polarıs

Polaris Authentication Integration Guide

7.7

© 2025



Innovative Interfaces, Inc. 789 E. Eisenhower Parkway Ann Arbor, MI 48108 contract.admin@clarivate.com

Legal Notices

© Innovative (Part of Clarivate) and/or its affiliates. All rights reserved. All trademarks shown are the property of their respective owners.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

The software and related documentation are provided under an agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of the software, unless required by law for interoperability, is prohibited.

Contents

Introduction	. 1
Minimum Requirements	. 3
OIDC with Active Directory and AD FS Authentication	3
OIDC with Azure AD Authentication	3
Basic OAuth 2.0	3
Basic Authentication	. 4
Upgrading to Polaris 7.7	. 5
Using Multiple Identity Providers	7
Configuring OIDC with Active Directory and AD FS	8
Install Active Directory Federation Services	9
Configure Active Directory Federation Services	.18
Verify Active Directory Federation Services Is Running	.27
Verify that OAuth 2.0 is Enabled	.30
Create an Application Group	.32
Configure the AD FS Web Application: Claims and Permitted Scopes	.38
Enable CORS on AD FS To Accept Requests from Polaris APIs	.44
Set Up Web Services and Applications for OIDC with Active Directory and AD FS	.44
Customize the AD FS Pages	.58
Change the Access Token Lifetime	. 59
Bind a New SSL Certificate	.59
Troubleshoot	.60
Configuring OIDC with Azure AD	.62
Register LeapWebApp with Azure AD	.62
Create Client Credentials	66

Add Authentication Redirect URIs	67
Expose the Polaris.ApplicationServices API	70
Configure an ID Token	72
Set Up Users and Groups	74
Control Access to LeapWebApp Using Azure AD	76
Set Up Web Services and Applications for OIDC with Azure AD	
Configuring Basic OAuth 2.0	
Set Up Web Services and Applications for Basic OAuth 2.0	94
Add Authentication Redirect URIs	
Configuring Basic Authentication	
Set Up Web Services and Applications for Basic Authentication	
Add a URL Rewrite Rule for LeapWebApp	
Sample Rewrite Rule Text	107
Additional URL Rewrite Resources	

Introduction

Both Leap and Polaris System Administration (web-based) support the following types of authentication:

- Open ID Connect (OIDC) OIDC is the identity layer that sits on top of OAuth 2.0. You can use Proof Key for Code Exchange (PKCE) for security with this configuration.
- Basic OAuth 2.0 This configuration doesn't use OIDC or PKCE. In addition, basic OAuth 2.0 doesn't rely on claims found within a JSON web token (JWT). Instead, it exchanges an access token (typically, an opaque token) to retrieve claims from the OAuth /userinfo endpoint.
- Basic authentication This configuration uses the Windows operating system's local security and/or its direct integration with Active Directory.

For detailed information about the types of authentication that are supported for each application and software version, see <u>Minimum Requirements</u>.

Staff authentication for each application is handled by an identity provider. We provide detailed instructions for configuring OIDC with Active Directory and Active Directory Federation Services (AD FS) or with Azure Active Directory (Azure AD). You can also use a different identity provider.

Note:

Azure AD has recently been rebranded as "Microsoft Entra ID". In this guide, references to Azure AD are interchangeable with Microsoft Entra ID.

To use this guide, review the <u>Minimum Requirements</u> for the configurations you plan to use:

- OIDC configurations:
 - OIDC with Active Directory and AD FS Authentication
 - OIDC with Azure AZ Authentication
- Basic OAuth 2.0
- Basic Authentication

Then, continue to one of the following configuration procedures:

- <u>Configuring OIDC with Active Directory and AD FS</u>
- Configuring OIDC with Azure AD
- Configuring Basic OAuth 2.0
- Configuring Basic Authentication

Minimum Requirements

This section discusses minimum requirements for the following types of configurations:

- OIDC configurations:
 - OIDC with Active Directory and AD FS Authentication
 - OIDC with Azure AZ Authentication
- Basic OAuth 2.0
- Basic Authentication

OIDC with Active Directory and AD FS Authentication

To use OIDC OAuth 2.0 with Azure AD authentication, you must have:

- Windows Server 2019 Standard
 - Polaris requires OAuth 2.0 with PKCE support
 - AD FS on Windows Server 2019 supports PKCE
- Active Directory Domain Services
- SSL Certificate
 - Publicly trusted CA signed certificate
- The Polaris 7.2 or later LeapWebApp installed
- The Polaris 7.1 or later PolarisAdmin installed

OIDC with Azure AD Authentication

To use OIDC OAuth 2.0 with Azure AD authentication, you must have:

- Access to Microsoft's Azure AD services
- The Polaris 7.3 or later LeapWebApp installed
- The Polaris 7.3 or later PolarisAdmin installed

Basic OAuth 2.0

To use basic OAuth 2.0, you must have:

- An identity provider configured for basic OAuth 2.0
- SSL Certificate
 - Publicly trusted CA signed certificate
- The Polaris 7.5 or later LeapWebApp installed
- The Polaris 7.6 or later PolarisAdmin installed

Basic Authentication

To use basic authentication, you must have the Polaris 7.7 or later PolarisAdmin installed.

Upgrading to Polaris 7.7

In previous versions, you had to configure authentication separately for Leap and Polaris System Administration (web-based). Leap supported basic authentication, OIDC, and basic OAuth 2.0. Polaris System Administration (web-based) only supported OIDC and basic OAuth 2.0.

Version 7.7 includes a new centralized authentication service that's designed to be used with both Leap and Polaris System Administration (web-based).

This means that if your library currently uses basic authentication for Leap, you can now begin using Polaris System Administration (web-based) with basic authentication. For more information, see <u>Configuring Basic Authentication</u>.

If your library currently uses OIDC or basic OAuth 2.0, you can continue using it for authentication for both Leap and Polaris System Administration (web-based). However, when you upgrade to version 7.7, you must edit your configuration so that it uses the new centralized authentication service. To do this, you must configure a .json file for each of the following:

- Polaris.Authentication (the application that authenticates Polaris users)
- Polaris.AuthenticationServices (the API service that provides backend support for authentication)

For more information, see the appropriate instructions for your configuration:

- Set Up Web Services and Applications for OIDC with Active Directory and AD FS
- Set Up Web Services and Applications for OIDC with Azure AD
- Set Up Web Services and Applications for Basic OAuth 2.0

Important:

If your library currently uses single sign-on (SSO) authentication with Leap, you must update the allowed redirect URLs that are specified in your identity provider configuration.

Regardless of your configuration or identity provider, in version 7.7 you must also configure the following two lines in the appsettings.user.json file for LeapWebApp:

```
"Polaris.AuthAPI.URL": "https://[server-
address]/Polaris.AuthenticationServices/",
"Polaris.AuthApp.URL": "https://[server-address]/polarisauth/",
```

Replace [server-address] with the server address for your identity provider.

Note:

If you are upgrading to version 7.7 from a previous version, find the version 7.7 appsettings.user.json template file for LeapWebApp, copy these two lines, and paste them into your appsettings.user.json file.

You must perform additional configuration if one or both of the following is true for your library:

- You want to enable permission overrides in Leap.
- You want to enable reauthentication for Leap.

Note:

Permission overrides and reauthentication are not supported if your system uses multiple identity providers for authentication.

If one or both of the above are true, refer to the instructions for your configuration in the 7.5 version of the <u>Polaris and OAuth 2.0 with OpenID Connect Integration Guide</u>:

- In the Configuring Active Directory with AD FS section, see the Set Up LeapWebApp instructions.
- In the Configuring Azure AD section, see the Configure LeapWebApp for Use with Azure AD instructions.

Using Multiple Identity Providers

As of version 7.7, if your system uses OIDC or basic OAuth 2.0, organizations in your system can use different identity providers to sign in to a single instance of Leap or Polaris Administration (web-based).

For example, a consortium system with three member libraries could use the following authentication configurations:

- Library A OAuth 2.0 with Active Directory and Active Directory Federation Services (AD FS)
- Library B OAuth 2.0 with Azure Active Directory
- Library C Basic OAuth 2.0 with Akamai

Configuring OIDC with Active Directory and AD FS

Important:

The mechanism used to connect an Active Directory user to a Polaris user is the user principal name (UPN) in the format of an email address. For example, user@mydomain.com. During the account verification process, we request the UPN claim from Active Directory. This must return a UPN in the name@domain format. The Polaris.AdminServices (API) can then use that information to map the AD user to a Polaris user.

To configure OIDC with Active Directory and AD FS, perform the following tasks:

- 1. Install Active Directory Federation Services.
- 2. <u>Configure Active Directory Federation Services</u>.
- 3. <u>Verify that Active Directory Federation Services is running</u>.
- 4. Verify that OAuth 2.0 is Enabled.
- 5. <u>Create an Application Group for Polaris LeapWebApp</u>.
- 6. <u>Configure the AD FS Web Application: Claims and Permitted Scopes.</u>
- 7. Enable CORS on AD FS to accept requests from Polaris APIs.
- 8. Set up web services and applications.
- 9. <u>Customize the AD FS pages</u>.
- 10. Change the access token lifetime.
- 11. Bind a new SSL certificate.
- 12. <u>Troubleshoot</u>.

After you complete these tasks, Add a URL rewrite rule for LeapWebApp.

Install Active Directory Federation Services

To install AD FS

- 1. Sign in to Windows Server 2019 with administrative privileges.
- 2. Start the Server Manager desktop application.

🔁 Server Manager		- 🗆 X		
Server M	lanager • Dashboard • 🕄 🚩 Manage Iools	<u>V</u> iew <u>H</u> elp		
Image: Dashboard WELCOME TO SERVER MANAGER Image: Local Server Image: Dashboard Image: DNS Image: Dashboard Image: DNS<				
	Connect this server to cloud services LEARN MORE ROLES AND SERVER GROUPS Roles: 3 Server groups: 1 Servers total: 1	Hide		
	Image AD DS 1 Image AD DS 1 <td< th=""><th>1</th></td<>	1		

3. On the Server Manager Dashboard view, select Add roles and features.

The Add Roles and Features Wizard opens.

🔁 Server Manager					— C) ×
Add Roles and Features Wizard		_		anage <u>T</u> ools	<u>V</u> iew	<u>H</u> elp
Before you begin		DEST PolarisOauth.pi	INATION SERVER pl-qa-oauth.local			^
Le Before You Begin Installation Type Server Selection Server Roles Features Confirmation Results	This wizard helps you install roles, role ser features to install based on the computing hosting a website. To remove roles, role services, or features Start the Remove Roles and Features Wizz Before you continue, verify that the follow • The Administrator account has a strong • Network settings, such as static IP addre • The most current security updates from If you must verify that any of the precedir complete the steps, and then run the wizz To continue, click Next.	rvices, or features. You determine which roles, r g needs of your organization, such as sharing d ard wing tasks have been completed: password esses, are configured Windows Update are installed ng prerequisites have been completed, close the ard again.	ole services, or ocuments, or e wizard,		Hid	le
	Services Performance BPA results	Previous Install Services Performance BPA results	Cancel Servic Perfo BPA r	es es eability res rmance esults	1	

4. On the **Before You Begin** tab, select **Next**.



5. On the **Installation Type** tab, select **Role-based or feature-based installation**, and then select **Next**.

🚡 Server Manager				– 🗆 X
Server Mar	nager • Dashboard		• @ M	lanage Tools View Help
Image: Dasht Select destinat Image: Local Before You Begin Image: AD PS Before You Begin Image: AD PS Server Selection Image: DNS Server Roles Image: File ar Features Confirmation Results	Select a server or a virtual hard disk Select a server from the server Select a virtual hard disk Server Pool Filter: Name IP Add PolarisOauth.pol-qa-cau 1025. 1 Computer(s) found This page shows servers that are rur and that have been added by using newly-added servers from which da	Pointsc on which to install roles and features. ool dress Operating System 197.8 Microsoft Windows Server 2019 S hning Windows Server 2012 or a newer release the Add Servers command in Server Manager, ta collection is still incomplete are not shown.	DESTINATION SERVER auch.pol-qa-auchlocal	Hide
	Performance BPA results		all Cancel Performance BPA results	
	File and Storage	Local Server 1	All Servers	1

6. On the Server Selection tab, select the server, and then select Next.

📥 Server Ma	nager				- 🗆 ×
$\textcircled{\Rightarrow}$	Server Manage	er • Dashboard		• 🗊 🚩	<u>M</u> anage <u>T</u> ools <u>V</u> iew <u>H</u> elp X
Dasht Local Local All See All See AD DS DNS File an	Select server roles Before You Begin Installation Type Server Selection Server Roles Features	Select one or more roles to install o Roles Active Directory Certificate Active Directory Tederation Active Directory Federation Active Directory Information	n the selected server. Services Services Services Directory Services	DESTINATION SERVI PolarisOauth.pol-qa-oauth.loc Description Active Directory Federation Service (AD FS) provides simplified, secure identity federation and Web single sign-on (SSO) capabilities. AD FS	sr al
	AD FS Confirmation Results	Active Directory Glights Mar Device Health Attestation DHCP Server DNS Server (Installed) Fax Server Tots Server (Installed) Fax Server Hyper-V Network Policy and Access Print and Document Service Remote Access Volume Activation Services Volume Activation Services Web Server (IIS) Windows Deployment Servi Windows Server Update Service	of 12 installed) Services s ces vices vice	includes a Federation Service that enables browser-based Web SSO.	Hide torage 1 lity
			< <u>P</u> revious <u>N</u> ext :	> Install Cancel	
		Performance BPA results	Performance BPA results	Perfc BPA	rmance results

- 7. On the **Server Roles** tab, do the following:
 - a. Verify that Active Directory Domain Services are installed.
 - b. Select the Active Directory Federation Services role.
 - c. Select Next.

📥 Server Ma	nager						- [) ×
$\textcircled{\Rightarrow}$	Server Manag	er • Dashboard		• ©	Manag	ge <u>T</u> ools	<u>V</u> iew	<u>H</u> elp
Local : Local : All Set AD DS DNS File an	Select features Before You Begin Installation Type Server Selection Server Roles Features AD FS Confirmation Results	Select one or more features to instal Features Network Virtualization Peer Name Resolution Proto Quality Windows Audio Vide RAS Connection Manager AA Remote Differential Compre- P Remote Server Administratic RPC over HTTP Proxy Setup and Boot Event Collec Simple TCP/IP Services SMB Bandwidth Limit SMTP Service Storage Migration Service Pr Storage Replica System Data Archiver (Install <	I on the selected server.	DESTINATI PolarisOauth.pol-qa Description Provides services to collec archive Windows Server sy	ION SERVER -oauth.local t and /stem data.	torage	Hid	e
			< Previous Next :	Install	Cancel			
		Performance BPA results	Performance BPA results		Performan BPA results	ce		

8. On the **Features** tab, select **Next**.

🚡 Server Manager				_	
Server Mana	iger • Dashboard		• 🕝 🚩 Mana — 🗆 X	ige <u>T</u> ools <u>V</u> iew	/ <u>H</u> elp
Image: Dasht Active Directory All Set Before You Begin All Set Installation Type DNS Installation Type File ar Server Roles Peatures AD FS Confirmation Results	 Federation Services (/ Active Directory Federation Services authenticate a user to multiple Wet bypass the need for secondary accorrights to trusted partners. In this fed own identities. Things to note: This computer must be joined to The Web Application Proxy role s service proxy and cannot be instation. The Web Application Proxy role s service proxy and cannot be instation. Azure Active Directon access management, apps. Learn more about Az Configure Office 365 	AD FS) s (AD FS) provides Web single-sign-on (S applications using a single user account bounts by allowing you to project a user's of derated environment, each organization of a domain before you can successfully ins revice in the Remote Access server role f alled on the same computer as the federative ry, a separate online service, can provide security reporting, single sign-on to clou- ture Active Directory with Azure Active Directory Connect	DESTINATION SERVER PolarisOauth.pol-qa-oauth.local SO) capabilities to t. AD FS helps organizations digital identity and access continues to manage its stall the Federation Service. functions as the federation ation service.	torage 1	Hide
		< <u>P</u> revious <u>N</u> ext >	Install Cancel		
	Performance BPA results	Performance BPA results	Performar BPA result	nce ts	

9. On the **AD FS** tab, read the Active Directory Federation Services (AD FS) information, and then select **Next**.

📥 Server Manager		-
Server	Manager 🕨 Dashboard	• ② <mark> </mark> Manage <u>T</u> ools <u>V</u> iew <u>H</u> elp − □ ×
Local: Local: All Set All Set AD DS DNS File ar File ar Confirm in Before You Beg Installation Typ Server Roles Features AD FS Confirmation Results	stallation selections in To install the following roles, role services, or features on select a Restart the destination server automatically if required b Optional features (such as administration tools) might be displuseen selected automatically. If you do not want to install these their check boxes. Active Directory Federation Services Export configuration settings Specify an alternate source path	DESTINATION SERVER PolarisOauthpol-ga-oauthJocal ted server, click Install. layed on this page because they have e optional features, click Previous to clear Hide Hide
	< <u>Previous</u> <u>Next</u>	t > Install Cancel
	Performance Performance BPA results BPA results	e Performance BPA results

10. On the **Confirmation** tab, confirm your selections, and then select **Install**.



11. On the **Results** tab, select **Close** when the installation is complete.



- 12. On the Server Manager dashboard, verify that AD FS is an installed role.
- 13. Restart the server.

Configure Active Directory Federation Services

To configure Active Directory Federation Services

1. Start the Server Manager desktop application.

The system generates a configuration notification.



2. Open the notification, and select **Configure the federation service on this server**.

The Active Directory Federation Services Configuration wizard opens.



3. On the Welcome tab, select Next.

Active Directory Federation Serv	ices Configuration Wizard			-		×
Connect to Active	Directory Domain Servic	es	PolarisOaut	TAR h.pol-qa	GET SER -oauth.l	VER ocal
Welcome Connect to AD DS Specify Service Properties Specify Service Account Specify Database Review Options Pre-requisite Checks Installation Results	Specify an account with Active Directory dom federation service configuration. iii\jeff.young	ain administrator	permissions to	perform	the	
	< Previou	s Next >	Config	ure	Cancel	

4. Select **Change**, and provide an administrator account. Then select **Next**.

ctive Directory Federation Sen	vices Configuration Wizard		-		
pecify Service Pr	roperties	PolarisOaut	TA h.pol-q	RGET SEI Ja-oauth	R\ .lo
Welcome	SSL Certificate:	*.polarislibrary.com ~		mport	
Specify Service Properties		View			
Specify Service Account	Federation Service Name:	dev-fs.polarislibrary.com ~			
Specify Database		Example: fs.contoso.com			
Review Options	Enders Constant Director Numer				
Pre-requisite Checks	Federation Service Display Name:	Polaris K&D Federation Service			
Installation		Users will see the display name at sigr Example: Contoso Corporation	n in.		
	<	Previous Next > Config	ure	Cance	el

5. If not already installed on the server, select **Import** to install an SSL certificate.

📥 Active Directory Federation Serv	ices Configuration Wizard		-		×
Specify Service Pr	operties	PolarisOauth	TAR pol-qa.	GET SER -oauth.	VER ocal
Welcome Connect to AD DS Specify Service Properties	SSL Certificate:	*.polarislibrary.com ~	Im	iport	
Specify Service Account Specify Database	Federation Service Name:	dev-fs.polarislibrary.com v Example: fs.contoso.com			
Pre-requisite Checks Installation Results	Federation Service Display Name:	Polaris R&D Federation Service Users will see the display name at sign Example: Contoso Corporation	in.		
	<	Previous Next > Configu	re	Cance	

- 6. Enter the following, and then select **Next**:
 - Federation Service Name
 - Federation Service Display Name

📥 Active Directory Federation Serv	rices Configuration Wizard			- [) X
Specify Service A	ccount		PolarisOaut	TARGE h.pol-qa-oa	T SERVER auth.local
▲ Group Managed Service Ad Welcome Connect to AD DS Specify Service Properties Specify Service Account Specify Database Review Options Pre-requisite Checks Installation Results	counts are not available beca Specify a domain user acc Create a Group Manag Account Name: Use an existing domai Account Name: Account Password:	ount or group Managed Service ged Service Account POL-QA-OAUTH n user account or group Manag POL-QA-OAUTH ••••••	e Account.	Show mor	e X
		< Previous Nex	t > Configu	ıre	Cancel

7. Specify a service account, and then select **Next**.

Active Directory Federation Serv	ices Configuration Wizard		_		\times
Active Directory Federation Servi Specify Configuration Welcome Connect to AD DS Specify Service Properties Specify Service Account Specify Database Review Options Pre-requisite Checks Installation Results	ices Configuration Wizard tion Database Specify a database to store the Active Di	F rectory Federation Service o g Windows Internal Databa database. <i>To use the default instance,</i>	TA PolarisOauth.pol-o configuration data ise.	RGET SEF ga-oauth.	X RVER local
	< Pre	evious Next >	Configure	Cance	21

8. Specify the location of the AD FS configuration database, and then select **Next**.

For simple scenarios, creating the local database is acceptable.

Active Directory Federation Servi	ces Configuration Wizard	-	-		×
Review Options		PolarisOauth.p	TARG ool-qa-	GET SER	/ER ocal
Welcome Connect to AD DS Specify Service Properties Specify Service Account Specify Database Review Options Pre-requisite Checks Installation Results	Review your selections: This server will be configured as the primary server in a new AD FS fs.polarislibrary.com'. AD FS configuration will be stored in Windows Internal Database. Windows Internal Database feature will be installed on this server Federation service will be configured to run as POL-QA-OAUTH\Pa	S farm 'dev- if it is not alread	dy insta	alled.	
	These settings can be exported to a Windows PowerShell script to additional installations Previous Next >	automate Configure	View	w script Cancel	

9. Review your selections, and then select **Next**.

Active Directory Federation Serv	ices Configuration Wizard	_		×
Pre-requisite Che	CKS PolarisOau	TAF th.pol-qa	₹GET SEF a-oauth.	RVER local
All prerequisite checks pass	ed successfully. Click 'Configure' to begin installation.	Show r	more	×
Welcome Connect to AD DS	Prerequisites must be validated before Active Directory Federation Services is computer.	configure	ed on th	is
Specify Service Properties	Rerun prerequisites check			
Specify Service Account	0			
Specify Database	Niew results			
Review Options	Prerequisites Check Completed			
Pre-requisite Checks	All prerequisite checks passed successfully. Click 'Configure' to begin in	stallatior	1.	
Installation				
	< Previous Next > Config	Jure	Cance	2

10. After you complete all pre-requisite checks, select **Configure**.

	PolarisOauth.pol-qa-oauth.lo
This server was successfully Welcome	Configured Show more X New detailed operation results
Connect to AD DS	A machine restart is required to complete ADFS service configuration. For more information, see: https://go.microsoft.com/fwlink/?Linkld=798725
Specify Service Properties Specify Service Account	The SSL certificate subject alternative names do not support host name 'certauth.dev- fs.polarislibrary.com'. Configuring certificate authentication binding on port '49443' and hostname 'dev-fs.polarislibrary.com'
Specify Database Review Options	 Failed to register SSL bindings for Device Registration Service: An item with the same key has already been added.
Pre-requisite Checks	
Results	
	Next steps required for completing your federation service deployment

11. When the configuration wizard has completed successfully, select **Close**, and then restart the server.

Verify Active Directory Federation Services Is Running

To verify that Active Directory Federation Services is running

1. Start the Server Manager desktop application.

AD FS should be green.

📥 Server Manager					
Server Ma	anager 🕨 Dasł	board			
 Dashboard Local Server All Servers AD DS AD FS DNS File and Storage Services ▷ 	LEARN MORE ROLES AND SERVE Roles: 4 Server grou AD DS AD DS Manageabil Events Services Performance BPA results	5 Col R GROUPS ps: 1 Servers total: 1 ty	nnect t	AD FS Manageability Events Services Performance BPA results	oud services

2. Start the Services application and check the status.

🔍 Services					_	Х
File Action View	Help					
🔶 🔿 📘 🖬 🖬	à 🔒 🛛 🖬 🕨 🔲 II 🕩					
🔍 Services (Local)	Name	Description	Status	Startup Type	Log On As	^
	Active Directory Domain Services	AD DS Dom	Running	Automatic	Local Syste	
	Active Directory Federation Services	Enables Acti	Running	Automatic (Delayed	POL-QA-O	
	Active Directory Web Services	This service	Running	Automatic	Local Syste	
	ActiveX Installer (AxInstSV)	Provides Us		Disabled	Local Syste	
	🔍 AllJoyn Router Service	Routes AllJo		Manual (Trigger Start)	Local Service	

- 3. Open the Edge (or Chrome) web browser and go to https://localhost/adfs/fs/federationserverservice.asmx
 - If you want to ignore certificate errors, select Advanced.

A page similar to the following image opens:

thtps://localhost/adfs/fs/federati x +			-		×
\leftarrow \rightarrow $ m C$ $ m A$ Not secure https://localhost/adfs/fs/federationserverservice.asmx	ŵ	ל≡	Ē		
This XML file does not appear to have any style information associated with it. The document tree is shown below.					
<pre>v<usdl:definitions "="" 07="" 2005="" <br="" activedirectory="" federationservice="" http:="" schemas.microsoft.com="" xmlns:ws10="http://www.w3.org/2005/08/addressing" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:wsu="http://docs.oasis:
200401-ws-wssecurity-utility-1.0.xsd" xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/04
xmlns:u6=">xmlns:wsp="http://schemas.microsoft.com/ws/2005/12/wsdl/contract" xmlns:ws2="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:ws="http://schemas.microsoft.com/ws/2005/12/wsdl/contract" xmlns:wsap="http://schemas.xmlsoap.org/wsdl/ xmlns:ws="http://schemas.misoap.org/ws/2004/08/addressing" xmlns:wsap="http://schemas.xmlsoap.org/wsdl/ xmlns:wsa="http://schemas.xmlsoap.org/wsdl/contract" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/ xmlns:wsa="http://www.w3.org/2001/XHLSchema" xmlns:tns="http://tempuri.org/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/ xmlns:wsa="http://www.w3.org/2001/XHLSchema" xmlns:tns="http://tempuri.org/" xmlns:soap="http://schemas.xmlsoap.org/soap/encoding/" name="ADFSITrustInformationService" targetNamespace="http://tempuri.org/" xmlns:soap="nttp://schemas.xmlsoap.org/soap/encoding/" v<wsp:policy wsu:id="BasicHttpBinding_ITrustInformationContract_policy"> v<wsp:policy v<wsp:policy> v<wsp:policy> v<wsp:policy> v<wsp:policy> v<sp:transportbinding xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy"> v<wsp:policy> v<sp:transportbinding xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy"> v<wsp:policy> v<sp:httpstoken requireclientcertificate="false"></sp:httpstoken></wsp:policy></sp:transportbinding></wsp:policy></sp:transportbinding></wsp:policy></wsp:policy></wsp:policy></wsp:policy></wsp:policy </wsp:policy></usdl:definitions></pre>	//mex" open.o sssing/ soap12 tadata org/ws	rg/wss/ policy" /" dl/soap	2004/01 /"	/oasis	-

This indicates that Active Directory Federation Services is running.

Verify that OAuth 2.0 is Enabled

To verify that OAuth 2.0 is enabled

1. Open the AD FS Management desktop application.



2. Open the Service folder, and then select the Endpoint folder.

🗌 AD FS 🍿 File Action View Window Help					×
🗢 🔿 🙍 🖬 🛛 🖬					
AD FS Er	ndpoints			A	ctions
 Service Attribute Stores Certificates Claim Descriptions Device Registration Endpoints Scope Descriptions Web Application Proxy Access Control Policies Relying Party Trusts Claims Provider Trusts Application Groups N 	Imabled Proxy Enabled inabled No io No	URL Path /adfs/services/trust/13/usemamebasictransport /adfs/services/trust/13/usemamemixed /adfs/services/trust/13/usuedtokenasymmetricbasic:256 /adfs/services/trust/13/usuedtokenasymmetricbasic:256sh /adfs/services/trust/13/usuedtokenmixedasymmetricbasic /adfs/services/trust/13/usuedtokenmixedasymmetricbasic /adfs/services/trust/13/usuedtokenmixedasymmetricbasic /adfs/services/trust/13/usuedtokenmixedasymmetricbasic /adfs/services/trust/13/usuedtokenmixedasymmetricbasic /adfs/services/trust/13/usuedtokensymmetricbasic.256 /adfs/services/trust/13/usuedtokensymmetricbasic256 /adfs/services/trust/13/usuedtokensymmetrictpiedes /adfs/services/trust/13/usuedtokensymmetrictpiedessha /adfs/services/trust/13/usuedtokensymmetrictpiedes /adfs/services/trust/13/usuedtokensymmetrictpiedes /adfs/services/trust/13/usuedtokensymmetrictpiedes /adfs/services/trust/13/usuedtokensymmetrictpiedes /adfs/services/trust/13/usuedtokensymmetrictpiedes /adfs/services/trust/13/usuedtokensymmetrictpiedes /adfs/services/trust/13/usidows /adfs/services/trust/13/windows /adfs/services/trust/atfactresolution /adfs/services/trust/atfactresolution /adfs/services/trust/atfactresolution /adfs/services/trust/atfactresolution	Type Type WS-Trust 1.3 WS-Trust 1.3 WS-Trust 1.4 WS-Trust 1.4 WS-Trust 1.4 WS-Trust 1.4 <t< th=""><th></th><th>ndpoints View View New Window from Here Refresh Help adfs/oauth2/ Disable on Proxy Disable Help</th></t<>		ndpoints View View New Window from Here Refresh Help adfs/oauth2/ Disable on Proxy Disable Help
<	Metadata		>	~	

- 3. Search for the oauth2 path.
- 4. In either the Edge or Chrome web browser, go to https://localhost/adfs/.wellknown/openid-configuration
 - If you want to ignore certificate errors, select Advanced.

A page similar to the following image opens:



This indicates that OAuth 2.0 is available.

Create an Application Group

To create an application group for use with Polaris Admin and LeapWebApp

1. Open the AD FS Management desktop application.



- 2. Select the Application Groups folder.
- 3. Under Actions, select Add Application Group.

The Add Application Group wizard opens.
翰 Add Application Group Wizard		
Welcome		
Steps	Name:	
Welcome	Polaris	
 Native application Apply Access Control Policy 	Description:	_
Summary Complete	Polaris web applications	
 Complete 	Template:	
	Client-Server applications Image: Native application accessing a web API	
	Server application accessing a web API	
	Standalone applications	
	Native application	
	Server application	
	Web API	
	More information	
	< Previous Next > Cancel	

- 4. On the **Welcome** tab, do the following:
 - a. In the Name box, enter Polaris.
 - b. In the **Description** box, enter **Polaris web applications**.
 - c. From the Template section, select **Web browser accessing a web application**.
- 5. Select Next.

翰 Add Application Group W	izard	×
Native application		
Steps	Name:	
Welcome	Polaris - Native application	
Native application	Client Identifier:	
Apply Access Control Policy	0a586b1e-eeb0-4c8a-8381-50e9cafec240	
Summary		
Complete	Redirect URI:	
	Example: https://Contoso.com	Add
	https://rd-polaris.polarislibrary.com/PolarisAdmin/login	Remove
	https://rd-polaris.polarislibrary.com/Polaris.AdminServices/swagger/oauth2-redirect.html	
	< >	
	Description:	
	< Previous Next >	Cancel

6. On the Native application tab, in the Redirect URI box, enter the following URIs:

- https://server_address/polarisauth/login
- https://server_address/polarisauth/logout
- https://server_address/polarisauth/signin-oidc
- https://server_address/polarisauth/signout-callback-oidc
- https://server address/leapwebapp/signin-oidc
- https://server address/leapwebapp/signin-override-oidc
- https://server address/leapwebapp/signout-callback-oidc

 https://server address/Polaris.ApplicationServices/swagger/oauth2redirect.html

Note: Replace *server address* with the FQDN that matches your Polaris System Administration (web-based) or Leap server name and certificate.

7. Copy the value in the **Client Identifier** box to Notepad.

You'll need this when you set up PolarisAdmin's appsettings.user.json.

8. Select Next.

🏟 Add Application Group Wi	🙀 Add Application Group Wizard		
Choose Access Contro	l Policy		
Steps	Choose an access control policy:		
Welcome Native application Apply Access Control Policy Summary Complete	Choose an access control policy: Name Pemit everyone Pemit everyone and require MFA Pemit everyone and require MFA for specific group Pemit everyone and require MFA from extranet access Pemit everyone and require MFA from unauthenticated Pemit everyone and require MFA, allow automatic devi Pemit everyone for intranet access Pemit everyone for intranet access Pemit everyone Policy Pemit everyone I do not want to configure the access control policy at thi application.	Description Grant access to everyone and require MFA f Grant access to the intranet users. Grant access to users of one or more specifi	3
		< Previous Next > Cancel	

9. On the **Apply Access Control Policy** tab, select an access control policy, and then select **Next**.

翰 Add Application Group W	izard	×
Summary		
Steps	Review the following settings and click 'Next' to create the application.	
 Welcome Native application Apply Access Control Policy Summary Complete 	Application Group Name: Polaris Description: Polaris web applications Native application Name: Polaris - Native application Identifier: 0a586b 1e-eeb0-4c8a-8381-50e9cafec240 Redirect URIs: https://rd-polaris.polarislibrary.com/PolarisAdmin/ https://rd-polaris.polarislibrary.com/PolarisAdmin/login https://rd-polaris.polarislibrary.com/PolarisAdmin/servicess https://rd-polaris.polarislibrary.com/Polaris.AdminServices/swagger/oauth2redirect.html Web application Identifiers: 0a586b 1e-eeb0-4c8a-8381-50e9cafec240 Access control policy: Permit everyone Application permissions:	> ×
	< Previous Next > Can	cel

10. On the **Summary** tab, review the settings and then select **Next**.



11. On the **Complete** tab, select **Close**.

Configure the AD FS Web Application: Claims and Permitted Scopes

To configure the AD FS web application

1. Open the AD FS Management desktop application.



- 2. Select the **Application Groups** folder.
- 3. Select the **Polaris** application group, and then select **Properties**.

olaris Properties		
eneral		
Name:		
Polaris		
Description:		
Polaris web applications		
Applications:		
Name	Description	
Native application		
Polaris - Native application		
Web application		
Polaris - Web application		
Add application	Edit Remove	
	OK Cancel Apply	

4. Select Polaris - Web application, and then select Edit.



5. Select the Issuance Transform Rules tab, and then select Add Rule.

🗌 Add Transform Claim Rule Wizard		
Select Rule Template		
Steps	Select the template for the claim rule that you want to create from the following list. The description provides	8
Choose Rule Type	details about each claim rule template.	
Configure Claim Rule	Claim rule template:	
	Send Claims Using a Custom Rule $\qquad \lor$	
	Claim rule template description:	
	Using a custom rule, you can create rules that can't be created with a rule template. Custom rules are written in the AD FS claim rule language. Capabilities that require custom rules include: • Sending claims from a SQL attribute store • Sending claims from a custom attribute store • Sending claims only when 2 or more incoming claims are present • Sending claims only when an incoming claim value matches a complex pattem • Sending claims with complex changes to an incoming claim value • Creating claims for use only in later rules	
	< Previous Next > Cancel	

6. On the Add Transform Claim Rule Wizard, select **Send Claims Using a Custom Rule** from the **Claim rule template** list, and then select **Next**.

🏟 Add Transform Claim Ru	Jle Wizard	×
Configure Rule		
Steps Choose Rule Type Configure Claim Rule	You can configure a custom claim rule, such as a rule that requires multiple incoming claims or that extracts claims from a SQL attribute store. To configure a custom rule, type one or more optional conditions and an issuance statement using the AD FS claim rule language. Claim rule name: Forward UPN Claim Rule template: Send Claims Using a Custom Rule Custom rule: c:[Type == "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"] => issue(claim = c);]	~
	< Previous Finish Cancel	

- 7. In the Claim rule name box, enter Forward UPN Claim.
- 8. In the **Custom rule** box, enter the following rule:

```
c:[Type ==
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"]
=> issue(claim = c);
```

- 9. Select Finish.
- 10. On the Issuance Transform Rules tab, select Add Rule.
- 11. On the Add Transform Claim Rule Wizard, select **Send Claims Using a Custom Rule** from the **Claim rule template** list, and then select **Next**.
- 12. In the Claim rule name box, enter Add Tenantld.
- 13. In the **Custom rule** box, enter the following rule:

```
=> issue(Type =
"http://schemas.microsoft.com/identity/claims/tenantid",
Value = "polaris");
```

14. Select Finish.

Polaris - Web applicat	on Properties			×
Identifiers Notes Acc	cess control policy Iss	uance Transform Rule	es Client	Permissions
Configure application pe <u>Client application (cal</u>	rmissions to enable clie ler):	ent applications to ace	ss this Web	o API
Name		Description		
Polaris - Native app	lication			
		A	dd	Remove
Permitted scopes:		_		_
Cases Name	Deserietien			
	Description			
	Requests the access	token claims in the id	entity token). Jean
	Bequest the email cla	im for the signed in us	ny refresh to	ken.
	The logon cert score	allows an application	nor. Into request	
openid	Request use of the O	penID Connect autho	rization pro	tocol
	Request profile related	d claims for the signed	l in user.	
user_imperso	Request permission for	or the application to a	ccess the re	esour 🧹
	_			
			1	New scope
		OK	Cancel	Apply

- 15. On the **Client Permissions** tab, verify that **email** and **openid** are selected.
- 16. Select **OK** to close the Web application Properties dialog.
- 17. Select **OK** to close the Polaris properties dialog.
- 18. Using the services applet, restart the Active Directory Federation Services service.

Enable CORS on AD FS To Accept Requests from Polaris APIs

To enable CORS on AD FS to accept requests from Polaris APIs

- 1. Refer to the information on the following page:
 - <u>https://docs.microsoft.com/en-us/windows-server/identity/ad-</u> <u>fs/operations/customize-http-security-headers-ad-fs#cross-origin-resource-</u> <u>sharing-cors-headers</u>
- 2. Use the following commands:
 - Set-AdfsResponseHeaders -EnableCORS \$true
 - Set-AdfsResponseHeaders -CORSTrustedOrigins https://rdpolaris.polarislibrary.com,https://example2.com

Note:

Replace https://rd-polaris.polarislibrary.com
and https://example2.com with your own URL or list of
URLs.

Set Up Web Services and Applications for OIDC with Active Directory and AD FS

To set up each of the following web services and applications, you must configure a .json file for each of the following:

- Polaris.Authentication (the application that authenticates Polaris users)
- Polaris.AuthenticationServices (the API service that provides backend support for authentication)
- PolarisAdmin (the web-based Polaris System Administration application)
- LeapWebApp (Leap)
- Polaris.ApplicationServices (Leap's API service)

The five .json files are all named appsettings.user.json, but they reside in different directories:

- C:\Program Files\Polaris\7.7\Polaris.Authentication
- C:\Program Files\Polaris\7.7\Polaris.AuthenticationServices
- C:\Program Files\Polaris\7.7\PolarisAdmin\assets
- C:\Program Files\Polaris\7.7\LeapWebApp
- C:\Program Files\Polaris\7.7\Polaris.ApplicationServices

This section contains the following topics:

- <u>Configure Polaris.Authentication for Use with AD FS</u>
- <u>Configure Polaris.AuthenticationServices for Use with AD FS</u>
- <u>Configure PolarisAdmin for Use with AD FS</u>
- Configure Polaris.ApplicationServices for Use with AD FS
- Configure LeapWebApp for Use with AD FS

Configure Polaris. Authentication for Use with AD FS

To configure Polaris.Authentication, update C:\Program Files\Polaris\7.7\Polaris.Authentication\appsettings.user.json. You will use several values copied from your identity provider.

To configure Polaris.Authentication

- 1. Open the following files in a text editor. You must run the editing application (for example, Notepad) as administrator.
 - C:\Program Files\Polaris\7.7\Polaris.Authentication\appsettings.user.json
 - C:\Program Files\Polaris\7.7\Polaris.Authentication\RELEASE-NOTES.md
- 2. In the RELEASE-NOTES.md file, copy the settings in the ADFS Template section. The image below shows the settings to copy.

```
## Configuring OIDC OAuth 2.0
### ADFS Template
. . .
"OAuth": {
  "Enabled": true,
 "SendOAuthAuthorityHeader": false,
  "Authorities": [
    {
      "IsActive": true,
      "Name": "ADFS",
      "Authority": "https://[adfs-server-address]/adfs/",
      "UseOidc": true,
      "UsePkce": true,
      "EndSessionEndpoint": "https://[adfs-server-address]/adfs/oauth2/logout",
      "ClientId": "[client-id]",
      "ClientSecret": "[client-secret]",
      "CallbackPath": "/oauth/callback",
      "SignedOutCallbackPath": "/signout-callback-oidc",
      "SignedOutRedirectUri": "/login",
      "AlternateUpnClaimType": "email",
      "AlternateLogoutUri": null,
      "Scopes": [
        "profile",
        "openid"
      ],
      "SaveTokens": false,
      "MetaAddress": "https://[adfs-server-address]/adfs/.well-known/openid-configuration",
      "ResponseMode": "form_post",
      "ResponseType": "code"
    }
 ]
```

- 3. In the .json file, replace the entire contents of the file with the settings you copied from the RELEASE-NOTES.md file.
- 4. Verify that Enabled is set to true.
- 5. Verify that IsActive is set to true.
- 6. Verify that UseOidc is set to true.
- 7. Update the Authority and EndSessionEndpoint properties with values from your AD FS well-known configuration URL.

Note: Sample well-known configuration URL: https://adfs-server-address/adfs/.well-known/openid-configuration

Replace *adfs-server-address* with the AD FS server address.

- a. Replace the Authority property value with the issuer value from the well-known configuration URL.
- b. Replace the EndSessionEndpoint property value with the end_session_ endpoint value from the well-known configuration URL.
- 8. On the AD FS server, open the AD FS Management desktop application.

 Ŷ AD FS Ŷ File Action View Window Help ◆ ◆ 2 □ ↓ 2 □ □ 			
AD FS	Application Groups		
Service Access Control Policies Relying Party Trusts	Name Polaris	Description Polaris web applications	
Application Groups	Polaris Properties		×
	Genera Polaris - Native ap	oplication Properties	×
	Pola Name:		
	Polaris - Native a	application	
	Dient Id: Da586b1e-eeb0	4c8a-8381-50e9cafec240	
	Appli Redirect URI:	Add	
	Na https://rd-polaris https://rd-polaris https://rd-polaris	polarislibrary.com/Polaris.Admin.Servi polarislibrary.com/PolarisAdmin// polarislibrary.com/PolarisAdmin/login polarislibrary.com/PolarisAdmin/login	
	Poi Description:		
	Ad	OK Cancel Apply	•
		OK Cancel	Apply

9. Copy the client ID from the Polaris - Native application properties dialog.

- 10. In the .json file, update the ClientId property. Replace [client-id] with the client ID copied from the AD FS Management desktop application.
- 11. Update the Scopes property if necessary for your AD FS configuration.

Note:

The default values provided for the Scopes property are appropriate for most AD FS configurations. However, it is possible that your configuration may use a different scope, for example, email.

- 12. In the MetaAddress property, replace [adfs-server-address] with the AD FS server address.
- 13. Save the .json file.

Configure Polaris. Authentication Services for Use with AD FS

To configure Polaris.AuthenticationServices, update C:\Program Files\Polaris\7.7\Polaris.AuthenticationServices\appsettings.user.json. You will use several values copied from your identity provider.

Important:

Polaris 7.7 includes a RELEASE-NOTES.md file that contains template settings that apply to AD FS. See the OIDC/OAuth Setup for AD FS section of the release notes file.

To configure Polaris.AuthenticationServices

- 1. Open the following files in a text editor. You must run the editing application (for example, Notepad) as administrator.
 - C:\Program Files\Polaris\7.7\Polaris.AuthenticationServices\appsettings.user.json
 - C:\Program Files\Polaris\7.7\Polaris.AuthenticationServices\RELEASE-NOTES.md
- 2. In the RELEASE-NOTES.md file, copy the settings in the OIDC/OAuth Setup for AD FS section. The image below shows the settings to copy.

```
### OIDC/OAuth Setup for AD FS
"OAuth": {
 "Enabled": true,
 "Authorities": [
     "Name": "ADFS",
     "UseOidc": true,
      "RequireHttpsMetadata": true,
      "RequireSignedTokens": true,
      "ValidateIssuer": true,
     "ValidateAudience": true,
     "Authority": "https://dev-fs.polarislibrary.com/adfs/",
     "Audience": "microsoft:identityserver:0a586b1e-eeb0-4c8a-8381-50e9cafec240",
     "MetaAddress": "https://dev-fs.polarislibrary.com/adfs/.well-known/openid-configuration",
      "ValidIssuers": [
        "https://dev-fs.polarislibrary.com/adfs",
        "http://dev-fs.polarislibrary.com/adfs/services/trust"
      ],
      "ValidAudiences": [
        "0a586b1e-eeb0-4c8a-8381-50e9cafec240",
        "microsoft:identityserver:0a586b1e-eeb0-4c8a-8381-50e9cafec240"
      ],
      "UpnClaimTypes": [ "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn" ],
      "IsUpnExternalId": false
```

- 3. In the .json file, replace the entire contents of the file with the settings you copied from the RELEASE-NOTES.md file.
- 4. Update the following properties. In each, replace dev-

fs.polarislibrary.com with the AD FS server address:

- Authority
- MetaAddress
- ValidIssuers
- 5. On the AD FS server, open the AD FS Management desktop application.



- 6. Copy the client ID from the Polaris Native application properties dialog.
- 7. Update the following properties. In each, replace *0a586b1e-eeb0-4c8a-8381-50e9cafec240* with the client ID copied from the AD FS Management desktop application:
 - Audience
 - ValidAudiences
- 8. In the UpnClaimTypes property, update the default value if you want to specify a different claim to serve as the user identifier. The default value applies to most

configurations, but you can specify a different claim that exists in a JSON Web Token (JWT).

9. Save the .json file.

Configure PolarisAdmin for Use with AD FS

To configure PolarisAdmin, update C:\Program Files\Polaris\7.7\PolarisAdmin\assets\appsettings.user.json.

To configure PolarisAdmin

- Open the C:\Program
 Files\Polaris\7.7\PolarisAdmin\assets\appsettings.user.json file in a text editor.
 You must run the editing application (for example, Notepad) as administrator.
- 2. In the .json file, update the server location in the apiUrlRoot. If you started from the template settings provided in the RELEASE-NOTES.md file, replace [my-server-domain-name] with the FQDN of the server that hosts both the API service for Polaris System Administration (web-based) and the Authentication Application.
- 3. Update the server location in the authAppUrl. If you started from the template settings provided in the RELEASE-NOTES.md file, replace [my-server-domain-name] with the FQDN of the server that hosts both the API service for Polaris System Administration (web-based) and the Authentication Application.
- 4. Save the .json file.

Configure LeapWebApp for Use with AD FS

Once you have made the changes described in the <u>Upgrading to Polaris 7.7</u> section, you only need to do additional configuration for LeapWebApp if one or both of the following is true:

- You want to enable permission overrides in Leap.
- You want to set ReAuthDisabled to false for Leap.

If neither of the above conditions is true, skip to the next section. See <u>Customize the</u> AD FS Pages.

Note:

Permission overrides and reauthentication are not supported if your system uses multiple identity providers for authentication.

To configure LeapWebApp, update C:\Program

Files\Polaris\7.7\LeapWebApp\appsettings.user.json. You will use several values copied from your identity provider.

To configure LeapWebApp

- 1. Open the following files in a text editor. You must run the editing application (for example, Notepad) as administrator.
 - C:\Program Files\Polaris\7.7\LeapWebApp\appsettings.user.json
 - C:\Program Files\Polaris\7.7\LeapWebApp\RELEASE-NOTES.md
- 2. In the RELEASE-NOTES.md file, copy the settings in the OIDC/OAuth Setup for AD FS section. The image below shows the settings to copy.



- 3. In the .json file, replace the OAuth contents of the file with the settings you copied from the RELEASE-NOTES.md file.
- 4. Update the following properties. In each, replace *dev*-*fs.polarislibrary.com* with the AD FS server address:
 - Authority
 - MetadataAddress
 - KnownAuthorities
- 5. On the AD FS server, open AD FS Management desktop application.



- 6. Copy the client ID from the Polaris Native application properties dialog.
- 7. In the .json file, update the ClientId property. Replace 3eb2a79f-db5a-4ba0b22f-e7d16a616d4a with the client ID copied from the AD FS Management desktop application.
- 8. Save the .json file.

Configure Polaris.ApplicationServices for Use with AD FS

To configure Polaris.ApplicationServices, update C:\Program Files\Polaris\7.7\Polaris.ApplicationServices\appsettings.user.json. You will use several values copied from your identity provider.

To configure Polaris.ApplicationServices

- 1. Open the following files in a text editor. You must run the editing application (for example, Notepad) as administrator.
 - C:\Program Files\Polaris\7.7\Polaris.ApplicationServices\appsettings.user.json
 - C:\Program Files\Polaris\7.7\Polaris.ApplicationServices\RELEASE-NOTES.md
- 2. In the RELEASE-NOTES.md file, copy the settings in the **Example OAuth2 Settings** section. The image below shows the settings to copy.

```
## Example OAuth2 Settings:
  "0Auth": {
      "Enabled": false,
      "Authorities": [
        {
          "Name": "ADFS",
          "Authority": "https://dev-fs.polarislibrary.com/adfs/",
          "Audience": "microsoft:identityserver:3eb2a79f-db5a-4ba0-b22f-e7d16a616d4a",
          "MetaAddress": "https://dev-fs.polarislibrary.com/adfs/.well-known/openid-configuration",
          "RequireHttpsMetadata": true,
          "RequireSignedTokens": true,
          "ValidateIssuer": true,
          "ValidIssuers": [
            "https://dev-fs.polarislibrary.com/adfs",
            "http://dev-fs.polarislibrary.com/adfs/services/trust"
          ],
          "ValidateAudience": true,
          "ValidAudiences": [
            "3eb2a79f-db5a-4ba0-b22f-e7d16a616d4a",
            "microsoft:identityserver:3eb2a79f-db5a-4ba0-b22f-e7d16a616d4a"
          ],
          "UPNClaimTypes": [ "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn" ],
          "IsUPNExternalID": false,
          "OpaqueToken": false,
          "UserInformationEndpoint": null
        }
      ],
       'Swagger": {
        "ClientID": "3eb2a79f-db5a-4ba0-b22f-e7d16a616d4a",
        "ClientSecret": "",
        "AppName": "Polaris.ApplicationServices",
        "AuthorizationUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/authorize",
        "TokenUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/token",
        "RefreshTokenUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/token",
        "LogoutUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/logout",
        "Scopes": [
            "Name": "openid", "Description": "Use OIDC to verify the user's identity" }, "Name": "email", "Description": "Optional to return user's email address" },
                                                                                          • },
            "Name": "urn:microsoft:userinfo", "Description": "urn:microsoft:userinfo" }
          {
        ]
     }
    },
```

3. On the AD FS server, open AD FS Management desktop application.



- 4. Copy the client ID from the Polaris Native application properties dialog. You will use this value in the steps below.
- 5. In the Authorities property, update the following properties:
 - a. In the Authority property, replace *dev-fs.polarislibrary.com* with the AD FS server address.
 - b. In the Audience property, replace 3eb2a79f-db5a-4ba0-b22fe7d16a616d4a with the client ID copied from the AD FS Management desktop application.

- c. In the MetaAddress property, replace *dev-fs.polarislibrary.com* with the AD FS server address.
- d. In both URLs in the ValidIssuers property, replace *dev*-*fs.polarislibrary.com* with the AD FS server address.
- e. In both values in the ValidAudiences property, replace 3eb2a79f-db5a-4ba0-b22f-e7d16a616d4a with the client ID copied from the AD FS Management desktop application.
- f. In the UpnClaimTypes property, update the default value if you want to specify a different claim to serve as the user identifier. The default value applies to most configurations, but you can specify a different claim that exists in a JSON Web Token (JWT).
- 6. In the Swagger property, update the following properties with information from your identity provider:
 - a. In the ClientID property, replace 3eb2a79f-db5a-4ba0-b22fe7d16a616d4a with the client ID copied from the AD FS Management desktop application.
 - b. In the AuthorizationUrl property, replace *devfs.polarislibrary.com* with the AD FS server address.
 - c. In the TokenUrl property, replace *dev-fs.polarislibrary.com* with the AD FS server address.
 - d. In the RefreshTokenUrl property, replace *devfs.polarislibrary.com* with the AD FS server address.
 - e. In the LogoutUrl property, replace *dev-fs.polarislibrary.com* with the AD FS server address.
- 7. Save the .json file.

Customize the AD FS Pages

Use the following resources to customize AD FS pages:

<u>https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/dn280950(v=ws.11)</u>

- Get-AdfsGlobalWebContent
- Set-AdfsGlobalWebContent

Examples: Customize the examples below to suit your library's needs.

```
PS C:\Windows\system32> Set-AdfsGlobalWebContent -
SignOutPageDescriptionText "You have successfully signed
out.<br>If you have been directed here immediately after
signing in, your session may have timed out."
```

PS C:\Windows\system32> Set-AdfsWebTheme -TargetName
default -Logo @{path="c:\ADFS Custom\leap logo.png"}

```
PS C:\Windows\system32> Set-AdfsGlobalWebContent -
CompanyName "Polaris R&D"
```

- Advanced customization:
 - <u>https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/dn636121(v=ws.11)</u>

Change the Access Token Lifetime

The default token lifetime for both access and ID tokens is 60 minutes. Execute the following command to increase the expiration time to 24 hours:

```
Set-AdfsWebApiApplication -TokenLifetime 1440 -TargetIdentifier "0a586ble-eeb0-4c8a-8381-50e9cafec240"
```

Note:

Replace *TargetIdentifier* with the Polaris Application Group native application client ID.

Bind a New SSL Certificate

If your web server certificate expires, use the instructions below to bind a new SSL certificate.

To bind a new SSL certificate

- 1. Install the certificate using Certificates Management.
- 2. Set the service communications certificate using the AD FS Management Console:
 - a. Expand the Services folder.
 - b. Select a new certificate.
 - c. Restart the AD FS service.
- 3. Attach the certificate to AD FS using PowerShell:
 - a. Get the certificate's thumbprint by viewing the certificate.

c:\> Set-AdfsSslCertificate -Thumbprint e8fd5016542796214e94f72d76095f9fc587c731

b. Restart the AD FS service.

Troubleshoot

Force a logout

• https://AD FS server address/adfs/oauth2/logout

Note:

Replace AD FS server address with your library's AD FS server address.

AD FS in one-way trust

Problem: Only local accounts are authenticating

Solution: Make sure the account running the AD FS service is a parent domain account and not a local account.

Receiving "User is not a valid Polaris user." error

• Check the setting Polaris.OAuth.ValidIssuer in the Polaris.AdminServices appsettings.user.json file.

Example value: http://AD FS server address/adfs/services/trust

Note: Replace *AD FS server address* with your library's AD FS server address.

• Verify a domain is attached to AD user accounts so the UPN claim can be added to the ID token's claims.

The UPN claim should look like user@mydomain.com.

Troubleshoot Redirect URIs

Redirect URIs are case-sensitive.

Configuring OIDC with Azure AD

Important:

The mechanism used to connect an Azure AD user to a Polaris user is the user principal name (UPN) in the format of an email address. For example, user@mydomain.com.

During the account verification process, we use the openid and profile scopes, which triggers Azure AD to return the upn claim or the preferred_username claim (or both). These must be returned in the name@domain format. The Polaris.ApplicationServices (API) can then use that information to map the Azure AD user to a Polaris user. If the preferred_username is a generic name, phone number, or other value, you can choose to apply the email scope to return the email.

See <u>Configure LeapWebApp for Use with Azure AD</u> for more information.

Note:

Azure AD has recently been rebranded as "Microsoft Entra ID". In this guide, references to Azure AD are interchangeable with Microsoft Entra ID.

To configure OIDC with Azure AD, perform the following tasks:

- 1. <u>Register LeapWebApp with Azure AD</u>.
- 2. Create client credentials.
- 3. Add authentication redirect URIs.
- 4. Expose the Polaris.ApplicationServices API.
- 5. Configure an ID token.
- 6. <u>Set up users and groups</u>.
- 7. <u>Control access to LeapWebApp using Azure AD</u>.
- 8. Set up web services and applications.

After you complete these tasks, Add a URL rewrite rule for LeapWebApp.

Register LeapWebApp with Azure AD

To register LeapWebApp with Azure AD

1. Sign in to the Azure portal:

https://portal.azure.com/

2. In the Azure services list, select Azure Active Directory.

Azure servic	ces				
+	٠	×	•	۲	
Create a resource	Azure Active Directory	Quickstart Center	Virtual machines	App Services	Storage accounts

3. In the list of options at the left side of the screen, select **App registrations**.

0	Overview
++	Preview features
×	Diagnose and solve problems
Ma	nage
2	Users
24	Groups
Û	External Identities
2,	Roles and administrators
8	Administrative units
14	Enterprise applications
	Devices
Щ,	App registrations
۵	Identity Governance

The App registrations page appears.

4. Select New registration.

The Register an application dialog appears.

Register an application				
* Name				
The user-facing display name for this application (this can be changed later).				
LeapWebApp 🗸				
Supported account types				
Who can use this application or access this API?				
 Accounts in this organizational directory only (Jeffrey Young only - Single tenant) 				
O Accounts in any organizational directory (Any Azure AD directory - Multitenant)				
O Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)				
O Personal Microsoft accounts only				
Help me choose				
Redirect URI (optional)				
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.				
Web V https://[your-fqdn]/leapwebapp/signin-oidc				

- 5. Enter "LeapWebApp" in the **Name** field.
- 6. Select an option from the **Supported account types** list.
- 7. Add a redirect URI:
 - a. Select the **Web** URI type.
 - b. Enter an address that uses the following format:

https://[FQDN]/leapwebapp/signin-oidc

Notes:

- Replace [FQDN] with the fully-qualified domain name of your LeapWebApp server.
- Example: https://leap.mylibrary.org/leapwebapp/signinoidc
- 8. Select **Register**.

The page for your new LeapWebApp application appears.

Home > Jeffrey Young >					
🌉 LeapWebApp 🖉 🖤					×
✓ Search (Ctrl+/) «	📋 Delete Endpoints	Review features			
🐺 Overview					
🗳 Quickstart	∧ Essentials				
💉 Integration assistant	Display name	: LeapWebApp	Client credentials	: Add a certificate or secret	
Manage	Object ID	: 9f2cda2a-804e-4eab-9f12-7f8eeec1c8de	Application ID URI	: Add an Application ID URI	
Branding & properties	Directory (tenant) ID	: 3e8e440e-d3c1-459d-87a0-a75887f696cc	Managed application in	: LeapWebApp	
Authentication	Supported account types	: My organization only			
Certificates & secrets					
Token configuration	Get Started Docume	intation			
->- API permissions					
🔷 Expose an API		Build your application with th	e Microsoft ic	lentity platform	
App roles		The Microsoft identity platform is an authentication service,	open-source libraries, and	application management tools.	
A Owners	You can create modern, standards-based authentication solutions, access and protect APIs, and add sign-in for your users and customers. Learn more 🖉				
and administrators					
Manifest		la 🔊 👝 👎 📧			
Support + Troubleshooting			22	Ó	
Troubleshooting		🔽 🚺 🤹 🔪 🚺	•		
New support request			Sign in usors in E.m.	inutor	
		Build More powerful apps with rich user and business data from Microsoft services and your own company's data sources. View API permissions	Use our SDKs to sign in steps. Use the quickstar app, SPA, or daemon ap View all quickstart g	users and call APIs in a few is to start a web app, mobile p. guides	

9. Copy the application (client) ID and paste it into Notepad (or a similar text editor) and save the file. You must have this value to complete several procedures later in the Azure AD configuration process.

In the example below, the application (client) ID is "efc04e56-0777-45a1-b7c5-793dbc1dbbd68".

🔣 LeapWebApp 🖉					
✓ Search (Ctrl+/) «	📋 Delete ⊕ Endpoints 💀 Preview features				
Overview					
📣 Quickstart					
	Display name : <u>LeapWebApp</u>				
🚀 Integration assistant	Application (client) ID : efc04e56-0777-45a1-b7c5-793dbc1dbd68				
Manage	Object ID : 9f2cda2a-804e-4eab-9f12-7f8eeec1c8de				
🔤 Branding & properties	Directory (tenant) ID : 3e8e440e-d3c1-459d-87a0-a75887f696cc				
Authentication	Supported account types : <u>My organization only</u>				

Create Client Credentials

To create client credentials for LeapWebApp

1. On the LeapWebApp page, select **Add a certificate or secret**.

🔣 LeapWebApp 🖉 ··				
	📋 Delete 🌐 Endpoint	ts 💀 Preview features		
Overview				
🗳 Quickstart				
🚀 Integration assistant	Display name	: LeapWebApp	Client credentials	Add a certificate or secret
	Application (client) ID	: efc04e56-0777-45a1-b7c5-793dbc1dbd68	Redirect URIs	: 1 web, 0 spa, 0 public client
Manage	Object ID	: 9f2cda2a-804e-4eab-9f12-7f8eeec1c8de	Application ID URI	: Add an Application ID URI
Branding & properties	Directory (tenant) ID	: 3e8e440e-d3c1-459d-87a0-a75887f696cc	Managed application in	I : <u>LeapWebApp</u>
Authentication	Supported account type	es : <u>My organization only</u>		

The Certificates & secrets page appears.

- 2. Select the **Client secrets** tab.
- 3. Select New client secret.

🔶 LeapWebApp Certificates & secrets 👒 …				
	📯 Got feedback?			
Overview	Credentials enable confidential applications to identify themselves to the authentica	ation		
🗳 Quickstart	scheme). For a higher level of assurance, we recommend using a certificate (instead of a			
🚀 Integration assistant				
Manage	() Application registration certificates, secrets and federated credentials can be found	l in th		
🔤 Branding & properties				
Authentication	Certificates (0) Client secrets (0) Federated credentials (0)			
📍 Certificates & secrets	A secret string that the application uses to prove its identity when requesting a to	ken. A		
Token configuration				
API permissions	+ New client secret			
🙆 Expose an API	Description Expires Value 🛈			
App roles	No client secrets have been created for this application.			

The Add a client secret dialog appears.

- 4. Enter a description in the **Description** field.
- 5. Select an option from the **Expires** list to specify when the client credentials expire.
- 6. Select Add.

The Add a client secret dialog closes. The client secret for LeapWebApp appears on the **Client secrets** tab of the Certificates & secrets page.

7. Copy the text in the **Value** column, then paste it into Notepad (or a similar text editor) and save the file. You must have this value to complete the <u>Configure</u> <u>LeapWebApp for Use with Azure AD</u> procedure.

Important:

- You must save this value now. Only a portion of the value appears when you return to the page later.
- Use the copy icon to be sure that you are copying the entire value.

In the example below, the value is "4I.8Q~-GpkdoymMQneGIYNg40FRjx2Hr1wWLDcbr".

Provide the second se					
✓ Search (Ctrl+/) «	🔗 Got feedback?				
👯 Overview					
🗳 Quickstart	\bigcirc Got a second to give us some feedback? \rightarrow				×
🚀 Integration assistant					
Manage	Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.				
Branding & properties					
Authentication	Application registration certificates, secrets and federated credentials can be found in the tabs below.				×
📍 Certificates & secrets					
Token configuration	Certificates (0) Client secrets (1) Feder	ated credentials (0)			
->- API permissions	Δ secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.				
🙆 Expose an API	2 Fr	, , , ,			
👪 App roles	+ New client secret				
🐣 Owners	Description	Expires	Value 🛈	Secret ID	
a. Roles and administrators	Leap client secret	1/6/2023	4I.8Q~-GpkdoymMQneGIYNg40FRjx2Hr	f60a4a23-049f-4117-9bc1-3dce57cedec4	D 📋
0 Manifest				_	

Add Authentication Redirect URIs

To add authentication redirect URIs

1. On the LeapWebApp page, select the link beside **Redirect URIs**.

🜉 LeapWebApp 🖈 …				
Search (Ctrl+/)	📋 Delete ⊕ Endpoint	ts 🐱 Preview features		
Overview	 For a distance 			
📣 Quickstart	A Essentiais			
A	Display name	: <u>LeapWebApp</u>	Client credentials	: <u>0 certificate, 1 secret</u>
Integration assistant	Application (client) ID	: efc04e56-0777-45a1-b7c5-793dbc1dbd68	Redirect URIs	: <u>1 web, 0 spa, 0 public client</u>
Manage	Object ID	: 9f2cda2a-804e-4eab-9f12-7f8eeec1c8de	Application ID URI	: Add an Application ID URI
🔤 Branding & properties	Directory (tenant) ID	: 3e8e440e-d3c1-459d-87a0-a75887f696cc	Managed application in I	: <u>LeapWebApp</u>
Authentication	Supported account type	es : <u>My organization only</u>		

The Authentication page appears.

2. Select Add a platform.

The Configure platforms dialog appears.

3. Select the **Single-page application** tile.

The Configure single-page application dialog appears.

- 4. In the **Redirect URIs** field, enter the following required redirect URIs:
 - https://[FQDN]/polarisauth/login
 - https://[FQDN]/polarisauth/logout
 - https://[FQDN]/polarisauth/signin-oidc

Notes:

- Replace [FQDN] with the fully-qualified domain name of your LeapWebApp server.
- Example: https://leap.mylibrary.org/polarisauth/login
- 5. If you are using permission overrides or reauthentication in Leap, enter the following additional redirect URIs in the **Redirect URIs** field:
 - https://[FQDN]/leapwebapp/signin-override-oidc
 - https://[FQDN]/leapwebapp/silent-logout-msal
- 6. If you are using Swagger, enter the following additional redirect URIs in the **Redirect URIs** field:
- https://[FQDN]/Polaris.ApplicationServices/swagger/oauth2-redirect.html
- https://[FQDN]/Polaris.AdminServices/swagger/oauth2-redirect.html
- 7. Select **Save** to save the redirect URIs.
- 8. Return to the Configure platforms dialog.
- 9. Select the Web tile.

The Configure web dialog appears.

- 10. In the **Redirect URIs** field, enter the following redirect URIs:
 - https://[FQDN]/polarisauth/signin-oidc

https://[FQDN]/polarisauth/login

Notes:

- Replace [FQDN] with the fully-qualified domain name of your LeapWebApp server.
- Example: https://leap.mylibrary.org/polarisauth/signin-oidc
- 11. Select **Configure**.

The new redirect URI appears on the Authentication page in the **Web** list.

12. Select **Save** to save the web redirect URIs.

Expose the Polaris.ApplicationServices API

To expose the Polaris.ApplicationServices API

1. On the LeapWebApp page, select Add an Application ID URI.

Note: You can also select **Expose an API** in the list of options at the left side of the screen.

👯 LeapWebApp 🖈 …				
	📋 Delete 🌐 Endpoin	is 🐱 Preview features		
Overview	∧ Essentials			
QuickstartIntegration assistant	Display name	: LeapWebApp	Client credentials	: <u>0 certificate, 1 secret</u>
Manage	Object ID	: 9f2cda2a-804e-4eab-9f12-7f8eeec1c8de	Application ID URI	: Add an Application ID URI
 Branding & properties Authentication 	Directory (tenant) ID Supported account typ	: 3e8e440e-d3c1-459d-87a0-a75887f696cc es : <u>My organization only</u>	Managed application in I.	: <u>LeapWebApp</u>
Certificates & secrets Token configuration	Get Started Docum	nentation		
 API permissions Expose an API 		Build your application with th	e Microsoft id	entity platform

The Expose an API page appears.

2. Select Add a scope.

The Add a scope dialog appears. The **Application ID URI** field contains an automatically-generated URI.



3. Select Save and continue.

The Add a scope dialog refreshes.

Add a scope	×
Scope name * 🛈	
pas	\sim
api://efc04e56-0777-45a1-b7c5-793dbc1dbd68/pas	
Who can consent? ()	
Admins and users Admins only	
Admin consent display name * 🛈	
Access Polaris.ApplicationServices	\checkmark
Admin consent description * ①	
Allows the app to access the Polaris.ApplicationServices web API.	~

- 4. Enter "pas" in the **Scope name** field.
- 5. Enter "Access Polaris.ApplicationServices" in the **Admin consent display name** field.
- 6. Enter "Allows the app to access the Polaris.ApplicationServices web API." in the **Admin consent description** field.
- 7. Select Add scope.

The Azure portal saves the scope and closes the Add a scope dialog.

8. On the Expose an API page, select **Add a client application**.

The Add a client application dialog appears.



- 9. In the **Client ID** field, enter the Application (client) ID that you copied and saved during the **Register LeapWebApp** with Azure AD procedure.
- 10. Select the **Authorized scopes** checkbox.
- 11. Select Add application.

The Azure portal saves your changes and closes the Add a client application dialog.

12. On the Expose an API page, copy the new scope, then paste it into Notepad (or a similar text editor) and save the file. Your value will be similar to this one:

api://efc04e56-0777-45a1-b7c5-793dbc1dbd68/pas

You must have this value to complete two procedures later in the Azure AD configuration process.

Configure an ID Token

To allow Leap to sign out of specific accounts, you must add an ID token that contains the login_hint claim.

To configure an ID token

1. On the LeapWebApp Overview page, select **Token configuration** from the list of options at the left side of the screen.

The Token configuration page appears.

2. Select Add optional claim.

The Add optional claim dialog appears.

Add optional clain	n ×			
Once a token type is selected, you may choose from a list of available optional claims.				
 Token type Access and ID tokens are used by a ID Access SAML 	applications for authentication. Learn more a			
■ Claim ↑↓	Description			
acct	User's account status in tenant			
auth_time	Time when the user last authenticated; See OpenID Con			
C ctry	User's country/region			
email	The addressable email for this user, if the user has one			
family_name	Provides the last name, surname, or family name of the			
fwd	IP address			
given_name	Provides the first or "given" name of the user, as set on			
in_corp	Signals if the client is logging in from the corporate net			
🗌 ipaddr	ipaddr The IP address the client logged in from			
✓ login_hint	Login hint			
onprem_sid	On-premises security identifier			

- 3. Set the **Token type** setting to the **ID** option.
- 4. Select the **login_hint** checkbox.
- 5. Select Add.

The Azure portal saves the token and closes the Add optional claim dialog.

6. Verify that the new login_hint claim appears on the Token configuration page.

LeapWebApp Token configuration 🖉 …				
	R Got feedback?			
 Overview Quickstart Integration assistant 	Optional claims Optional claims are used to configure additional information which is returned in one or more tokens. Learn more additional optional claim + Add groups claim			
Manage				
Branding & properties	Claim 1. Description	Token type \uparrow_\downarrow		
Certificates & secrets	login_hint Login hint	ID		
Token configuration				

Set Up Users and Groups

To set up users and groups

1. On the Azure AD Overview page, select the **Enterprise applications** option from the list at the left side of the screen.

Note:

You can also select **Enterprise applications** from the list of services on the Azure portal home page.

The All applications page appears.

2. Select the LeapWebApp link.

The LeapWebApp Overview page appears.

LeapWebApp Overview						
~						
👢 Overview	Properties					
Deployment Plan	LE Name () Copy to clipboard					
Manage	LeapWebApp 🗈					
Broperties	Application ID 🛈					
Properties	efc04e56-0777-45a1-b7c5 🗈					
A Owners Object ID ①						
Boles and administrators	71ed79e9-bedd-451e-9323 🗈					
Users and groups	Getting Started					
Single sign-on						
Provisioning						
Application proxy	1. Assign users and groups					
elf-service	Provide specific users and groups access to the applications					
Custom security attributes (preview)	Assign users and groups					
Security						

3. Select the **Users and groups** option from the list at the left side of the screen. You can also select the **Assign users and groups** tile.

The Users and groups page appears.

4. Select Add user/group.

The Add Assignment page appears.

5. Select the **None Selected** link.

The Users and groups dialog appears.

Users	s and groups $\qquad \qquad \qquad$
₽ Sear	ch
СО	cogswell cogswell@jeffyoungpolarislibrary.onmicrosoft.com Selected
YL	Jeffrey Young jeff.young@polarislibrary.com
LU	Leap Users
PS	Polaris Supervisor polaris.supervisor@jeffyoungpolarislibrary.onmicrosoft.com Selected

- 6. Select the users and groups that you want to allow access to LeapWebApp.
- 7. Click Select.

The Users and groups dialog closes.

8. On the Add Assignment page, select **Assign**.

The Azure portal saves the user and group assignments.

Control Access to LeapWebApp Using Azure AD

To control access to LeapWebApp using Azure AD

1. On the Azure AD Overview page, select the **Enterprise applications** option from the list at the left side of the screen.

Note:

You can also select **Enterprise applications** from the list of services on the Azure portal home page.

The All applications page appears.

2. Select the LeapWebApp link.

The LeapWebApp Overview page appears.

🙏 LeapWebApp - Microsoft Azure 🗙	+			
\leftrightarrow \rightarrow C $$ portal.azure.com	n/#view/Microsoft_AAD_IAM/ManagedAppMenuBlac			
■ Microsoft Azure P Search resources, services, and docs (G+/)				
Home > Polaris Test AAD > Enterprise applications >				
Enterprise Application				
II. Overview	Properties			
🛄 Deployment Plan	Name ()			
Manage	LeapWebApp 🗈			
Properties	Application ID 🕢			
A Owners				
Roles and administrators	71ed79e9-bedd-451e-9323			

3. Select the **Properties** option from the list at the left side of the screen.

The Properties page appears.

- 4. Set the **Assignment requirement?** setting to **Yes**. This allows access to be controlled by the users and groups assigned to the LeapWebApp enterprise application. (When it is set to **No**, all users can sign in.)
- 5. Set the **Visible to users?** setting to **Yes**. This makes the LeapWebApp application visible to users in their Microsoft My Apps portal and on their Office 365 page.

My Apps X			1 Microsoft Office H	ome X	
← → C			\leftrightarrow \rightarrow C $$ office.com/?auth=2		
🔛 My Apps 🗸	Search a			Office \rightarrow	
Apps 🗸 🕀			Apps		
			Add-Ins	🕼 LeapWebApp	
Ê	() polaris		All apps $ ightarrow $		
Add-Ins	LeapWebApp				

Important:

When a user accesses LeapWebApp from the Microsoft My Apps portal or their Office 365 page, they might have to click the Polaris Leap Sign In button. This is because cookies are a part of the Leap authentication process.

6. Select Save.

The Azure portal saves your changes.

Set Up Web Services and Applications for OIDC with Azure AD

To set up each of the following web services and applications, you must configure a .json file for each of the following:

- Polaris.Authentication (the application that authenticates Polaris users)
- Polaris.AuthenticationServices (the API service that provides backend support for authentication)
- PolarisAdmin (the web-based Polaris System Administration application)
- LeapWebApp (Leap)
- Polaris.ApplicationServices (Leap's API service)

The five .json files are named appsettings.user.json, but they reside in different directories:

- C:\Program Files\Polaris\7.7\Polaris.Authentication
- C:\Program Files\Polaris\7.7\Polaris.AuthenticationServices
- C:\Program Files\Polaris\7.7\PolarisAdmin\assets
- C:\Program Files\Polaris\7.7\LeapWebApp
- C:\Program Files\Polaris\7.7\Polaris.ApplicationServices

This section contains the following topics:

- Configure Polaris.Authentication for Use with Azure AD
- Configure Polaris.AuthenticationServices for Use with Azure AD
- Configure PolarisAdmin for Use with Azure AD
- <u>Configure LeapWebApp for Use with Azure AD</u>
- Configure Polaris.ApplicationServices for Use with Azure AD

Configure Polaris. Authentication for Use with Azure AD

To configure Polaris.Authentication, update C:\Program Files\Polaris\7.7\Polaris.Authentication\appsettings.user.json. You will use several values copied from your identity provider.

Important:

By default, the appsettings.user.json template file contains configuration settings that apply to AD FS. Polaris 7.7 includes a RELEASE-NOTES.md file that contains the template settings that apply to Azure AD. See the Azure AD Example section of the release notes file.

To configure Polaris.Authentication

- 1. Open the following files in a text editor. You must run the editing application (for example, Notepad) as administrator.
 - C:\Program Files\Polaris\7.7\Polaris.Authentication\appsettings.user.json
 - C:\Program Files\Polaris\7.7\Polaris.Authentication\RELEASE-NOTES.md
- 2. In the RELEASE-NOTES.md file, copy the settings in the Azure AD Example. The image below shows the settings to copy.

```
### AzureAD Example
"OAuth": {
  "Enabled": true,
  "SendOAuthAuthorityHeader": false,
  "Authorities": [
     "IsActive": true,
      "Name": "AzureAD",
      "UseOidc": true,
      "UsePkce": true,
      "EndSessionEndpoint": "https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-
a75887f696cc/oauth2/v2.0/logout",
      "ClientId": "efc04e56-0777-45a1-b7c5-793dbc1dbd68",
     "ClientSecret": "YLU8Q~TGdT_KIcDZ5px643zcsLmCzr16S1jiKcZI",
      "CallbackPath": "/signin-oidc",
      "SignedOutCallbackPath": "/signout-callback-oidc",
      "SignedOutRedirectUri": "/login",
      "Authority": "https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-a75887f696cc/v2.0/",
      "MetaAddress": "https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-a75887f696cc/v2.0/.well-
known/openid-configuration",
      "ResponseMode": "form_post",
      "ResponseType": "code",
      "Scopes": [
        "profile",
        "openid"
      "SaveTokens": false
   }
 ]
```

- 3. In the .json file, replace the entire contents of the file with the settings you copied from the RELEASE-NOTES.md file.
- 4. In the EndSessionEndpoint property, make the following updates:
 - a. Replace login.microsoftonline.com with your Azure AD server address.
 - b. Replace 3e8e440e-d3c1-459d-87a0-a75887f696cc with the tenant ID copied from the Azure portal.
- 5. In the ClientId property, replace *efc04e56-0777-45a1-b7c5-793dbc1dbd68* with the application (client ID) you copied during the <u>Register</u> <u>LeapWebApp with Azure AD</u> step.
- 6. In the ClientSecret property, replace YLU8Q~TGdT_ KIcDZ5px643zcsLmCzr16S1jiKcZI with the client secret you copied during the Create Client Credentials step.
- 7. In the Authority property, make the following updates:

- a. Replace login.microsoftonline.com with your Azure AD server address.
- b. Replace 3e8e440e-d3c1-459d-87a0-a75887f696cc with the tenant ID copied from the Azure portal.
- 8. In the MetaAddress property, make the following updates:
 - a. Replace login.microsoftonline.com with your Azure AD server address.
 - b. Replace 3e8e440e-d3c1-459d-87a0-a75887f696cc with the tenant ID copied from the Azure portal.
- 9. Save the .json file.

Configure Polaris.AuthenticationServices for Use with Azure AD

To configure Polaris.AuthenticationServices, update C:\Program Files\Polaris\7.7\Polaris.AuthenticationServices\appsettings.user.json. You will use several values copied from your identity provider.

Important:

By default, the appsettings.user.json template file contains configuration settings that apply to AD FS. Polaris 7.7 includes a RELEASE-NOTES.md file that contains the template settings that apply to Azure AD. See the OIDC/OAuth Setup for AzureAD/Entra section of the release notes file.

To configure Polaris.AuthenticationServices

- 1. Open the following files in a text editor. You must run the editing application (for example, Notepad) as administrator.
 - C:\Program Files\Polaris\7.7\Polaris.AuthenticationServices\appsettings.user.json
 - C:\Program Files\Polaris\7.7\Polaris.AuthenticationServices\RELEASE-NOTES.md
- 2. In the RELEASE-NOTES.md file, copy the settings in the OIDC/OAuth Setup for AzureAD/Entra section. The image below shows the settings to copy.



- 3. In the .json file, replace the entire contents of the file with the settings you copied from the RELEASE-NOTES.md file.
- 4. Verify that Enabled is set to true.
- 5. Verify that UseOidc is set to true.
- 6. In the Authority property, do the following:
 - a. Replace *login.microsoftonline.com* with the server address for your identity provider.
 - b. Replace 0a586b1e-eeb0-4c8a-8381-50e9cafec240 with the application (client) ID you copied during the Register LeapWebApp with Azure AD step.
- 7. In the Audience property, replace *0a586b1e-eeb0-4c8a-8381-50e9cafec240* with the application (client) ID copied from the Azure AD portal.
- 8. In the MetaAddress property, do the following:
 - a. Replace *login.microsoftonline.com* with the server address for your identity provider.

- b. Replace 0a586b1e-eeb0-4c8a-8381-50e9cafec240 with the application (client) ID copied from the Azure AD portal.
- 9. In the ValidIssuers property, do the following:
 - a. Replace *login.microsoftonline.com* with the server address for your identity provider.
 - b. Replace *0a586b1e-eeb0-4c8a-8381-50e9cafec240* with the application (client) ID copied from the Azure AD portal.
- 10. In the ValidAudiences property, replace *0a586b1e-eeb0-4c8a-8381-50e9cafec240* with the application (client) ID copied from the Azure AD portal.
- 11. In the UpnClaimTypes property, update the default value if you want to specify a different claim to serve as the user identifier. The default value applies to most configurations, but you can specify a different claim that exists in a JSON Web Token (JWT).
- 12. Save the .json file.

Configure PolarisAdmin for Use with Azure AD

To configure PolarisAdmin, update C:\Program Files\Polaris\7.7\PolarisAdmin\assets\appsettings.user.json.

To configure PolarisAdmin

- Open the C:\Program
 Files\Polaris\7.7\PolarisAdmin\assets\appsettings.user.json file in a text editor.
 You must run the editing application (for example, Notepad) as administrator.
- 2. In the .json file, update the server location in the apiUrlRoot. If you started from the template settings provided in the RELEASE-NOTES.md file, replace [my-server-domain-name] with the FQDN of the server that hosts both the API service for Polaris System Administration (web-based) and the Authentication Application.
- 3. Update the server location in the authAppUrl. If you started from the template settings provided in the RELEASE-NOTES.md file, replace [my-server-domain-name] with the FQDN of the server that hosts both the API service for Polaris System Administration (web-based) and the Authentication Application.
- 4. Save the .json file.

Configure LeapWebApp for Use with Azure AD

Once you have made the changes described in the <u>Upgrading to Polaris 7.7</u> section, you only need to do additional configuration for LeapWebApp if one or both of the following is true:

- You want to enable permission overrides in Leap.
- You want to set ReAuthDisabled to false for Leap.

If neither of the above conditions is true, skip to the next section. See <u>Configure</u> Polaris.ApplicationServices for Use with Azure AD.

Note:

Permission overrides and reauthentication are not supported if your system uses multiple identity providers for authentication.

To configure LeapWebApp, update C:\Program

Files\Polaris\7.7\LeapWebApp\appsettings.user.json using the following information:

- Endpoint URIs copied from the Azure portal
- Values that you copied and saved during earlier steps in the Azure AD configuration process

To configure LeapWebApp

- 1. In the Azure portal, select **App registrations** from the list of options at the left side of the screen.
- 2. Select LeapWebApp.
- 3. On the LeapWebApp page, select **Endpoints**.

賎 LeapWebApp 🖉 🗉				
	📋 Delete 🕀 Endpoint	s 🐱 Preview features		
🖶 Overview				
🗳 Quickstart				
d Intervetion assistant	Display name	: <u>LeapWebApp</u>	Client credentials	Add a certificate or secret
F integration assistant	Application (client) ID	: efc04e56-0777-45a1-b7c5-793dbc1dbd68	Redirect URIs	: <u>1 web, 0 spa, 0 public client</u>
Manage	Object ID	: 9f2cda2a-804e-4eab-9f12-7f8eeec1c8de	Application ID URI	: Add an Application ID URI
Branding & properties	Directory (tenant) ID	: 3e8e440e-d3c1-459d-87a0-a75887f696cc	Managed application in I.	: <u>LeapWebApp</u>
Authentication	Supported account type	es : <u>My organization only</u>		

The Endpoints dialog appears. Leave this browser tab open so that you can copy endpoint URIs from it and paste them into the LeapWebApp appsettings.user.json file.

Endpoints	×
OAuth 2.0 authorization endpoint (v2)	Copy to clipboard
https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-a75887f696cc/oauth2/v2.0/authorize	
OAuth 2.0 token endpoint (v2)	
https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-a75887f696cc/oauth2/v2.0/token	D
OAuth 2.0 authorization endpoint (v1)	
https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-a75887f696cc/oauth2/authorize	D
OAuth 2.0 token endpoint (v1)	
https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-a75887f696cc/oauth2/token	D
OpenID Connect metadata document	
https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-a75887f696cc/v2.0/.well-known/openid-configuration	D
Microsoft Graph API endpoint	
https://graph.microsoft.com	D
Federation metadata document	
https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-a75887f696cc/federationmetadata/2007-06/federationmetadata/	əta.xml 🗈
WS-Federation sign-on endpoint	
https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-a75887f696cc/wsfed	D
SAML-P sign-on endpoint	
https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-a75887f696cc/saml2	D

- 4. Open the following files in a text editor. You must run the editing application (for example, Notepad) as administrator.
 - C:\Program Files\Polaris\7.7\LeapWebApp\appsettings.user.json
 - C:\Program Files\Polaris\7.7\LeapWebApp\RELEASE-NOTES.md
- 5. In the RELEASE-NOTES.md file, copy the settings in the OIDC/OAuth Setup for AzureAD section. The image below shows the settings to copy.

```
### OIDC/OAuth Setup for AzureAD
  "OAuth": {
    "Authority": "https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-a75887f696cc/",
    "AuthorizationEndpoint": null,
    "TokenEndpoint": null,
    "UserInformationEndpoint": null,
    "ClientId": "efc04e56-0777-45a1-b7c5-793dbc1dbd68"
    "ClientSecret": "YLU8Q~TGdT_KIcDZ5px643zcsLmCzr16S1jiKcZI",
    "MetadataAddress": "https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-a75887f696cc/v2.0/.well-
known/openid-configuration",
    "KnownAuthorities": [ "login.microsoftonline.com" ],
    "CallbackPath": "/signin-oidc",
    "SignedOutCallbackPath": "/signout-callback-oidc",
    "SignedOutRedirectUri": "/login",
    "RemoteAuthenticationTimeout": 1,
    "RemoteFailureRedirectUri": "/leapwebapp/logout",
    "ResponseMode": "form_post",
"ResponseType": "code",
    "SaveTokens": false,
    "Scopes": [ "openid", "profile", "api://efc04e56-0777-45a1-b7c5-793dbc1dbd68/pas" ],
    "UseOIDC": true,
    "UsePkce": true,
    "AlternateUpnClaimType": "preferred_username",
    "AlternateLogoutUri": null,
    "OptionalAuthorizeParameters": null,
    "OptionalEndSessionParameters": null,
    "SendAccessTokenAsHeaderValue": false,
    "AccessTokenHeaderName": "
    "BasicAuthorizationCredentials": {
      "Username": null,
      "Password": null
    }
  },
```

- 6. In the .json file, replace the OAuth contents of the file with the settings you copied from the RELEASE-NOTES.md file.
- 7. On the Endpoints dialog, copy the root value and tenant ID from the **OAuth 2.0 authorization endpoint (v2)** box and paste it into the Authority property in the .json file. Your value will be similar to this one:

```
https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-
a75887f696cc/
```

Note:

The value must include the trailing slash character.

8. In the .json file, replace the ClientId property value with the application (client) ID you copied during the <u>Register LeapWebApp with Azure AD</u> step. Your value will be similar to this one:

efc04e56-0777-45a1-b7c5-793dbc1dbd68

9. In the .json file, replace the ClientSecret property value with the client secret you copied and saved during the <u>Create Client Credentials</u> step. Your value will be similar to this one:

4I.8Q~-GpkdoymMQneGIYNg40FRjx2Hr1wWLDcbr

10. On the Endpoints dialog, copy the value from the **OpenID Connect metadata document** box and paste it into the MetadataAddress property in the .json file. Your value will be similar to this one:

```
https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-
a75887f696cc/v2.0/.well-known/openid-configuration
```

11. On the Endpoints dialog, copy the FQDN of the Microsoft server from the **OpenID Connect metadata document** box and paste it into the KnownAuthorities property in the .json file. The value will be identical to this one:

```
login.microsoftonline.com
```

12. In the .json file, update the Scopes property to add the scope you copied and saved during the Expose the Polaris.ApplicationServices API step. Your value will be similar to this one:

api://efc04e56-0777-45a1-b7c5-793dbc1dbd68/pas

Note:

If you want Azure AD to return the user's email address, add "email" to the Scopes property. You might choose to do this if the preferred_username is a generic name, phone number, or other value.

13. In the .json file, set the AlternateUpnClaimType property to "preferred_ username".

Note:

You can also set this property to "email", if you choose.

- 14. Save the .json file.
- 15. Leave the browser tab displaying the Endpoints dialog open, and continue to the <u>Configure Polaris.ApplicationServices for Use with Azure AD</u> procedure.

Configure Polaris.ApplicationServices for Use with Azure AD

To configure Polaris.ApplicationServices, update C:\Program Files\Polaris\7.7\Polaris.ApplicationServices\appsettings.user.json using the following information:

- Endpoint URIs copied from the Azure portal
- Values that you copied and saved during earlier steps in the Azure AD configuration process

To configure Polaris.ApplicationServices

- 1. Open the following files in a text editor. You must run the editing application (for example, Notepad) as administrator.
 - C:\Program
 Files\Polaris\7.7\Polaris.ApplicationServices\appsettings.user.json
 - C:\Program Files\Polaris\7.7\Polaris.ApplicationServices\RELEASE-NOTES.md
- 2. In the RELEASE-NOTES.md file, find the **Azure AD** section and copy the settings in the **Example authority settings** section. The image below shows the settings to copy.

```
## Azure AD
Example authority settings object:
{
    "Name": "AzureAD",
    "Authority": "https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-a75887f696cc/oauth2/v2.0/", "Audience": "api://f9f429e3-8355-4eb3-876b-0e45b6ddd295",
    "MetaAddress": "https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-a75887f696cc/v2.0/.well-known/openid-
configuration",
    "RequireHttpsMetadata": true,
    "RequireSignedTokens": true,
    "ValidateIssuer": true,
    "ValidIssuers": [
         "https://sts.windows.net/3e8e440e-d3c1-459d-87a0-a75887f696cc/"
         "https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-a75887f696cc/v2.0"
    ],
"ValidateAudience": true,
         "f9f429e3-8355-4eb3-876b-0e45b6ddd295",
         "api://f9f429e3-8355-4eb3-876b-0e45b6ddd295"
    "JPNClaimTypes": [ "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn", "upn", "preferred_username" ],
"IsUPNExternalID": false,
    "OpaqueToken": false,
    "UserInformationEndpoint": null
```

- 3. In the .json file, replace the existing OAuth.Authorities property with the settings you copied from the RELEASE-NOTES.md file.
- 4. In the RELEASE-NOTES.md file, copy the settings in the **Example Swagger settings** section. The image below shows the settings to copy.

```
Example Swagger settings:

"Swagger": {
    "ClientID": "efc04e56-0777-45a1-b7c5-793dbc1dbd68",
    "ClientSecret": "",
    "AppName": "Polaris.ApplicationServices",
    "AuthorizationUrl": "https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-
a75887f696cc/oauth2/v2.0/authorize",
    "TokenUrl": "https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-a75887f696cc/oauth2/v2.0/token",
    "RefreshTokenUrl": "https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-a75887f696cc/oauth2/v2.0/token",
    "LogoutUrl": "https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-a75887f696cc/oauth2/v2.0/token",
    "Scopes": [
        { "Name": "openid", "Description": "Use OIDC to verify the user's identity" },
        { "Name": "email", "Description": "Optional to return user's email address" },
        { "Name": "api://efc04e56-0777-45a1-b7c5-793dbc1dbd68/pas", "Description": "API Scope defined in AzureAD
for LeapWebApp" }
    ]
}
```

- 5. In the .json file, replace the existing OAuth.Swagger property with the settings you copied from the RELEASE-NOTES.md file.
- 6. Make the following update to the Authority property:
 - a. On the Endpoints dialog of the Azure AD portal, copy the value from the **OAuth 2.0 authorization endpoint (v2)** box but omit the trailing *authorize*.
 - b. Paste this value into the Authority property in the .json file. Your value will be similar to this one:

```
https://login.microsoftonline.com/3e8e440e-d3c1-459d-
87a0-a75887f696cc/oauth2/v2.0/
```

- 7. In the .json file, update the Audience property:
 - a. Locate the application (client) ID you copied and saved during the <u>Register</u> <u>LeapWebApp with Azure AD</u> step.
 - b. Use it to construct a string with the following format:

api://[application (client) ID]

c. Paste this value into the Audience property. Your value will be similar to this one:

api://efc04e56-0777-45a1-b7c5-793dbc1dbd68

8. On the Endpoints dialog, copy the value from the **OpenID Connect metadata document** box and paste it into the MetaAddress property in the .json file. Your value will be similar to this one:

https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0a75887f696cc/oauth2/v2.0/well-known/openid-configuration

- 9. In the .json file, add a value to the ValidIssuers property:
 - a. On the Endpoints dialog, copy the tenant ID from the **OAuth 2.0 authorization** endpoint (v2) box.
 - b. Use it to construct a URI with the following format:

https://sts.windows.net/[tenant ID]

c. Paste this value into the ValidIssuers property. Your value will be similar to this one:

```
https://sts.windows.net/3e8e440e-d3c1-459d-87a0-
a75887f696cc/
```

- 10. In the .json file, add a second value to the ValidIssuers property:
 - a. On the Endpoints dialog, copy the value from the **OAuth 2.0 authorization endpoint (v2)** box but omit the trailing *authorize*.
 - b. Paste this value into the ValidIssuers property in the .json file. Your value will be similar to this one:

```
https://login.microsoftonline.com/3e8e440e-d3c1-459d-
87a0-a75887f696cc/oauth2/v2.0
```

- 11. In the .json file, add a value to the ValidAudiences property:
 - a. Locate the application (client) ID you copied and saved during the <u>Register</u> <u>LeapWebApp with Azure AD</u> step.
 - b. Paste this value into the ValidAudiences property. Your value will be

similar to this one:

```
efc04e56-0777-45a1-b7c5-793dbc1dbd68
```

- 12. In the .json file, add a second value to the ValidAudiences property:
 - a. Locate the application (client) ID you copied and saved during the <u>Register</u> <u>LeapWebApp with Azure AD</u> step.
 - b. Use it to construct a string with the following format:

api://[application (client) ID]

c. Paste this value into the ValidAudiences property. Your value will be similar to this one:

api://efc04e56-0777-45a1-b7c5-793dbc1dbd68

13. In the .json file, update the UPNClaimTypes property to add "upn" and "preferred_username" if those values are not already present. Your UPNClaimTypes property will be similar to this one:

"UPNClaimTypes": ["http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn", "preferred_username"],

- 14. In the Swagger property in the .json file, update the ClientID property:
 - a. Locate the application (client) ID you copied and saved during the <u>Register</u> LeapWebApp with Azure AD step.
 - b. Paste this value into the ClientID property. Your value will be similar to this one:

```
efc04e56-0777-45a1-b7c5-793dbc1dbd68
```

15. On the Endpoints dialog, copy the value from the OAuth 2.0 authorization endpoint (v2) box and paste it into the AuthorizationUrl property in the .json file. Your value will be similar to this one:

```
https://login.microsoftonline.com/3e8e440e-d3c1-459d-87a0-
a75887f696cc/oauth2/v2.0/authorize
```

16. In the .json file, update the TokenUrl property:

- a. On the Endpoints dialog, copy the value from the **OAuth 2.0 token endpoint** (v2) box but replace *authorize* with *token*.
- b. Paste the value into the TokenUrl property in the .json file. Your value will be similar to this one:

```
https://login.microsoftonline.com/3e8e440e-d3c1-459d-
87a0-a75887f696cc/oauth2/v2.0/token
```

- 17. In the .json file, update the RefreshTokenUrl property:
 - a. On the Endpoints dialog, copy the value from the **OAuth 2.0 token endpoint** (v2) box but replace *authorize* with *token*.
 - b. Paste the value into the RefreshTokenUrl property in the .json file. Your value will be similar to this one:

```
https://login.microsoftonline.com/3e8e440e-d3c1-459d-
87a0-a75887f696cc/oauth2/v2.0/token
```

- 18. In the .json file, update the LogoutUrl property:
 - a. On the Endpoints dialog, copy the value from the **OAuth 2.0 authorization endpoint (v2)** box but replace *authorize* with *logout*.
 - b. Paste the value into the LogoutUrl property in the .json file. Your value will be similar to this one:

```
https://login.microsoftonline.com/3e8e440e-d3c1-459d-
87a0-a75887f696cc/oauth2/v2.0/logout
```

- 19. In the .json file, add a new scope Name and Description:
 - a. Locate the scope you copied and saved during the <u>Expose the</u> <u>Polaris.ApplicationServices API step.</u>
 - b. Copy the scope and use it to construct a new Name property. Your value will be similar to this one:

"Name": "api://efc04e56-0777-45a1-b7c5-793dbc1dbd68/pas"

- c. Paste the Name property into the .json file.
- d. Add a Description property that matches the example below:

```
"Description": "API Scope defined in AzureAD for LeapWebApp"
```

20. Save the .json file. Your updated values in the Swagger property should look similar to the example below.



Note:

Changes to the appsettings.user.json files do note take effect until the IIS application pools are restarted or IIS is reset.

Configuring Basic OAuth 2.0

To configure basic OAuth 2.0, perform the following tasks:

- 1. <u>Set up web services and applications</u>.
- 2. Add authentication redirect URIs.

After you complete these tasks, <u>Add a URL rewrite rule for LeapWebApp</u>.

Set Up Web Services and Applications for Basic OAuth 2.0

To set up each of the following web services and applications, you must configure a .json file for each of the following:

- Polaris.Authentication (the application that authenticates Polaris users)
- Polaris.AuthenticationServices (the API service that provides backend support for authentication)
- PolarisAdmin (the web-based Polaris System Administration application)
- Polaris.ApplicationServices (Leap's API service)

Note:

Since basic OAuth 2.0 doesn't support reauthentication or permission overrides, the only configuration needed for the C:\Program Files\Polaris\7.7\LeapWebApp\appsettings.user.json file are the changes described in the Upgrading to Polaris 7.7 section.

The four .json files are all named appsettings.user.json, but they reside in different directories:

- C:\Program Files\Polaris\7.7\Polaris.Authentication
- C:\Program Files\Polaris\7.7\Polaris.AuthenticationServices
- C:\Program Files\Polaris\7.7\PolarisAdmin\assets
- C:\Program Files\Polaris\7.7\Polaris.ApplicationServices

This section contains the following topics:

- Configure Polaris.Authentication for Use with Basic OAuth 2.0
- Configure Polaris.AuthenticationServices for Use with Basic OAuth 2.0
- Configure PolarisAdmin for Use with Basic OAuth 2.0
- Configure Polaris.ApplicationServices for Use with Basic OAuth 2.0

Configure Polaris. Authentication for Use with Basic OAuth 2.0

To configure Polaris.Authentication, update C:\Program Files\Polaris\7.7\Polaris.Authentication\appsettings.user.json. You will use several values copied from your identity provider.

Important:

By default, the appsettings.user.json template file contains configuration settings that apply to AD FS. Polaris 7.7 includes a RELEASE-NOTES.md file that contains the template settings that apply to basic OAuth 2.0. See the Configuring Basic OAuth 2.0 section of the release notes file.

To configure Polaris.Authentication

- 1. Open the following files in a text editor. You must run the editing application (for example, Notepad) as administrator.
 - C:\Program Files\Polaris\7.7\Polaris.Authentication\appsettings.user.json
 - C:\Program Files\Polaris\7.7\Polaris.Authentication\RELEASE-NOTES.md
- 2. In the RELEASE-NOTES.md file, copy the settings in the Configuring Basic OAuth 2.0 section. The image below shows the settings to copy.

```
## Configuring Basic OAuth 2.0
### AWS Cognito Template
"OAuth": {
  "Enabled": true,
  "SendOAuthAuthorityHeader": false,
  "Authorities": [
    {
      "IsActive": true,
      "Name": "AmazonCognito",
      "UseOidc": false,
      "AuthorizationEndpoint": "[COGNITO-DOMAIN]/oauth2/authorize",
      "TokenEndpoint": "[COGNITO-DOMAIN]/oauth2/token",
      "UserInformationEndpoint": "[COGNITO-DOMAIN]/oauth2/userInfo",
      "EndSessionEndpoint": "[COGNITO-DOMAIN]/logout_logout_uri=https%3A%2F%2F[POLARIS-BASE-URL]%2Fpolarisauth%
2Flogin&client_id=[CLIENT-ID]",
      "ClientId": "[CLIENT-ID]",
"ClientSecret": "[CLIENT-SECRET]",
"CallbackPath": "/oauth/callback",
      "UsePkce": false,
      "Scopes": [
         "profile",
         "openid"
       "SaveTokens": false,
       "SendAccessTokenAsHeaderValue": false,
      "AccessTokenHeaderName": ""
       "BasicAuthorizationCredentials": {
        "Username": ""
        "Password": "",
       "OptionalAuthorizeParameters": null
    }
 1
```

- 3. In the .json file, replace the entire contents of the file with the settings you copied from the RELEASE-NOTES.md file.
- 4. Verify that Enabled is set to true.
- 5. Verify that IsActive is set to true.
- 6. In the Name property, replace AmazonCognito with a unique name that describes your OAuth identity provider (for example, Azure AD or AD FS).
- 7. Verify that UseOidc is set to false.
- 8. Update the following properties. In each, replace [COGNITO-DOMAIN] with the server address for your identity provider:
 - AuthorizationEndpoint
 - TokenEndpoint

- UserInformationEndpoint
- EndSessionEndpoint
- 9. In the EndSessionEndpoint property, replace [POLARIS-BASE-URL] with the FQDN of the server that hosts the Authentication Application.
- 10. In the ClientId property, replace [CLIENT-ID] with the client ID from your identity provider.
- 11. In the ClientSecret property, replace [CLIENT-SECRET] with the client secret from your identity provider.
- 12. Save the .json file.

Configure Polaris. AuthenticationServices for Use with Basic OAuth 2.0

To configure Polaris.AuthenticationServices, update C:\Program Files\Polaris\7.7\Polaris.AuthenticationServices\appsettings.user.json. You will use several values copied from your identity provider.

Important:

By default, the appsettings.user.json template file contains configuration settings that apply to AD FS. Polaris 7.7 includes a RELEASE-NOTES.md file that contains the template settings that apply to basic OAuth 2.0. See the OAuth2 Only Setup for AmazonCognito section of the release notes file.

To configure Polaris.AuthenticationServices

- 1. Open the following files in a text editor. You must run the editing application (for example, Notepad) as administrator.
 - C:\Program Files\Polaris\7.7\Polaris.AuthenticationServices\appsettings.user.json
 - C:\Program Files\Polaris\7.7\Polaris.AuthenticationServices\RELEASE-NOTES.md
- 2. In the RELEASE-NOTES.md file, copy the settings in the OAuth2 Only Setup for AmazonCognito section. The image below shows the settings to copy.

```
### OAuth2 Only Setup for AmazonCognito
"OAuth": {
  "Enabled": true,
  "Authorities": [
   {
     "Name": "AmazonCognito",
     "UseOidc": false,
      "RequireHttpsMetadata": false,
      "RequireSignedTokens": false,
      "ValidateIssuer": false,
      "Authority": null,
      "Audience": "4k6l3mtmifukol3bc9a8mjkb8r",
      "MetaAddress": null,
      "UserInformationEndpoint": "https://polaris-leap.auth.us-east-1.amazoncognito.com/oauth2/userInfo",
      "ValidIssuers": [
        "https://polaris-leap.auth.us-east-1.amazoncognito.com/"
       "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_zNpQPq701"
      ],
      "ValidateAudience": false,
      "ValidAudiences": [ "4k613mtmifuko13bc9a8mjkb8r" ],
      "UpnClaimTypes": [ "sub" ],
      "IsUpnExternalId": true,
      "SendAccessTokenAsHeaderValue": false,
      "AccessTokenHeaderName": ""
      "BasicAuthorizationCredentials": {
        "Username": "",
"Password": ""
     }
```

- 3. In the .json file, replace the entire contents of the file with the settings you copied from the RELEASE-NOTES.md file.
- 4. Verify that Enabled is set to true.
- 5. In the Name property, replace *AmazonCognito* with a unique name that describes your OAuth identity provider (for example, Azure AD or AD FS).
- 6. Verify that UseOidc is set to false.
- 7. In the UserInformationEndpoint property, replace *polarisleap.auth.us-east-1.amazoncognito.com* with the server address for your identity provider.
- 8. In the ValidIssuers property, replace the default URLs with the valid issuer URLs from your identity provider.
- 9. In the UpnClaimTypes property, update the default value if you want to specify a different claim to serve as the user identifier. The default value applies to most configurations, but you can specify a different claim that exists in a JSON Web

Token (JWT).

10. Save the .json file.

Configure PolarisAdmin for Use with Azure AD

To configure PolarisAdmin, update C:\Program Files\Polaris\7.7\PolarisAdmin\assets\appsettings.user.json.

To configure PolarisAdmin

- Open the C:\Program
 Files\Polaris\7.7\PolarisAdmin\assets\appsettings.user.json file in a text editor.
 You must run the editing application (for example, Notepad) as administrator.
- 2. In the .json file, update the server location in the apiUrlRoot. If you started from the template settings provided in the RELEASE-NOTES.md file, replace [my-server-domain-name] with the FQDN of the server that hosts both the API service for Polaris System Administration (web-based) and the Authentication Application.
- 3. Update the server location in the authAppUrl. If you started from the template settings provided in the RELEASE-NOTES.md file, replace [my-server-domain-name] with the FQDN of the server that hosts both the API service for Polaris System Administration (web-based) and the Authentication Application.
- 4. Save the .json file.

Configure Polaris. Application Services for Use with Basic OAuth 2.0

To configure Polaris.ApplicationServices, update C:\Program Files\Polaris\7.7\Polaris.ApplicationServices\appsettings.user.json. You will use several values copied from your identity provider.

Important:

By default, the appsettings.user.json template file contains configuration settings that apply to AD FS. Polaris7.7 includes a RELEASE-NOTES.md file that contains the template settings that apply to basic OAuth 2.0. See the **Example OAuth2 with Opaque Tokens and /userinfo endpoint** section of the release notes file.

To configure Polaris.ApplicationServices

- 1. Open the following files in a text editor. You must run the editing application (for example, Notepad) as administrator.
 - C:\Program Files\Polaris\7.7\Polaris.ApplicationServices\appsettings.user.json
 - C:\Program Files\Polaris\7.7\Polaris.ApplicationServices\RELEASE-NOTES.md
- 2. In the RELEASE-NOTES.md file, copy the settings in the **Example OAuth2 with Opaque Tokens and /userinfo endpoint** section. The image below shows the settings to copy.

```
## Example OAuth2 with Opaque Tokens and /userinfo endpoint
"OAuth": {
      "Enabled": true,
      "Authorities": [
          "Name": "OAuth2MockService",
          "Authority": null,
          "Audience": null,
          "MetaAddress": null,
          "RequireHttpsMetadata": false,
          "RequireSignedTokens": false,
          "ValidateIssuer": false,
          "ValidIssuers": [
            "http://localhost:8080/"
          ],
          "ValidateAudience": false,
          "ValidAudiences": [],
          "UPNClaimTypes": [ "sub" ],
          "IsUPNExternalID": true,
          "OpaqueToken": true,
          "UserInformationEndpoint": "http://localhost:8080/userinfo",
          "SendAccessTokenAsHeaderValue": false,
          "BasicAuthorizationCredentials": {
            "Username": null,
            "Password": null
         }
       }
      ],
       'Swagger": {
        "ClientID": "7fbe747f-1101-451c-a947-fa6ef898a517",
        "ClientSecret": "",
        "AppName": "Polaris.ApplicationServices",
        "AuthorizationUrl": "http://localhost:8080/authorize",
        "TokenUrl": "http://localhost:8080/token",
        "RefreshTokenUrl": "http://localhost:8080/token",
        "LogoutUrl": "http://localhost:8080/endsession",
        "Scopes": [
          {
            "Name": "profile",
            "Description": "Return basic profile information."
          }
        ]
      }
```

3. In the .json file, replace the entire contents of the file with the settings you copied from the RELEASE-NOTES.md file.

- 4. In the Authorities property, update the following properties with information from your identity provider:
 - a. In the Name property, replace OAuth2MockService with a unique name that describes your OAuth identity provider (for example, Azure AD or AD FS).
 - b. Update the ValidIssuers property:
 - i. Copy the tenant ID from your identity provider.
 - ii. Use it to construct a URI with the following format:

https://sts.windows.net/[tenant ID]

iii. Paste this value into the ValidIssuers property. Your value will be similar to this one:

```
https://sts.windows.net/3e8e440e-d3c1-459d-87a0-
a75887f696cc/
```

- c. Update the UserInformationEndpoint property with the user information endpoint from your identity provider.
- 5. In the Swagger property, update the following properties with values copied from your identity provider:
 - a. In the ClientID property, replace 7fbe747f-1101-451c-a947fa6ef898a517 with the client ID for your configuration.
 - b. In the ClientSecret property, enter the client secret from your identity provider.
 - c. In the following properties, replace *localhost:8080* with the server address from your identity provider:
 - AuthorizationUrl
 - TokenUrl
 - RefreshTokenUrl
 - LogoutUrl
- 6. Save the .json file.

Add Authentication Redirect URIs

Add the following URIs to your identity provider's hosted UI configuration:

- Allowed callback URIs:
 - https://[FQDN]/leapwebapp/login
 - https://[FQDN]/leapwebapp/oauth/callback
 - https://[FQDN]/polarisauth/login
 - https://[FQDN]/polarisauth/oauth/callback
- Allowed sign-out URIs:
 - https://[FQDN]/leapwebapp/login
 - https://[FQDN]/leapwebapp/logout
 - https://[FQDN]/polarisauth/login
 - https://[FQDN]/polarisauth/logout

Notes:

- Replace [FQDN] with the fully-qualified domain name of your server.
- Example: https://leap.mylibrary.org/leapwebapp/login

Configuring Basic Authentication

To configure basic authentication, perform the following task:

• Set Up Web Services and Applications for Basic Authentication.

After you complete this task, Add a URL rewrite rule for LeapWebApp.

Set Up Web Services and Applications for Basic Authentication

To set up basic authentication, you must configure a .json file for each of the following:

- Polaris.Authentication (the application that authenticates Polaris users)
- Polaris.AuthenticationServices (the API service that provides backend support for authentication)

Note:

When setting up basic authentication, the only configuration needed for the C:\Program Files\Polaris\7.7\LeapWebApp\appsettings.user.json file are the changes described in the Upgrading to Polaris 7.7 section.

The two .json files are both named appsettings.user.json, but they reside in different directories:

- C:\Program Files\Polaris\7.7\Polaris.Authentication
- C:\Program Files\Polaris\7.7\Polaris.AuthenticationServices

This section contains the following topics:

- Configure Polaris. Authentication for Use with Basic Authentication
- <u>Configure Polaris.AuthenticationServices for Use with Basic Authentication</u>

Configure Polaris. Authentication for Use with Basic Authentication

To configure Polaris.Authentication, update C:\Program Files\Polaris\7.7\Polaris.Authentication\appsettings.user.json.

To configure Polaris.Authentication
1. Open the C:\Program

Files\Polaris\7.7\Polaris.Authentication\appsettings.user.json file in a text editor. You must run the editing application (for example, Notepad) as administrator.

- 2. In the BasicAuth property, verify that the Enabled property is set to true.
- 3. In the OAuth property, set the Enabled property to false.
- 4. Save the .json file.

Configure Polaris. Authentication Services for Use with Basic Authentication

To configure Polaris.AuthenticationServices, update C:\Program Files\Polaris\7.7\Polaris.AuthenticationServices appsettings.user.json.

To configure Polaris.AuthenticationServices

- Open the C:\Program
 Files\Polaris\7.7\Polaris.AuthenticationServices\appsettings.user.json file in a
 text editor. You must run the editing application (for example, Notepad) as
 administrator.
- 2. In the BasicAuth property, set the Enabled property to true.
- 3. In the OAuth property, set the Enabled property to false.
- 4. Save the .json file.

Add a URL Rewrite Rule for LeapWebApp

Adding a URL rewrite rule redirects incoming URLs to the correct address for the LeapWebApp. This must be done manually, since the library may already use other URL rewrite rules.

To add a URL rewrite rule, you must have the Microsoft IIS URL Rewrite 2.1 extension. For more information, see <u>https://www.iis.net/downloads/microsoft/url-rewrite</u>.

To add a URL rewrite rule

1. Open the root IIS web.config file, found in the following location:

C:\inetpub\wwwroot\web.config

2. Add a rewrite rule to the **system.webServer** node.



Note:

For sample rewrite rule text that you can copy and paste, see Sample Rewrite Rule Text.

In the example above, if the incoming URL includes a path that contains any of the following, the rewrite rule redirects to /leapwebapp:

- /LeapWebApp
- /Leapwebapp
- /LEAPWEBAPP
- 3. Save the web.config file.

Note:

When registering redirect URIs for LeapWebApp in AD FS, the URIs should

be lowercase. For example:

- https://rd-polaris.polarislibrary.com/leapwebapp/signin-oidc
- https://rd-polaris.polarislibrary.com/leapwebapp/signin-overrideoidc
- https://rd-polaris.polarislibrary.com/leapwebapp/signout-callbackoidc

Sample Rewrite Rule Text

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <system.webServer>
     <rewrite>
        <rules>
          <rule name="UrlToLowercase" stopProcessing="true">
             <match url="(.*)" ignoreCase="true" />
               <action type="Redirect" url="https://{HTTP HOST}</pre>
                {ToLower:{PATH_INFO}}" redirectType="Found"
               appendQueryString="true" />
               <conditions>
                  <add input="{PATH INFO}" pattern="^/LeapWebApp
                  (.*) | ^/Leapwebapp(.*) | ^/LEAPWEBAPP(.*) "
                  ignoreCase="false" />
               </conditions>
          </rule>
        </rules>
     </rewrite>
  </system.webServer>
</configuration>
```

Additional URL Rewrite Resources

See Microsoft's URL Rewrite Module Configuration Reference for additional information:

<u>https://docs.microsoft.com/en-us/iis/extensions/url-rewrite-module/url-rewrite-module-configuration-reference</u>