MyID MFA and PSM

Version 5.3.2

Federation with Microsoft 365

Lutterworth Hall, St Mary's Road, Lutterworth, Leicestershire, LE17 4PS, UK
www.intercede.com | info@intercede.com | @intercedemyid | +44 (0)1455 558111

Document reference: IMP2058-5.3.2

November 2025

Copyright

Conventions used in this document

- Lists:
  - Numbered lists are used to show the steps involved in completing a task when the order is important.
  - Bulleted lists are used when the order is unimportant or to show alternatives.
- **Bold** is used for menu items and for labels.

  For example:
  - Record a valid email address in **'From' email address**.
  - Select **Save** from the **File** menu.
- *Italic* is used for emphasis:

  For example:
  - Copy the file *before* starting the installation.
  - Do *not* remove the files before you have backed them up.
- ***Bold and italic*** hyperlinks are used to identify the titles of other documents.

  For example: "See the ***Release Notes*** for further information."

  Unless otherwise explicitly stated, all referenced documentation is available on the product installation media.
- A `fixed width` font is used where the identification of spaces is important, including filenames, example SQL queries and any entries made directly into configuration files or the database.
- **Notes** are used to provide further information, including any prerequisites or configuration additional to the standard specifications.

  For example:

  Note: This issue only occurs if updating from a previous version.
- Warnings are used to indicate where failure to follow a particular instruction may result in either loss of data or the need to manually configure elements of the system.

  For example:

  Warning: You must take a backup of your database before making any changes to it.

Contents

1 Introduction

Note: MyID MFA and MyID PSM were previously known as Authlogics products. Authlogics is now an Intercede Group company and the products have been rebranded accordingly. The term 'Authlogics' may still appear in certain areas of the product.

Microsoft supports federated access to Microsoft 365 resources through SAML 2.0 and WS-Fed federation protocols. This document details the steps required to configure MyID MFA 5.0 federation with Microsoft 365 using SAML 2.0.

MyID MFA natively supports multi-domain SAML 2.0, and does not require ADFS for integration.

2 Prerequisites

This document does not detail how to set up a hybrid environment with Microsoft 365 or Entra ID; this must already be in place. Specifically, you must have already set up the following:

- A Microsoft 365 tenant in "Managed" state (that is, not currently federated).
- The `Microsoft.Graph` PowerShell module.

  See section *2.1*, *PowerShell*.
- Directory synchronization with Microsoft Entra Connect (Azure ID Connect), or other management method for the Entra Immutable ID.

  For details of setting the Entra Immutable ID without a directory, see section *6*, *Federation without directory synchronization*.
- Entra ID admin logon.
- A deployed MyID MFA Server, with:
  - MFA users configured and tested (for example, using the Self Service Portal).
  - Public DNS entry for the IdP.
  - Public SSL certificate configured on the MFA server matching the DNS entry.

    For more information, see section *2.6*, *DNS and SSL*.
  - Inbound SSL access to the MyID MFA server from the Internet.

You must also carry out the following:

- Verify your current federation configuration.

  See section *2.2*, *Verify the current federation configuration*.
- You may want to remove an existing federation configuration.

  See section *2.3*, *Removing an existing federation configuration*.
- Check that you are not trying to federate the Entra default domain.

  See section *2.4*, *Verify the Entra default domain*.

You may want to:

- Switch from WS-Fed to SAML2 federation protocols.

  See section *2.5*, *Switching from WS-Fed to SAML2*

2.1 PowerShell

You require the `Microsoft Graph` PowerShell module for steps in this document.

To check if you already have this module installed, run the following command at an administrator PowerShell prompt.

```
PS C:\> Get-Module -Name Microsoft.Graph -ListAvailable
```

If you do not have the module installed, to install the module, run the following commands at an administrator PowerShell prompt.

```
PS C:\> Install-Module Microsoft.Graph -Scope AllUsers -Repository
PSGallery -Force
```

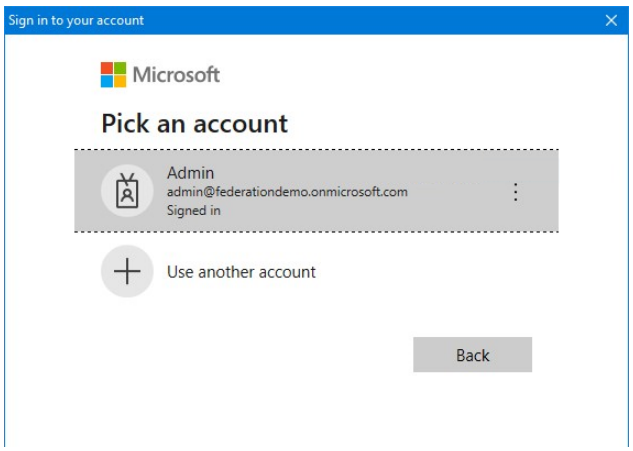
```
PS C:\> Import-Module Microsoft.Graph.Identity.DirectoryManagement
```

To execute a Microsoft Graph PowerShell command, you must authenticate with Entra. You are recommended to use an Entra ID administrator account, not a hybrid account, while configuring federation settings.

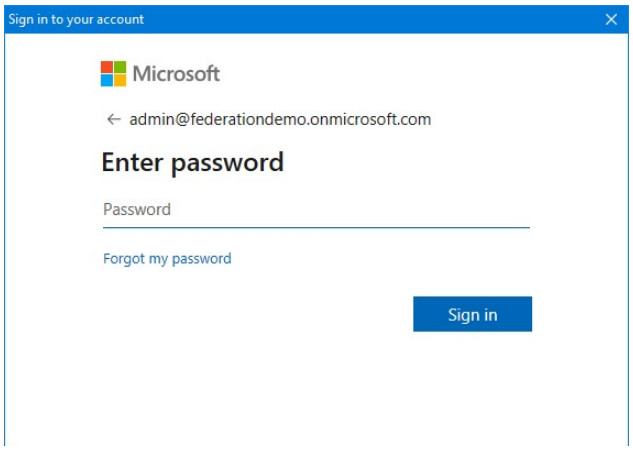
1. Run the following command:

   ```
   PS C:\> Connect-MgGraph -Scopes
   'User.ReadWrite.All,Group.ReadWrite.All,Directory.ReadWrite.All,Directo
   ry.AccessAsUSer.All'
   ```

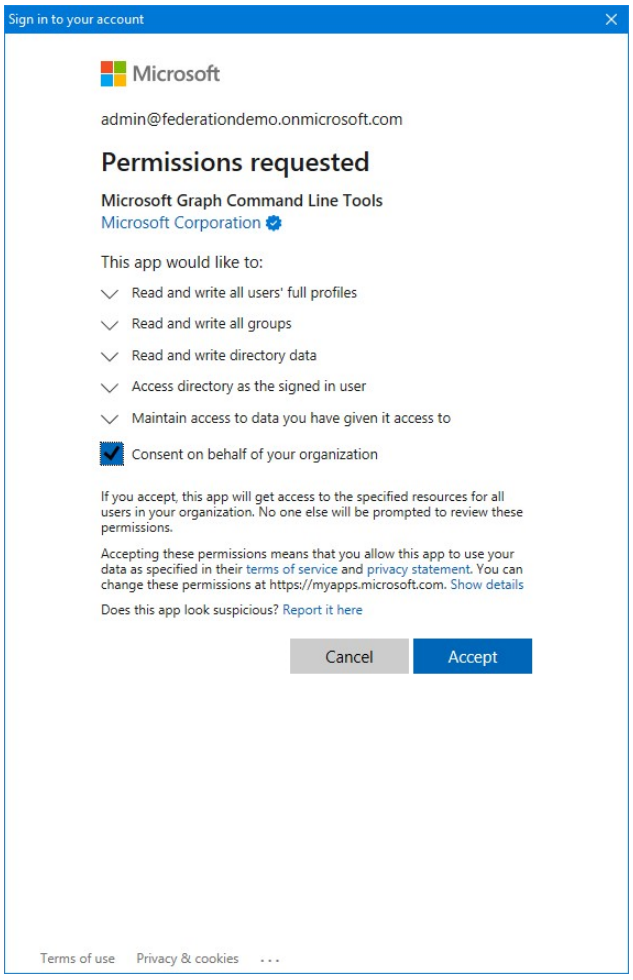
   You are prompted to authenticate using the appropriate method configured on the Entra account:



2. Provide your authentication details and sign in.



3. Select the **Consent on behalf of your organization** option:



2.2 Verify the current federation configuration

Ensure that the Office tenant is not already set up to use another federation server. Connect to MS Online and check the domain status is `Verified` and `Managed` by running the following commands:

```
PS C:\> Get-MgDomain | select Id, IsVerified, AuthenticationType


Id                                  IsVerified AuthenticationType
--                                  ---------- ------------------
federationdemo.onmicrosoft.com            True Managed
federationdemo.com                        True Federated
federationdemo.mail.onmicrosoft.com       True Managed
```

2.3 Removing an existing federation configuration

To remove an existing federation configuration and set the authentication in Entra back to `Managed`, run the following command:

```
PS C:\> Update-MgDomain -DomainId 'federationdemo.com' -AuthenticationType
'Managed'
```

Note: This may take up to two hours to take effect fully in Entra ID, even if the PowerShell commands show that the configuration has been changed.

2.4 Verify the Entra default domain

Microsoft Entra does not allow the default domain to be federated. To verify that the domain you want to federate with is *not* the default domain, run the following command:

```
PS C:\> Get-MgDomain | select Id, IsDefault

Id                                  IsDefault
--                                  ---------
federationdemo.onmicrosoft.com           True
federationdemo.com                      False
federationdemo.mail.onmicrosoft.com     False
```

If the domain with which you want to federate has the `IsDefault` value of `True`, you can set the `xxx.onmicrosoft.com` (or another) domain as the default by running the following command:

```
PS C:\> Update-MgDomain -DomainId 'federationdemo.onmicrosoft.com' -
IsDefault
```

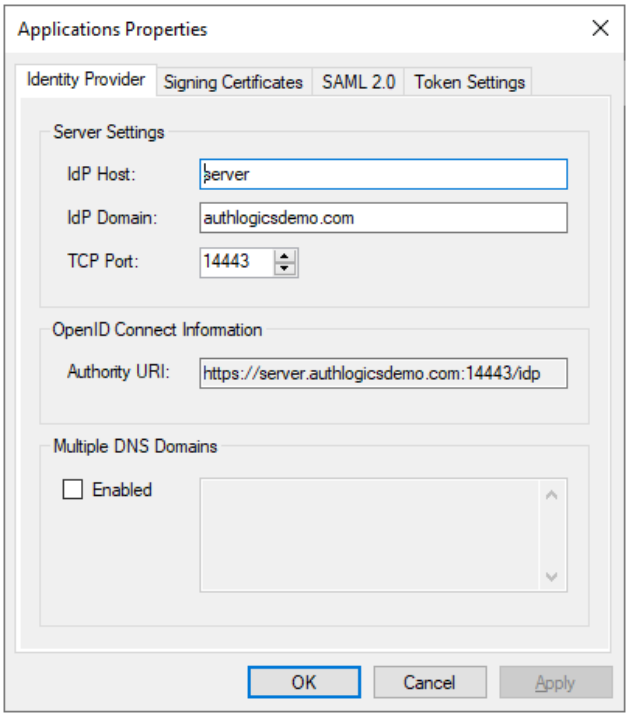
2.5 Switching from WS-Fed to SAML2

Microsoft 365 supports both SAML2 and WS-Fed federation protocols. To change from WS-Fed to SAML2 you must first disable federation for the domain by making it "Managed" (see section *2.3*, *Removing an existing federation configuration*) and then create a new SAML2 configuration following the instructions in this guide.

2.6 DNS and SSL

The MyID MFA server requires a publicly trusted SSL certificate. The DNS name in the SSL certificate must match the MyID MFA-configured IdP Host and Domain configuration; for example:

```
idp.federationdemo.com
```

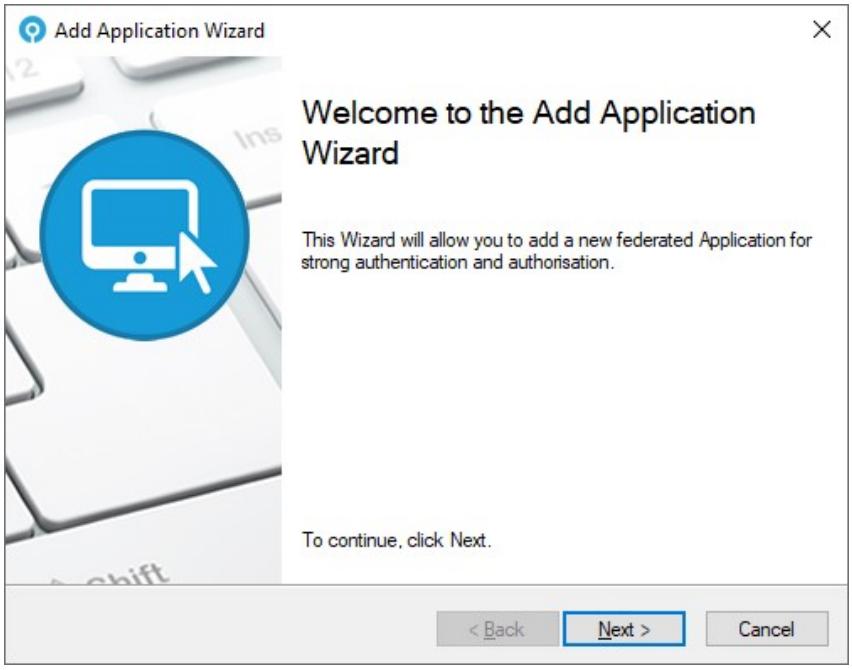


The DNS A or CNAME record must resolve to the MyID MFA server through any firewalls or load balancers. Firewalls must allow TCP port 443 from the Internet to the MyID MFA server.

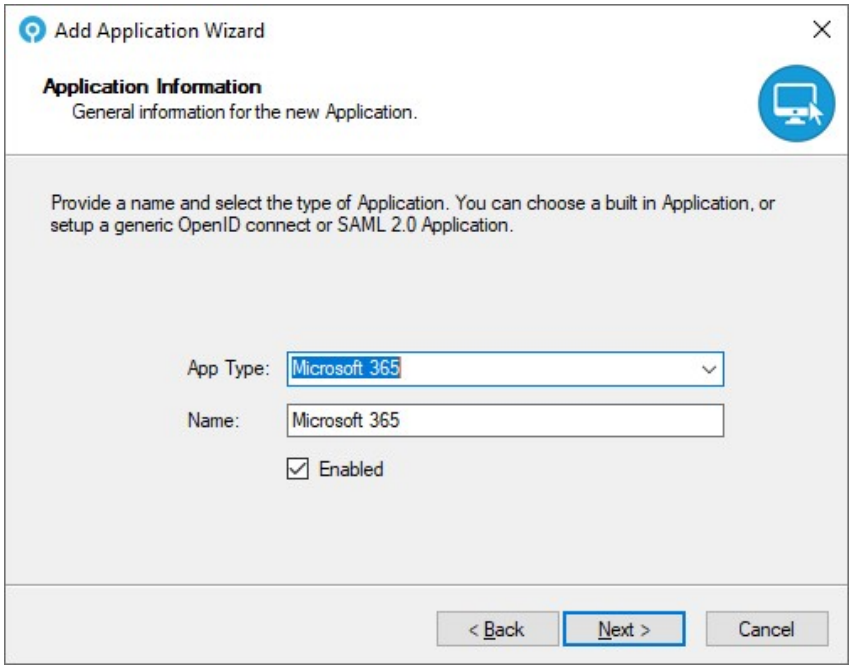
3 Adding the Microsoft 365 application

Open the MyID MMC to add an Application.

1. Start the Add Application Wizard.

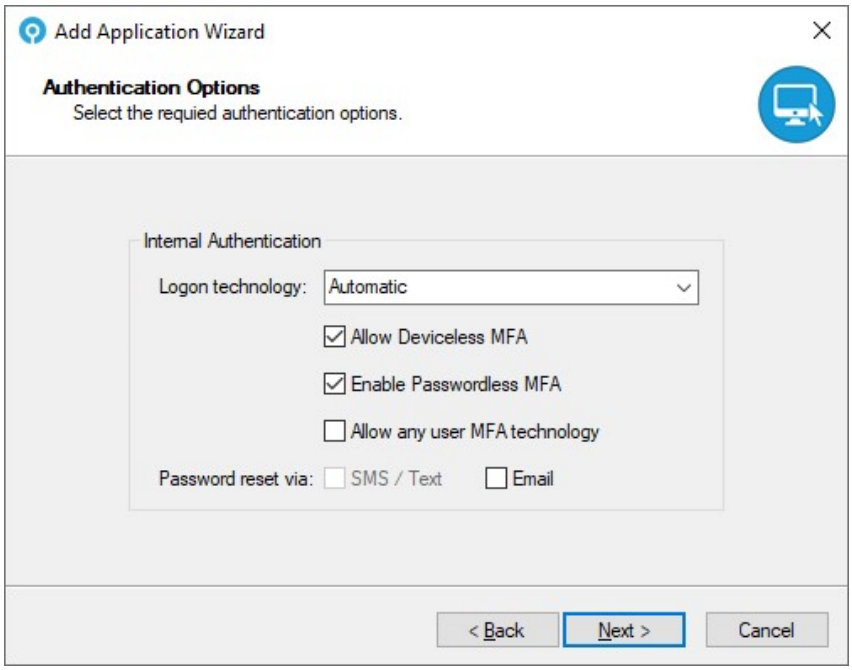


2. Click **Next**.

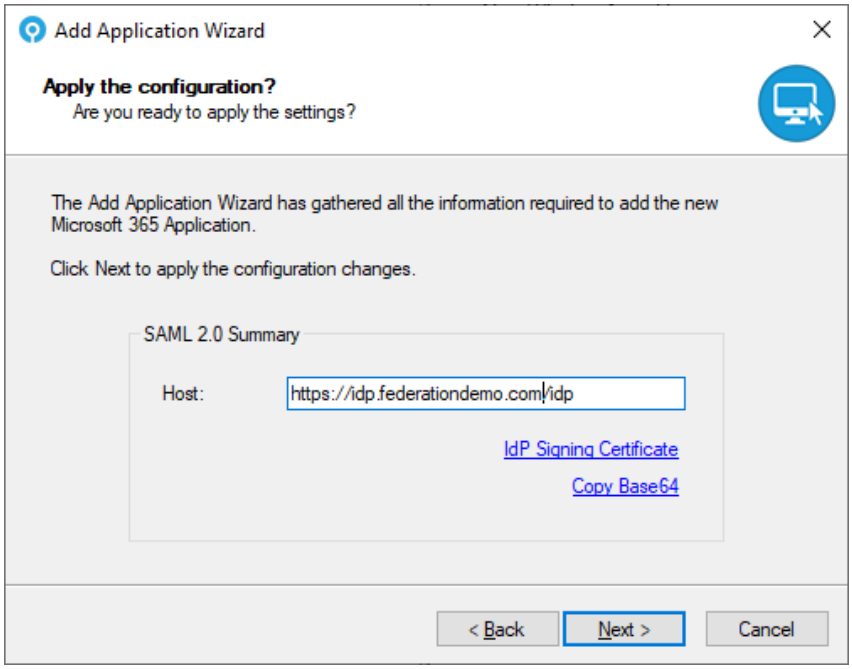


3. Select Microsoft 365 from the list and enter a custom name if required.

4. Click **Next**.

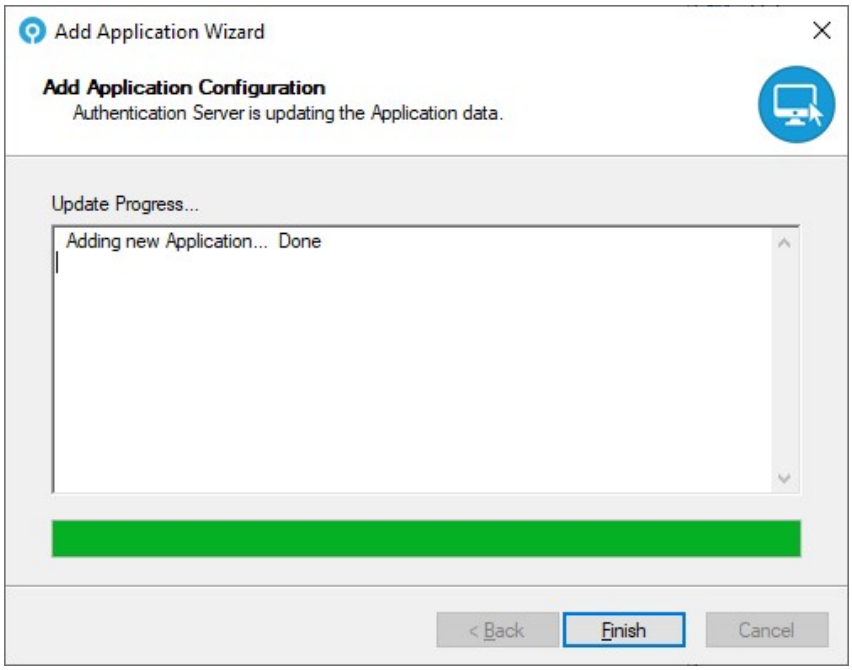


5. Select the required logon technology and authentication options.
6. Click **Next**.



7. Confirm the Host configuration information.
8. Click **Copy Base64** to copy the Base64 signing certificate information to the clipboard.

9. Click **Next**.



10. Click **Finish**.

The MyID side of the Microsoft 365 configuration is now complete.

4 Configuring federation

You perform the Microsoft 365 side of the configuration using PowerShell commands. To simplify the process, configure a custom PowerShell script with the settings for your environment. When you run the PowerShell script, it configures Microsoft 365 to use the MyID MFA Server for federated authentication.

When configuring single domain federation:

- Enable single domain federation using PowerShell.

  See section *4.1*, *Enable single domain federation using PowerShell*.
- Verify the configuration.

  See section *4.2*, *Verify the configuration*.
- Test the federation setup.

  See section *4.3*, *Test the federation setup*.

4.1 Enable single domain federation using PowerShell

Configure the yellow highlighted variables in the following sample script with the values of your domain and MyID MFA server.

- `$domain` – The DNS domain to federate, which must already be set up in Microsoft 365.
- `$display` – The friendly name shown to users when signing in to Microsoft 365. You are recommended to use something that is familiar, like your company name.
- `$issuerUri` – The **Issuer URI** from the **SAML 2.0** tab in the Applications Properties dialog. The default value will be a derivative of your IdP host and domain name; for example:

  `'urn:uri:ipdfederationdemocom'`
- `$signinUrl` – The SAML 2.0 IdP sign-on page URL. The path is fixed; however, the DNS name must match the public DNS name of the IdP.
- `$signoutUrl` – The SAML 2.0 IdP logout page URL. The path is fixed; however, the DNS name must match the public DNS name of the IdP.
- `$certificate` – The Base64 representation of the IdP signing certificate. You can obtain this value by clicking the copy icon on the **Signing Certificates** tab of the Applications Properties dialog or by clicking the **Copy Base64** link at the end of the Add Application Wizard.

1. Copy the text from the sample below to a new plain text document and save it as a `.ps1` file.
2. Configure the yellow highlighted variables in the following sample script with the values of your domain and MyID MFA server.
3. Run the script at a PowerShell command prompt to apply the configuration.

4.1.1 Sample single domain PowerShell script

```
$domain = 'federationdemo.com'
$display = 'Federation Demo'
$issuerUri = 'urn:uri:idpfederationdemocom'
$signinUrl = 'https://idp.federationdemo.com/idp/SAML/SingleSignOnService'
$signoutUrl = 'https://idp.federationdemo.com/idp/SAML/SingleLogoutService'
$certificate =
'MIIDGDCCAgCgAwIBAgIQFaTIA1mLiLtJ+wsZt9M2ejANBgkqhkiG9w0BAQsFADAf
MR0wGwYDVQQDDBQqLmZlZGVyYXRpb25kZW1vLmNvbTAeFw0yNDAzMDUxNDI0NTZa
Fw0zNDAzMDUxNDM0NTVaMB8xHTAbBgNVBAMMFCouZmVkZXJhdGlvbmRlbW8uY29t
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAnyjM01KPv3y4DUiKYpTH
DT9Gi4c/EGOU6bs8jh0Mke8TjTVWuHGiD98Mj4qLbb/yhk4LHemt58gtjxdj9+pj
gG38OU3dF0n7RMXES6EwK4KlsO16nrXEG6YtP6ElelJPkCNXzzSeoeHPCTSMxp1gF
mY/z8fOyI//x/8AmRI2JfGr43exXCbMjYx4sgr85HOCVdw27uHEK9w0hAPPHt2vq
7BMDAfYj2IisbpVekasJDmXtyhvRFptESJ80qvmmyTLD85iHmO7aME1/7vYnl1RQ
CqbZbhtrWYl4VBAiy/ySnqJdcaJT3KCOVJZOKZxurjXXNJbTHe8i3sQZ0dP2poJZ
tQIDAQABo1AwTjAOBgNVHQ8BAf8EBAMCBaAwHQYDVR0lBBYwFAYIKwYBBQUHAwEG
CCsGAQUFBwMCMB0GA1UdDgQWBBR2d84eaCTxgzIeXnY41uMia8DkJDANBgkqhkiG
9w0BAQsFAAOCAQEACOEinC4t1V80Kgs9MXu843e0UqLseOkoClNZbhxM4n3Y9cTP
b9QYQLQ69g8Q2d6tG+DzTCAnJeTdM2A9QWpePNuGceSWlFHHXHv/ZuzixA2SS2mn
AVvs9GgP1W/l1anMD1mhd4p9F+U0E/KMnn8yo2pYGI/wlwYm0yW3uaDdAQ1WS+fZ
ev2n5WcDbQ6WGklOL5jOJPvkiXcXmzhPc1ogsCvsWCL9OGhFxy3buLTp1N3Rk4dj
Z2hWyoeU8WjYax2436rfQx2qYJvgtAD4MDAzl95N28kzGBWr+eOO460NzDJ2OGc0
rrIZUyo19Uqjje3lNPPyVAzGp+cyrqeRQWpMjg=='
New-MgDomainFederationConfiguration -DomainId $domain -DisplayName $display
-IssuerUri $issuerUri -PassiveSignInUri $signinUrl -SignOutUri $signoutUrl
-SigningCertificate $certificate -PreferredAuthenticationProtocol saml -
federatedIdpMfaBehavior enforceMfaByFederatedIdp | Format-List
```

4.1.2 Single domain sample output

```
ActiveSignInUri                          :
DisplayName                              : Federation Demo
FederatedIdpMfaBehavior                  : enforceMfaByFederatedIdp
Id                                       : 523dd120-113b-480a-af4d-
853fdee26510
IsSignedAuthenticationRequestRequired :
IssuerUri                                : urn:uri:idpfederationdemocom
MetadataExchangeUri                      :
NextSigningCertificate                   :
PassiveSignInUri                         :
https://idp.federationdemo.com/idp/SAML/SingleSignOnService
PreferredAuthenticationProtocol          : saml
PromptLoginBehavior                      :
SignOutUri                               :
https://idp.federationdemo.com/idp/SAML/SingleLogoutService
SigningCertificate                       :
MIIDGDCCAgCgAwIBAgIQFaTIA1mLiLtJ+wsZt9M2ejANBgkqhkiG9w0BAQsFADAfMR0wGwYDVQ
QDDBQqLmZlZGVyYXRpb25kZW1vLmNvbTAeFw0yNDAzMDUxNDI0NTZaFw0zNDAzMDUxNDM0NTVa
MB8xHTAbBgNVBAMMFCouZmVkZXJhdGlvbmRlbW8uY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ
8AMIIBCgKCAQEAnyjM01KPv3y4DUiKYpTHDT9Gi4c/EGOU6bs8jh0Mke8TjTVWuHGiD98Mj4qL
bb/yhk4LHemt58gtjxdj9+pjgG38OU3dF0n7RMXES6EwK4KlsO16nrXEG6YtP6ElJPkCNXzzS
eoeHPCTSMxp1gFmY/z8fOyI//x/8AmRI2JfGr43exXCbMjYx4sgr85HOCVdw27uHEK9w0hAPPH
t2vq7BMDAfYj2IisbpVekasJDmXtyhvRFptESJ80qvmmyTLD85iHmO7aME1/7vYnl1RQCqbZbh
trWYl4VBAiy/ySnqJdcaJT3KCOVJZOKZxurjXXNJbTHe8i3sQZ0dP2poJZtQIDAQABo1AwTjAO
BgNVHQ8BAf8EBAMCBaAwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCMB0GA1UdDgQWBB
R2d84eaCTxgzIeXnY41uMia8DkJDANBgkqhkiG9w0BAQsFAAOCAQEACOEinC4t1V80Kgs9MXu8
43e0UqLseOkoClNZbhxM4n3Y9cTPb9QYQLQ69g8Q2d6tG+DzTCAnJeTdM2A9QWpePNuGceSWlF
HHXHv/ZuzixA2SS2mnAVvs9GgP1W/l1anMD1mhd4p9F+U0E/KMnn8yo2pYGI/wlwYm0yW3uaDd
AQ1WS+fZev2n5WcDbQ6WGklOL5jOJPvkiXcXmzhPc1ogsCvsWCL9OGhFxy3buLTp1N3Rk4djZ2
hWyoeU8WjYax2436rfQx2qYJvgtAD4MDAzl95N28kzGBWr+eOO460NzDJ2OGc0rrIZUyo19Uqj
je3lNPPyVAzGp+cyrqeRQWpMjg==
SigningCertificateUpdateStatus           :
Microsoft.Graph.PowerShell.Models.MicrosoftGraphSigningCertificateUpdateSt
atus
AdditionalProperties                     : {[@odata.context,
https://graph.microsoft.com/v1.0/$metadata#domains
('federationdemo.com')/federationConfiguration/$entity]}
```

For further information relating to the `New-MgDomainFederationConfiguration` PowerShell command, see:

learn.microsoft.com/en-us/powershell/module/microsoft.graph.identity.directorymanagement/new-mgdomainfederationconfiguration

4.2 Verify the configuration

To verify the configuration, run the following command.

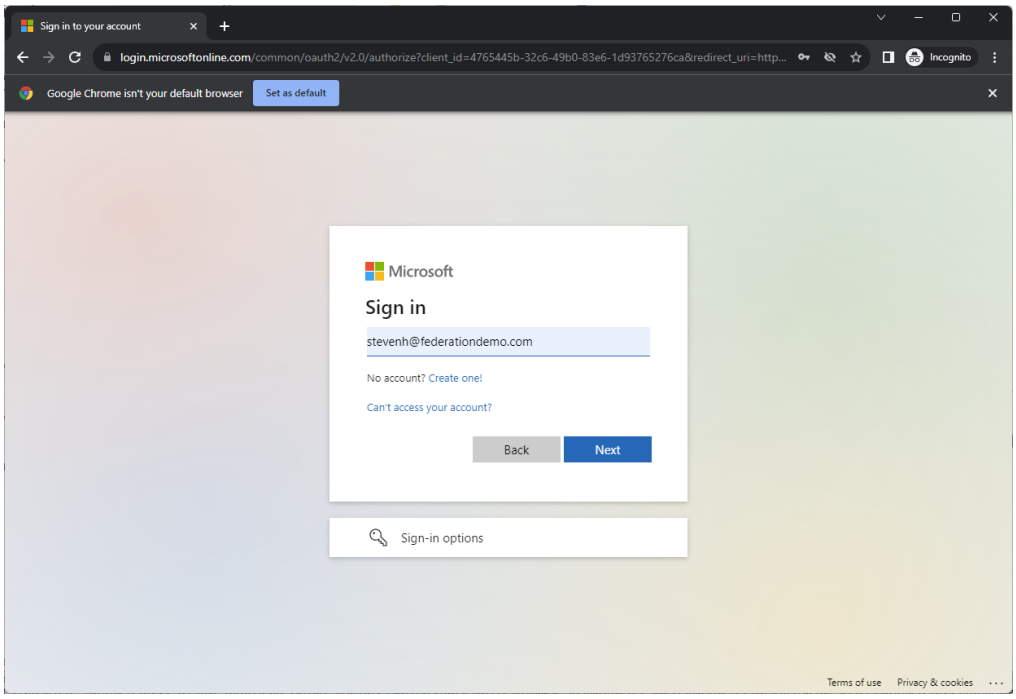
```
PS C:\> Get-MgDomainFederationConfiguration -DomainId 'federationdemo.com'
| Format-List


ActiveSignInUri                          :
DisplayName                              : Federation Demo
FederatedIdpMfaBehavior                  :
Id                                       : dbb35c60-4dc7-4396-86df-
5985564224ff
IsSignedAuthenticationRequestRequired    :
IssuerUri                                : urn:uri:msp:federationdemocom
MetadataExchangeUri                      :
NextSigningCertificate                   :
PassiveSignInUri                         :
https://idp.federationdemo.com/idp/SAML/SingleSignOnService/federationdemo
com
PreferredAuthenticationProtocol          : saml
PromptLoginBehavior                      :
SignOutUri                               :
https://idp.federationdemo.com/idp/SAML/SingleLogoutService/federationdemo
com
SigningCertificate :
MIIDATCCAemgAwIBAgIQdPDr/iI1jbhDMTj5VYya+TANBgkqhkiG9w0BAQsFADAWMRQwEgYDVQ
QDEwt3d3cuaWRwLmNvbTAeFw0xMzExMjIwODIwNTJaFw00OTEyMzExNDAwMDBaMBYxFDASBgNV
BAMTC3d3dy5pZHAuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAi0XJRLDrcb
SyqUd8XG4BgxObQMYLAkENlmJOsAEpl1xMabUiq1X4v0Fc8ZaCpUE3fFGENMEWgBjnQUUE0WtV
Uh5JPMsukolf9qljbJkCkvHXH3O4Uen7vA2oNQWt4bK96SpXADpZKFvpk4D7btKOgU/NamjiqW
HI4fI8kFJKwKBJchRPUQdC4ljRRmGIrSnpY+t25/d3KGXwbe9Z2MGGy2hyA0tgOWuchIK+1vAK
KBUh9nDEXfr80+xW680w5TqHyDcqbWvQsXXhH0yZLfINKNS6/IojHPsBy7tf36Ck9H5Pw+1PPu
6NzBFSz5ZkC8KzrS6vuZXc/ImYrnheMQsqqQIDAQABo0swSTBHBgNVHQEEQDA+gBD4dY4MCPEm
G4sxZrcni8vtoRgwFjEUMBIGA1UEAxMLd3d3LmlkcC5jb22CEHTw6/4iNY24QzE4+VWMmvkwDQ
YJKoZIhvcNAQELBQADggEBABhak2aR84MCdyXO4AKOQvZybsCMdhRq2i1i0WhD4/xe7Ry5haC6
TeXIp8Q4cC3MzsrDal74xHI714BW0loafpHAsXfd9EvkKTVaJ+1Zpe16+SsTL4upS1cGydigqw
UzsdpGck4wI1moJ9477O+46If2gF27u9Cdk7Onxe/5dwLIxWmkVRdbQIH5GsKUeAjOdRQmy+X1
MX6KyRoaCwWGYwxi5Sa+r+3AtDvD4BX0EJGKFZeeM3J/yMpYh/75aN0cFQfDEdJ7C5NE0vonid
E0QtIFvsoWtZUtur2fiW7yBxse38TPQsi2r6A6c/TZsZ5bq31yh3gr3kSN62H8iVKLQLA=
SigningCertificateUpdateStatus           :
Microsoft.Graph.PowerShell.Models.MicrosoftGraphSigningCertificateUpdateSt
atus
AdditionalProperties                     : {}
```

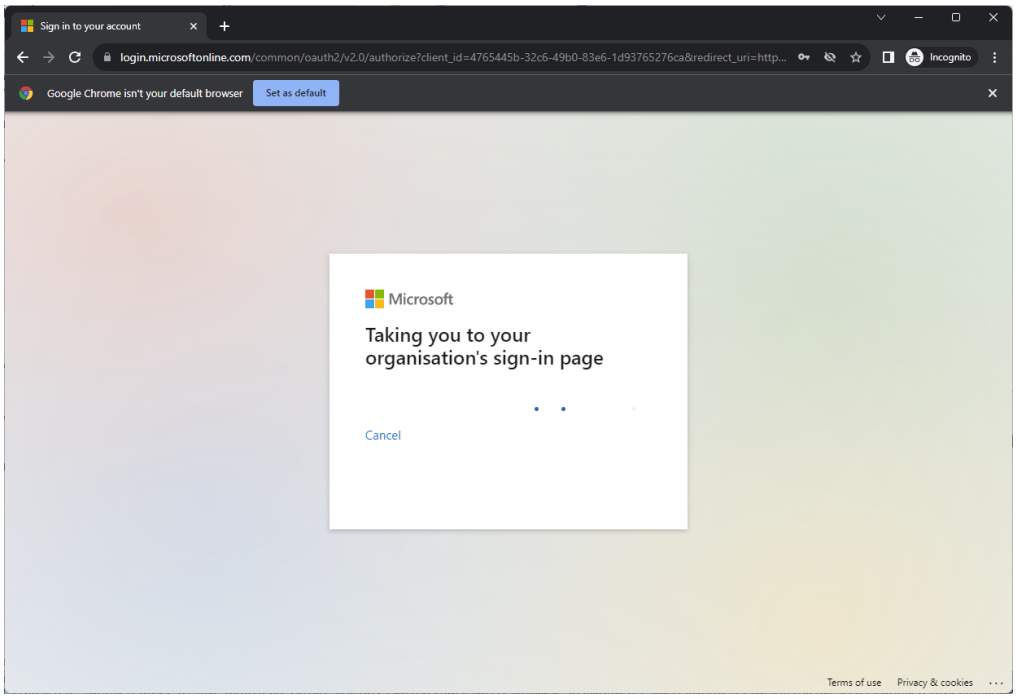
4.3 Test the federation setup

1. To test the federation setup, go to the following URL and sign in:

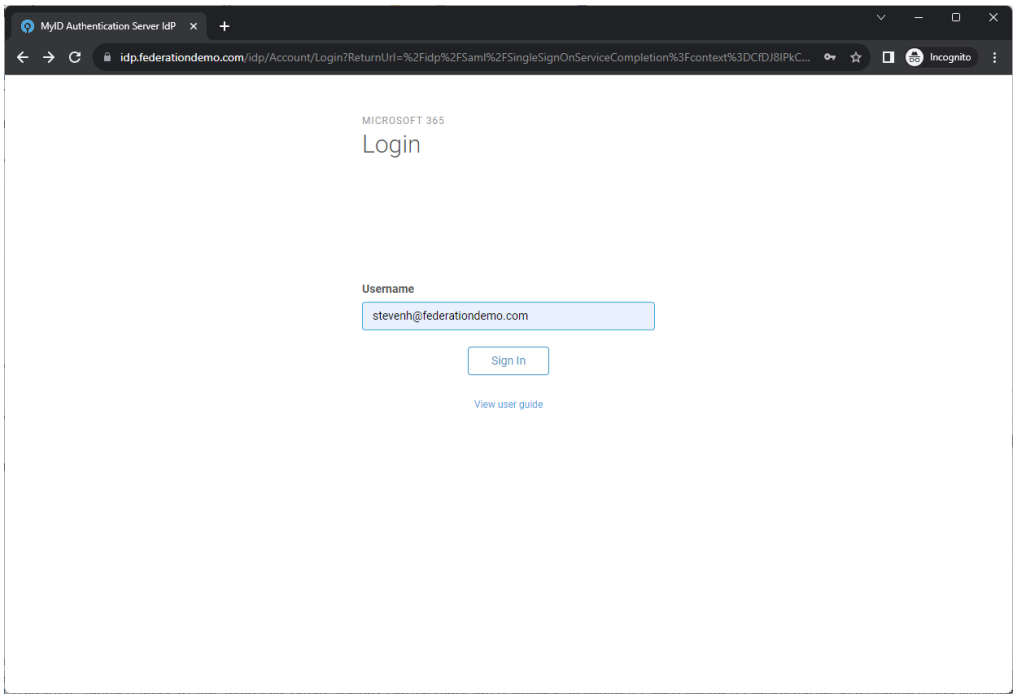
   portal.office.com



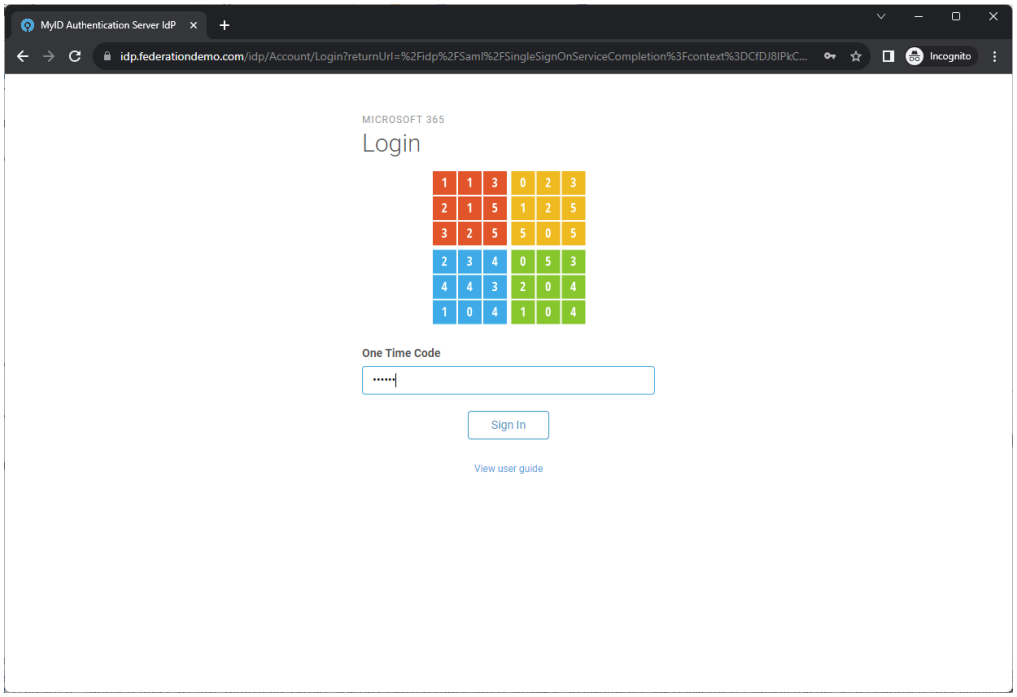
2. Enter your Account name.



3. Wait while Microsoft redirects you to the MyID MFA logon page.

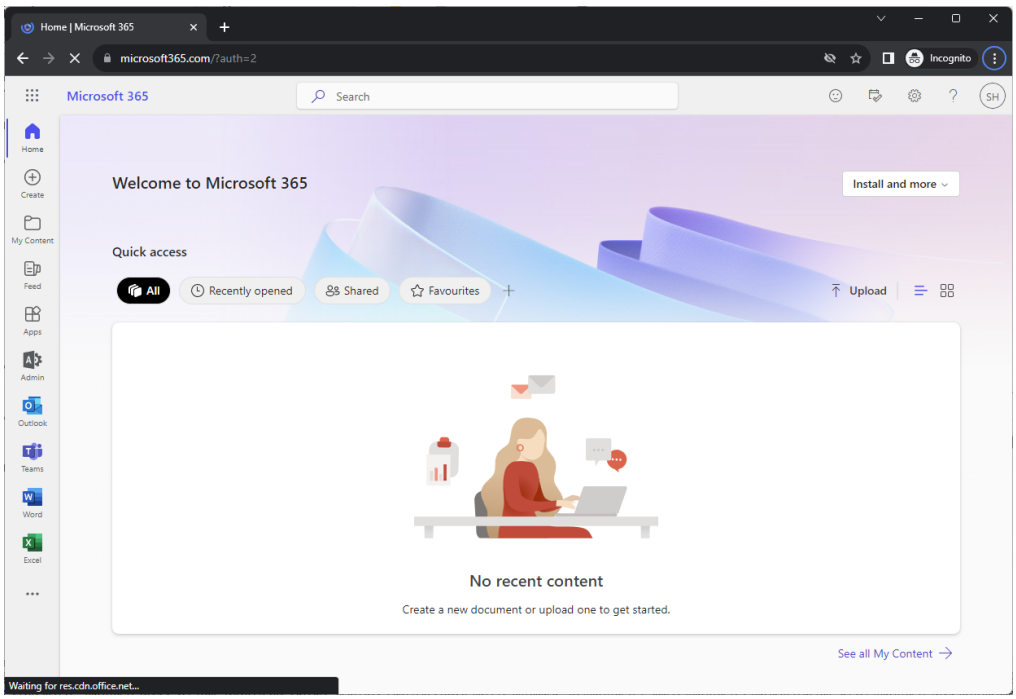


4. Confirm your account name.



5. Enter your MFA details based on the configuration.

   This example uses Passwordless & Deviceless Grid authentication.



Once validated, you are redirected back to Microsoft 365 and are signed in.

5 Multi-domain configuration

Microsoft Entra (Azure AD) requires a unique SAML IssuerUri for each DNS domain across all Azure tenants. The IssuerUri is a federation server value, and not a Microsoft 365 application specific value.

Note: If you have only one DNS domain to federate, this configuration is not required.

When setting up multi-domain federation, you must:

- Add additional domain names.

  section *5.1*, *Adding additional domain names*
- Make multiple unique IssuerUris.

  See section *5.2*, *IssuerUri format impact*.
- Enable multi-domain federation.

  See section *5.3*, *Enable multi-domain federation using PowerShell*.

You may also want to:

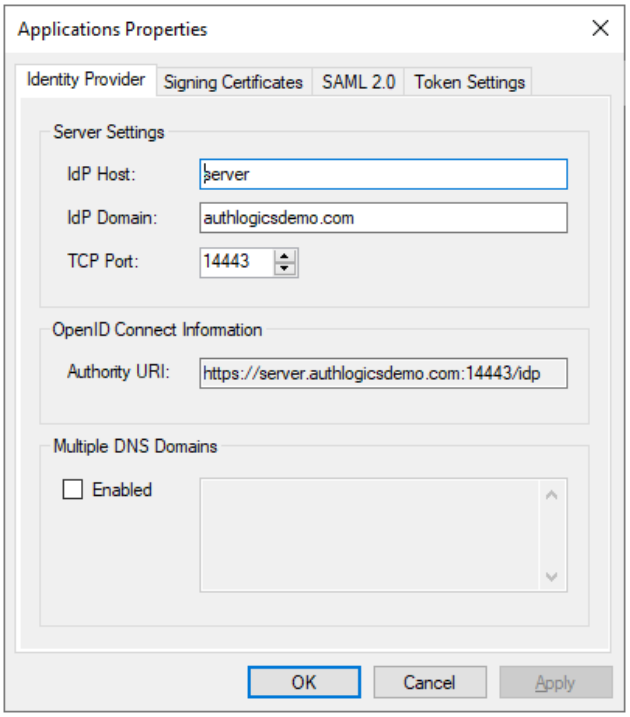
- Change from a single to multi-domain configuration.

  See section *5.4*, *Changing from a single to multi-domain configuration*.

5.1 Adding additional domain names

To support multiple domains, the actual domain names to be federated must be specified. You must add, remove, and view the names through the MMC or through the Rest API.

To specify the names through the MMC, in the **Multiple DNS Domains** section, set the **Enabled** option, then list the domains in the text box.



To specify the names through the API, use the following calls:

- `AddIdpDnsName`
- `RemoveIdpDnsName`
- `GetIdpDnsNames`

When calling `AddIdpDnsName` or `RemoveIdpDnsName`, you must:

- Specify the full DNS name of the domain; for example:

  `entrademo.com`
- Run `iisreset` for the IdP to read the new configuration.

Note: These API calls require administrator rights.

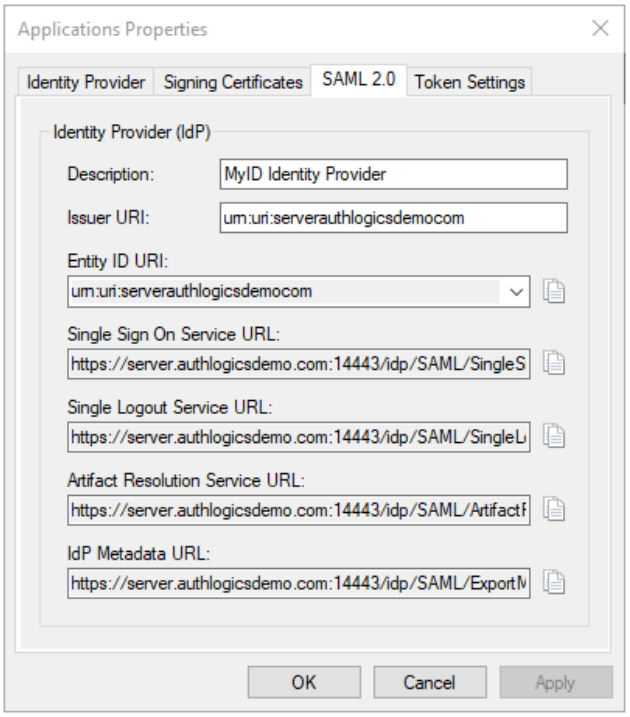
5.2 IssuerUri format impact

The default IssuerUri format for the MyID Authentication Server is:

`urn:uri:{idp-fqdn-without-dots}`

For example:

`urn:uri:idpfederationdemocom`



This ensures that each MyID MFA deployment has a unique IssuerUri for each MyID MFA customer. However, when multiple domains are required to be federated, multiple unique IssuerUri values are also required.

The Authentication Server automatically generates unique IssuerUri names using the configured IssuerUri name appended with the specified DNS domains you want to federate. The resulting format is as follows:

`urn:uri:{idp-fqdn-without-dots}:{federated-domain-without-dots}`

For example, if `acme.com` and `contoso.com` are federated domains with an Authentication Server having a configured IssuerUri of `urn:uri:idpfederationdemocom`, the resulting domain specific URIs are:

- `urn:uri:idpfederationdemocom:acmecom`
- `urn:uri:idpfederationdemocom:contosocom`

You must use these domain specific URIs during each Microsoft 365 configuration.

5.3 Enable multi-domain federation using PowerShell

Configure the yellow highlighted variables in the following sample script with the values of your domain and MyID MFA server.

- `$domain` – The DNS domain to federate, which must already be set up in Microsoft 365.
- `$display` – The friendly name shown to users when signing in to Microsoft 365. You are recommended to use something that is familiar, like your company name.
- `$issuerUri` – The **Issuer URI** from the **SAML 2.0** tab in the Applications Properties dialog. The default value will be a derivative of your IdP host and domain name. This is then appended with a colon and the federated domain name without dots; for example:

  `'urn:uri:ipdfederationdemocom:customdomaincom'`
- `$signinUrl` – The SAML 2.0 IdP sign-on page URL. The path is fixed; however, the DNS name must match the public DNS name of the IdP.
- `$signoutUrl` – The SAML 2.0 IdP logout page URL. The path is fixed; however, the DNS name must match the public DNS name of the IdP.
- `$certificate` – The Base64 representation of the IdP signing certificate. You can obtain this value by clicking the copy icon on the **Signing Certificates** tab of the Applications Properties dialog or by clicking the **Copy Base64** link at the end of the Add Application Wizard.

1. Copy the text from the sample below to a new plain text document and save it as a `.ps1` file.
2. Configure the yellow highlighted variables in the following sample script with the values of your domain and MyID MFA server
3. Add the domain name to the MFA Server using the `AddIdpDnsName` API call.
4. Run the script at a PowerShell command prompt to apply the configuration.

5.3.1 Sample multi-domain PowerShell script

```
$domain = 'federationdemo.com'

$display = 'Federation Demo'

$issueruri = 'urn:uri:idpfederationdemocom:federationdemocom'

$signinUrl =
'https://idp.federationdemo.com/idp/SAML/SingleSignOnService/
federationdemocom'

$signoutUrl =
'https://idp.federationdemo.com/idp/SAML/SingleLogoutService/
federationdemocom'

$certificate =
'MIIDGDCCAgCgAwIBAgIQFaTIA1mLiLtJ+wsZt9M2ejANBgkqhkiG9w0BAQsFADAf
MR0wGwYDVQQDDBQqLmZlZGVyYXRpb25kZW1vLmNvbTAeFw0yNDAzMDUxNDI0NTZa
Fw0zNDAzMDUxNDM0NTVaMB8xHTAbBgNVBAMMFCouZmVkZXJhdGlvbmRlbW8uY29t
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAnyjM01KPv3y4DUiKYpTH
DT9Gi4c/EGOU6bs8jh0Mke8TjTVWuHGiD98Mj4qLbb/yhk4Lhemt58gtjxdj9+pj
gG38OU3dF0n7RMXES6EwK4KlsO16nrXEG6YtP6ElJPkCNXzzSeoeHPCTSMxp1gF
mY/z8fOyI//x/8AmRI2JfGr43exXCbMjYx4sgr85HOCVdw27uHEK9w0hAPPHt2vq
7BMDAfYj2IisbpVekasJDmXtyhvRFptESJ80qvmmyTLD85iHmO7aME1/7vYnl1RQ
CqbZbhtrWYl4VBAiy/ySnqJdcaJT3KCOVJZOKZxurjXXNJbTHe8i3sQZ0dP2poJZ
tQIDAQABo1AwTjAOBgNVHQ8Baf8EBAMCBaAwHQYDVR0lBBYwFAYIKwYBBQUHAwEG
CCsGAQUFBwMCMB0GA1UdDgQWBBR2d84eaCTxgzIeXnY41uMia8DkJDANBgkqhkiG
9w0BAQsFAAOCAQEACOEinC4t1V80Kgs9Mxu843e0UqLseOkoClNZbhxM4n3Y9cTP
b9QYQLQ69g8Q2d6tG+DzTCAnJeTdM2A9QWpePNuGceSWlFHHXHv/ZuzixA2SS2mn
Avvs9GgP1W/l1anMD1mhd4p9F+U0E/KMnn8yo2pYGI/wlwYm0yW3uaDdAQ1WS+fZ
ev2n5WcDbQ6WgklOL5jOJPvkiXcXmzhPc1ogsCvsWCL9OghFxy3buLTp1N3Rk4dj
Z2hWyoeU8WjYax2436rfQx2qYJvgtAD4MDAzl95N28kzGBWr+eOO460NzDJ2Ogc0
rrIZUyo19Uqjje3lNPPyVAzGp+cyrqeRQWpMjg=='


New-MgDomainFederationConfiguration -DomainId $domain -DisplayName $display
-IssuerUri $issuerUri -PassiveSignInUri $signinUrl -SignOutUri $signoutUrl
-SigningCertificate $certificate -PreferredAuthenticationProtocol saml -
federatedIdpMfaBehavior enforceMfaByFederatedIdp | Format-List
```

5.4 Changing from a single to multi-domain configuration

To change from a single domain to a multi-domain configuration:

1. Remove the existing federated connection to Entra.

   See section *2.3*, *Removing an existing federation configuration*.

2. Add the federated domain names to the MFA server.

   See section *5.1*, *Adding additional domain names*.

3. Add the new multi-domain federation configuration.

   See section *5.3*, *Enable multi-domain federation using PowerShell*.

6 Federation without directory synchronization

For federation to work with Entra ID (Microsoft 365 / Azure AD) there must be a specific mapping between the MFA user account and the Entra ID user account.

In a typical Microsoft hybrid configuration, the Microsoft Entra Connect (Azure ID Connect) synchronizes the required fields to link the accounts. If Entra Connect is not in place (for example, in a Managed Service Provider environment) you must map the account attributes manually.

The following user specific attributes must match in both MFA and Entra ID:

- `NameID`

  The Entra user's Immutable ID (`OnPremisesImmutableId`).

  If Microsoft Entra Connect is deployed, the Entra Immutable ID property contains the Base64 equivalent value of the Active Directory `objectGUID` property.

  Important: Once you have set the Immutable ID value, you cannot change it; that is the defining feature of *immutable* IDs.

  If Microsoft Entra Connect has *not* been deployed, the Entra Immutable ID property is empty and *must* be populated before federation will work for the user.

- `IDPEmail`

  The User Principal Name (UPN) value in Entra; not the email address field (despite the attribute name being `IDPEmail`). The UPN is typically the same as the email address, although this may vary.

For further information, see:

learn.microsoft.com/en-us/entra/identity/hybrid/connect/how-to-connect-fed-saml-idp

6.1 Checking the Entra ID values for a user

You can check the Entra ID values for a user with PowerShell or using the Microsoft Entra admin center.

6.1.1 Using PowerShell

To view the values stored in Entra for a user, run the following PowerShell script, replacing the highlighted values for `$tenantId` and `$accountUpn`:

```
$tenantId = 'federationdemo.com'

$accountUpn = 'johnd@federationdemo.com'

Connect-MgGraph -TenantId $tenantId -Scopes 'User.ReadWrite.All' -NoWelcome

Get-MgUser -UserId $accountUpn -Property UserPrincipalName,
OnPremisesImmutableId | fl UserPrincipalName, OnPremisesImmutableId
```

The PowerShell script outputs the user details as follows:

```
UserPrincipalName     : johnd@federationdemo.com

OnPremisesImmutableId : X1vAuhkd70KAAfnfxA6DyA==
```

6.1.2 Using the Microsoft Entra admin center

To view the values stored in Entra for a user using a browser:

1. Open the Entra ID Users page:

   entra.microsoft.com/#blade/Microsoft_AAD_UsersAndTenants/UserManagementMenuBlade/menuId/

2. Select the user.
3. Select the **Properties** tab near the top.
4. Locate the **User principal name** value.
5. Locate the **On-premises immutable ID** value.

6.2 Creating an Immutable ID value in Entra

If you have not deployed Microsoft Entra Connect, the user account does not have an Immutable ID value. If your intention is to configure a hybrid deployment, you must configure Microsoft Entra Connect to set the Immutable ID value. If you do *not* intend to configure a hybrid deployment (that is, you want to remain cloud-only) you must generate and configure an Immutable ID value in both MFA and Entra.

To generate a new Immutable ID value and view the results in Entra for a user, run the following PowerShell script, replacing the highlighted values for `$tenantId` and `$accountUpn`:

```
$tenantId = 'federationdemo.com'

$accountUpn = 'johnd@federationdemo.com'

$newGuid = [system.guid]::newguid()

$base64 = [system.convert]::ToBase64String(([GUID]$newGuid).ToByteArray())

Connect-MgGraph -TenantId $tenantId -Scopes 'User.ReadWrite.All' -NoWelcome

Update-MgUser -UserId $accountUpn -OnPremisesImmutableId $base64

Get-MgUser -UserId $accountUpn -Property UserPrincipalName,
OnPremisesImmutableId | fl UserPrincipalName, OnPremisesImmutableId
```

Important: Once you have set the Immutable ID value, you cannot change it; that is the defining feature of *immutable* IDs.

6.3 Adding the Immutable ID to the MFA user account

Use the MFA Server Rest API to get and set the user account `EntraID` property.

You can specify the value as Base64 or a GUID formats when setting it. When reading the value, the Base64 version is always returned to allow you to match the value displayed in Entra.

Note: This API call requires administrator rights.

6.3.1 Active Directory user

An MFA user created on an Active Directory user account automatically has an `EntraID` property, as this is derived from the `objectGUID` property and cannot be changed. You can only get the value using the MFA server Rest API; you cannot set it.

If you are using an Active Directory user account that is not synchronized with Entra, extract the value from the MFA server and configure it on the Entra user account. Do *not* generate a random GUID (as above) as the account will never match.

6.3.2 External / Realm user

An MFA user created on an External / Realm based user account has an empty `EntraID` property by default. As it is not an actual Active Directory user account, the `objectGUID` value is not used. Instead, you can get and set the user's `EntraID` property using the MFA server Rest API, and you can change it if required.

7 Troubleshooting

This section contains troubleshooting information for situations that may occur when working with federated access.

These situations are:

- The browser looping between Microsoft 365 and the IdP through redirects.

  See section *7.1*, *Browser redirect loop between Microsoft 365 and the IdP*.

- Attempting to use a non-unique IssuerUri value.

  See section *7.2*, *Unique IssuerUri values*.

- A signing certificate error.

  See section *7.3*, *Signing certificate error*.

7.1 Browser redirect loop between Microsoft 365 and the IdP

Microsoft Entra (Azure AD) and 365 are regularly adding features, changing settings, and updating security defaults. This may cause problems with federated connections from time to time.

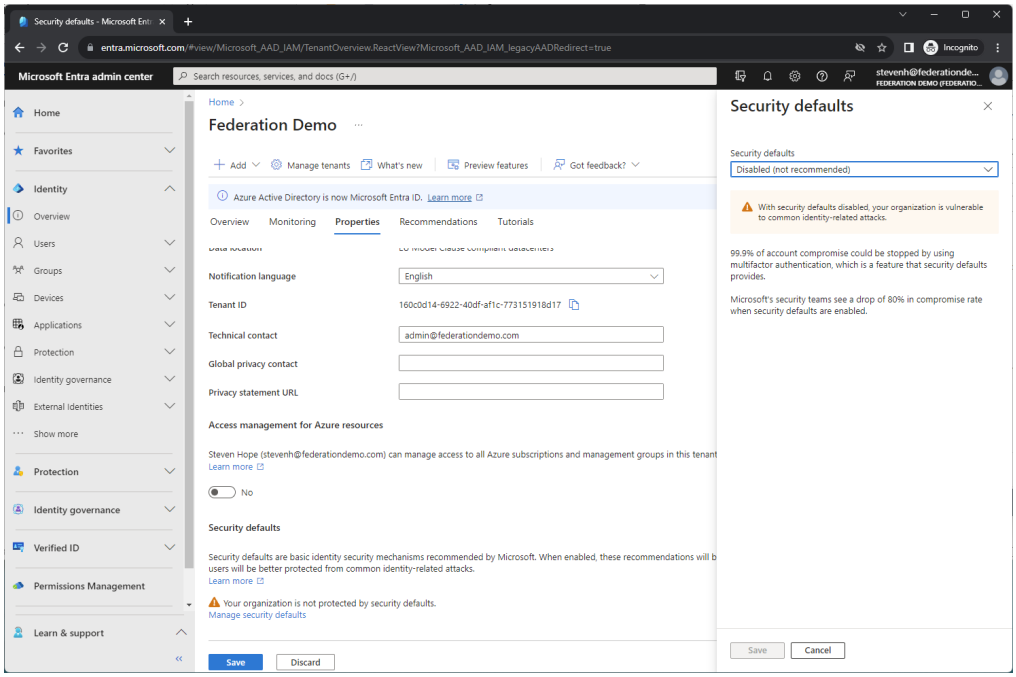
The security defaults of Entra may interfere with third-party federated access, as Entra tries to apply Entra MFA security policies which are not being used. This may result in the inability to access the 365 website after a successful federated logon, including a browser redirect loop.

To resolve this problem using the Microsoft Entra UI:

1. Go to the following URL:

   entra.microsoft.com/#view/Microsoft_AAD_IAM/TenantOverview.ReactView?Microsoft_AAD_IAM_legacyAADRedirect=true

2. Select the **Properties** tab.
3. Scroll down to the bottom and click **Manage security defaults**.
4. Select **Disabled (not recommended)**.
5. Save the settings.



To resolve this problem using PowerShell, run the following script:

```
# Authenticate scope to change Security Defaults
Connect-MgGraph -scope 'Policy.ReadWrite.SecurityDefaults,Policy.Read.All'

# Show Setting
Get-MgPolicyIdentitySecurityDefaultEnforcementPolicy | Select DisplayName, IsEnabled

# Enable Security Defaults
Update-MgPolicyIdentitySecurityDefaultEnforcementPolicy -IsEnabled
```

```
# Disable Security Defaults
Update-MgPolicyIdentitySecurityDefaultEnforcementPolicy -BodyParameter @{isEnabled =
$false}
```

7.2 Unique IssuerUri values

Microsoft requires a unique IssuerUri for each DNS domain registered in Entra; this applies across all tenants, not just your own tenant. If you attempt to federate a domain in Entra using an IssuerUri that has previously been used, may see a PowerShell error similar to:

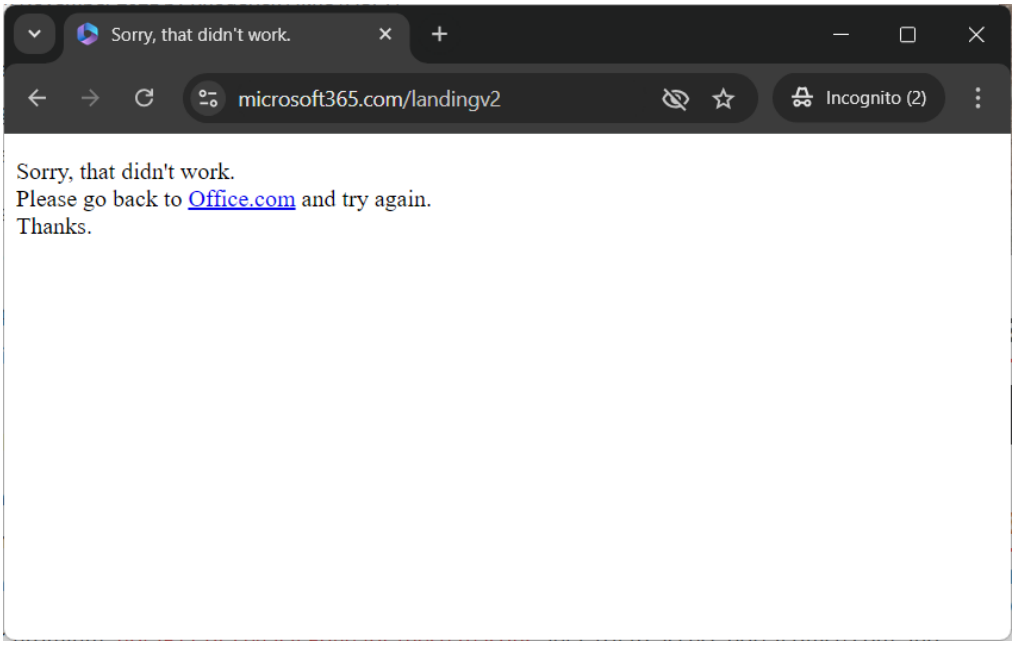
```
Unable to complete this action. Try again later.
```

If you wait and try again as stated, this has no effect. You must make sure your IssuerUri is unique, or make use of the MyID multi-domain configuration; see section *5*, *Multi-domain configuration*.

7.3 Signing certificate error

If, after the MFA login as the browser is redirected back to Microsoft 365, an error occurs similar to the following:

```
Sorry, that didn't work.
Please go back to Office.com and try again.
Thanks.
```

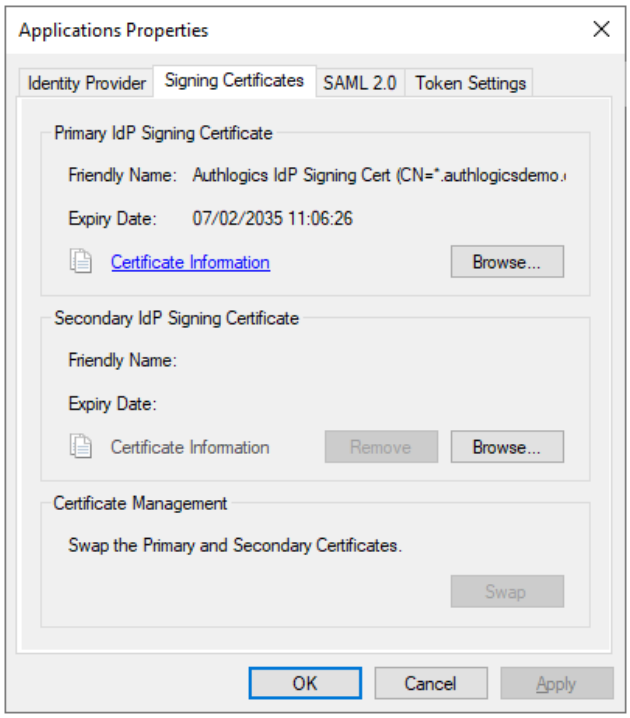


This error can be caused by the following:

- The IdP signing certificate is incorrectly configured.

  To resolve this issue:

  1. On the Applications Properties dialog, select the **Signing Certificates** tab, then click the copy icon to get the Base64 value of the IdP signing certificate.



  2. Run the following PowerShell command:

```
PS C:\> Get-MgDomainFederationConfiguration -DomainId
'federationdemo.com' | select SigningCertificate
```

     The PowerShell returns the signing certificate:

```
SigningCertificate
------------------
MIIDGDCCAgCgAwIBAgIQFaTIA1mLiLtJ+wsZt9M2ejANBgkqhkiG9w0BAQsFADAfMR0w
GwYDVQQDDBQqLmZlZGVyYXRpb25kZW1vLmNvbTAeFw0yNDAzMDUxNDI0NTZaFw0zNDAz
MDUxNDM0NTVaMB8xHTAbBgNVBAMMFCouZmVkZXJhdGlvbmRlbW8uY29tMIIBIjANB...
```

  3. Check that the result returned from the PowerShell script is the same as the Base64 you obtained from the Applications Properties dialog.

     If the results are not the same, this means that Entra is configured to use a different certificate to the IdP. Ensure the IdP is using the correct signing certificate (not to be confused with the SSL certificate), then configure Entra to use the correct Base64 value.

- The MFA authentication server's time or timezone is incorrect.

  If the time or timezone on the authentication server is incorrect, the signing of the SAML information is carried out at an invalid time.

  To resolve this issue, set the time and timezone to the correct settings.