challenge* – A client always has to provide a code challenge which needs to be hashed.
- *Only require PKCE when no secret is provided, allow plain challenge* – A client must provide a code challenge when

not providing a secret. The code challenge doesn't need to be hashed.
- *Only require PKCE when no secret is provided, require hashed challenge*– A client must provide a code challenge when not providing a secret. The code challenge needs to be hashed.

SAML2 Identity Provider for SSO

Security Assertion Markup Language 2.0 (SAML 2.0) is an industry standard for exchanging identification, authentication and authorization data between trusted parties. Application Workspace implements a SAML 2.0 Identity Provider that can federate with SAML 2.0 Service Providers (SP) for exchanging data.

Settings screen

Metadata section

Allow metadata queries – If enabled, external parties can receive the metadata of this identity provider. In most cases, it's recommended to allow it, so that federations can be performed quickly and easily. This option can always be disabled after setting up a federation; it's important to remember that some service providers can monitor the metadata to automatically pick up changes like Active Directory Federation Service (AD FS).
Metadata URL – The URL where the metadata is published. Note that even if the **Allow metadata queries** option is disabled, an administrator can always download the metadata by using the **Download** button.

Single sign in/logout sections

Allow post requests (recommended) – If enabled, the identity provider allows incoming authentication/logout request messages by HTTP POST.
Allow redirect requests – If enabled, the identity provider allows incoming authentication/logout request messages by HTTP REDIRECT.
Require signing – If enabled, the authentication/logout messages must be signed by a trusted certificate. If disabled, any signing will be ignored.

> Recommendations
> We recommend you use the POST method over REDIRECT, especially when **Require signing** is enabled.
> Note that in the case of HTTP REDIRECT, the size of the messages can exceed the limit of the browser URL and also HTTP REDIRECT is less secure than HTTP POST.
> We also recommend activating the **Require signing** option for logout messages, otherwise it could be possible for a third-party to start the logout process for any known username.

Session section

Insert the number of seconds you want a SAML session to be valid for, without being refreshed. If you insert the value of zero, SAML sessions will never expire.

Service Providers screen

Service providers contains the definition and settings of the trusted third-parties that are federated with this identity provider.

Recast

If you click **Create service provider** in the table toolbar of the **Service Providers** tab, a dialog box with the same name will open.

Create service provider type

You can create a service provider manually. Find below the description of the most important elements of a *Create service provider* type of provider.

General tab

General section
Name – Provide the name of the service provider. It will be visible for users when they log out.
Entity ID – The SAML 2.0 Entity ID for this service provider.
Signing Certificate – The certificate used for verifying the SAML message signatures.
Profile – The profile that will be used to send additional information with a successful authentication response.

Name Identifier section
Format – The format of the unique identifier used for the authenticated user:

- *Transient* – an identifier that is only valid within a SAML session. If a user authenticates again through SAML, another newly created identifier will be used.
- *Persistent* – an identifier that is always valid for the same user. If a user authenticates again through SAML, the same identifier will be used.
- *Unspecified* – an identifier that is agreed upon between the two parties and that is not defined within the SAML specifications.
- *Windows Domain Qualified Name* – The authentication name as used for Active Directory.
- *Email Address* – The email address of the user.

Attribute – Specify the value of the name identifier.

End points tab

Single sign in/logout sections
Post URL – The service provider's URL to which SAML 2.0 POST authentication/logout messages should be sent.
Redirect URL – The service provider's URL to which SAML 2.0 REDIRECT authentication/logout messages should be sent.

Import Service provider from Metadata type

You can also create a service provider based on the metadata of that SP. The metadata can be provided by URL, file or plain XML text.

Edit service provider dialog box

To view the details of a service provider or edit it, double click its entry.

Access conditions tab

With access control you can limit which users can make use of this service provider. When the access is denied for a user, the user will be redirected to the requesting web application where an access denied error message will be displayed.

The filters available are the same as those for contexts.

Advanced tab

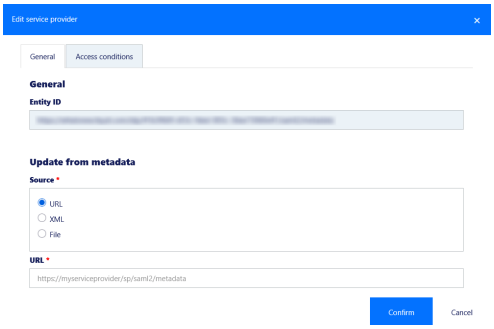
Advanced section

Use consumer URL if provided in an authentication request – If the service provider requests an authentication, it can also provide a consumer URL optionally. A consumer URL can be used to deep redirect the user into the service provider application. We recommend enabling this option when the signing of authentication messages is required, as this can leak potentially user data to untrusted third parties. No domain restriction is applied.
Do full logout of the workspace if this service provider request a logout – If enabled, when the service provider requests a logout, a full logout of the Application Workspace will execute. Otherwise, the logout will apply only for the service provider that requested it.
The service provider supports that the logout action is embedded (iframe) – Enable this option if the client supports being logged out within an HTML iframe. If there are problems with logging out, we recommend you disable this option.

Custom Logout section
Overrule logout action – If enabled, the logout action will ignore the SAML configuration and will use a custom logout page. This is especially handy for service providers that don't support SAML Single Logout.
Logout URL field – If the logout page supports redirect after logout, then use "${slo.return.url}" as variable to generate a URL back to the Application Workspace that will report the logout status to the logout page.
Does the custom logout report status back – If enabled, the logout page will wait for the status report. Otherwise, the logout page assumes a successful logout after a few seconds.
Does the custom logout use a secure connection – If enabled, the logout URL and redirects on the service provider are considered secure. A secure connection is important especially in scenarios where embedded content is supported by specific browsers.

Metadata tab

The **URL** and **XML** fields in this tab cannot be edited here, they just display the existing information. To import metadata from URL, XML or a local file you must first click **Update from metadata** and the following dialog will be displayed:



Here you can choose to enter the URL where the IdP SAML configuration is stored, copy the content of the metadata XML or import the metadata XML from a local folder on your computer.

After you click **Confirm** all the options that can be derived from the metadata will be updated.

Profiles screen

When a user authenticates to a service provider, the identity provider can send additional information about the user: group membership, company information etc.
Define a profile for every different set of information that should be sent.

Okta Connector

Recast

The Okta connector requires an Access Manager license.

This connector type allows you to setup a connection to Okta to import the applications defined there as ready to use applications within Application Workspace for distribution.

Prerequisites

- The domain name of the Okta tenant you wish to use.
- An application token from Okta to use for identification. To generate the Okta token, follow the instructions provided by the okta Developer guides.

If the connector is not used for some time, the token can be invalidated by Okta.

For more information about the **Overview**, **Entitlements**, **Synchronization Profile**, **Releases** and **Managed packages** screens, see Overview.

Settings screen

Okta Domain – The domain name of the Okta tenant, for example `liquit.okta.com`.
Token – The token that will be used to authenticate to Okta.

Office 365 groups support for Azure AD Identity Source

The Microsoft Entra ID (Azure AD) identity source allows you to register an application with Microsoft Entra ID (Azure AD) as a mean to authenticate against Application Workspace. This way you will leverage your Microsoft Entra ID (Azure AD) as the single point of entry. See SSO with Microsoft Entra ID (Azure AD) for configuration instructions.

You can connect Microsoft Entra ID (formerly Azure AD) to the following national clouds:

- Azure portal for US Government
- Azure portal China (operated by 21Vianet)
- Azure portal (global service).

For more details, see Microsoft Entra authentication & national clouds – Microsoft identity platform | Microsoft Learn.

Create identity source dialog box

The following options are available in this dialog box:

- Microsoft Entra ID (Azure AD) environment – to connect to Azure portal (global service)
- Custom environment – to connect to Azure portal for US Government or China (operated by 21Vianet)

Recast

Detailed view of the Microsoft Entra ID (Azure AD) identity source

See below the description of each screen in the detailed view of of the Microsoft Entra ID (Azure AD) identity source, and what actions you can perform in each of them.

Overview screen

Here you can configure a few basic options for the identity source.

Name – The name of the identity source. In the case of Microsoft Entra ID (Azure AD), we recommend you use the same value as the NetBIOS name of the it.
Type – The type of identity source.
Hidden – When an identity source is hidden, it will not appear on the login screen.

> Note that the **Name** and **Type** cannot be changed once the identity source is created.

Settings screen

It is required you register a new application in the Azure Portal before you can configure the settings for your Microsoft Entra ID (Azure AD) identity source. Below you find a list of its configurable settings.

Application section
Application ID – The value of the **Application (client) ID** field in Azure Portal > **Overview** page of the Microsoft Entra ID (Azure AD) app registration.
Client secret – The Microsoft Entra ID (Azure AD) app registration secret.
Use application ID as resource – When selected, the application ID will be used to request access to the Microsoft Entra ID (Azure AD). Otherwise, the default Azure AD Graph ID will be used.
Use redirect URI – The site to which the authorization server directs the user when the app has been successfully approved and an authorization code or access token has been issued. The redirection URL needs to be encoded to work properly.

OAuth 2 url's section
Fetch OAuth 2 url's – If you click this button, the system will prefill all the fields in this section and the**Microsoft Graph** section, based on a Microsoft Entra ID (Azure AD) tenant ID.
Authorization URI – The value of the **OAuth 2.0 authorization endpoint (v1)** field in Azure Portal > **Overview** page > **Endpoints** tab of the Microsoft Entra ID (Azure AD) app registration. For example: `https://login.microsoftonline.com/[Tennant ID]/oauth2/authorize`
Token URI – The value of the **OAuth 2.0 token endpoint (v1)** field in Azure Portal > **Overview** page > **Endpoints** tab of the Microsoft Entra ID (Azure AD) app registration. For example: `https://login.microsoftonline.com/[Tennant ID]/oauth2/token`
Logout URI – The logout URI provided by the Microsoft Entra ID (Azure AD) app registration. For example: `https://login.microsoftonline.com/[Tennant ID]/oauth2/logout?post_logout_redirect_uri=< redirection URL >` The redirection URI needs to be encoded to work properly.
JSON Web Key Set URI – The public key used to verify the signatures of JSON Web Tokens (JWTs).For more information, see Microsoft documentation.
Domain hint – You can provide Microsoft Entra ID (Azure AD) login page with a hint to which domain you want to authenticate. If the user has multiple active Microsoft Entra ID (Azure AD) sessions, and one session is matching the domain hint, then Microsoft Entra ID (Azure AD) will use that account and will not ask the user to select an account anymore. For example: `recastsoftware.com`

Microsoft Graph section
Graph endpoint – The Microsoft Graph API endpoint used to connect to the appropriate Azure environment—Global, US

Government, or China—based on your organization's cloud deployment. Once prefilled by clicking the '**Fetch OAuth 2 url's** button, we do not recommend changing the endpoint manually. For more information, see Microsoft documentation.

Synchronization section

Photos – Select if photos need to be synchronized or not. This option requires *User.Read.All* permissions in Microsoft Entra ID (Azure AD). See Register an application in Azure Portal, step 9 and SSO with Azure Active Directory for more information.
Use delta synchronization – When selected, delta synchronization of the Microsoft Entra ID (Azure AD) will be enabled. This causes an initial full synchronization to be performed, after which only changes are incrementally synchronized per Application Workspace server. This reduces the time it takes to fetch all users and groups from Microsoft Entra ID (Azure AD) after the initial synchronization is completed.
Include groups that are not security enabled – Enable support for groups that are not security enabled within Microsoft Entra ID (Azure AD). Like Microsoft 365 groups. This feature requires the Access Manager license.
Modifications – What kind of modifications are allowed for Microsoft Entra ID (Azure AD). Additional permissions are needed for modifying group membership. See SSO with Microsoft Entra ID (Azure AD) for more information.

Authentication screen

Here you can configure the methods available to authenticate.

Token exchange – Allow the token exchange to be used by third party integrators. For this option you need to insert the Application ID URI which is located in Azure Portal > Microsoft Entra ID (Azure AD) > App registration > Manage > Expose an API.
For more information, see How to setup your exchange token.
Federated – Allow authentication via federation. For example: Active Directory Federation Service (AD FS).
Form Authentication – Allow the user to login via the Application Workspace login page (http/https).
Basic Authentication – Allow basic authentication.

Contacts screen

Enable contacts – If enabled, contacts from this identity source will be used.
Require Email – If enabled, all objects without an email address will be hidden.
Group – Only show members of a certain group.

Show attributes

Choose which attributes to be synchronized to Application Workspace.

Authenticator screen

Assign an authenticator to the identity source.

Authenticator – You can select one of the existing authenticators defined in Application Workspace.
Prefix – Insert a string to add before the username to form the base distinguished name (DN).
Suffix – Insert a string to add after the username to form the base distinguished name (DN).