Integration of Microsoft Azure

Contents

Abbreviations:

DCEM = DoubleClue Enterprise Management

1. Introduction

This documentation describes how to integrate Microsoft Azure with DoubleClue Identity & Access Management (IAM). General knowledge and experience with Azure is required for the configuration. For a general instruction on how to use Microsoft Azure, please visit https://azure.microsoft.com/en-us/get-started/.

2. Methods of Integration

There are three different scenarios to integrate Microsoft Azure with DoubleClue:

1. Azure as Identity Provider for Username and Password while MFA Policies are defined in DCEM
2. Azure as full Identity Provider for DoubleClue
3. Azure as a Service Provider Hybrid with Active Directory

3. Azure as Identity Provider for Username and Password (MFA Policies are defined in DCEM)

In this scenario, the DCEM login procedure will verify the user credentials with Microsoft Azure.

If Azure successfully verifies the credentials of a user during login, DCEM will automatically import the user into DCEM database if not yet present. The User will automatically inherit all DCEM privileges according to the Azure Groups he is a member of.

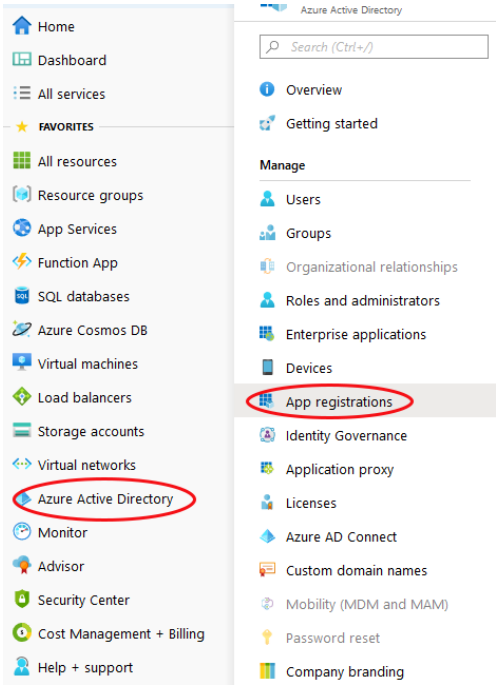
After the Azure verification, DCEM will process the authentication policies and execute the MFA requirements as defined in the DCEM the policies.

Azure will not demand MFA form users during the login in this scenario, even if they are configured for MFA. MFA will be handled by DCEM according to the DCEM policies.

3.1 Create and configure the Azure Application

1. Log into your Global Administrator account on https://portal.azure.com

2. Go to “Azure Active Directory” and choose “App Registrations” in the submenu.

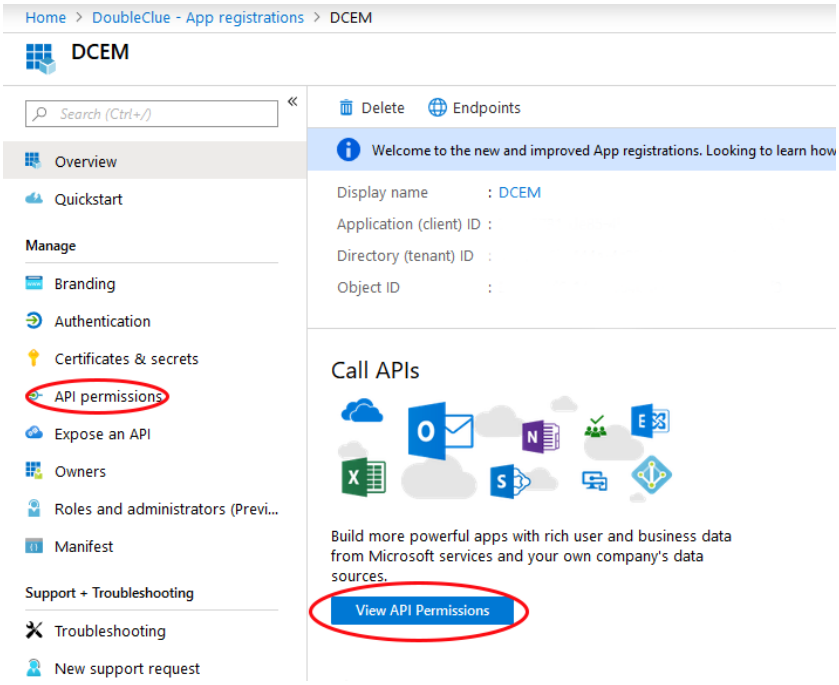


3. Click on “New Registration”. If you have already registered an app you wish to use, select it instead and skip directly to step 5.

4. Give your application an easily recognisable name, eg. “DoubleClue”. Then choose the account type you want to support and click “Register”.

5. Copy the “Application (client) ID” and “Directory (tenant) ID” from the application side. Store them in a safe location – you will need them later.

6. Click on “View API Permissions” or “API Permissions” from the submenu on the left side of the app registration page.



7. Choose “Add a permission” and select “Microsoft Graph” then “Application permissions”. This will open the menu to select permissions.

8. Open the menu item “Directory” and enable “Directory.Read.All”. Then open the item “Users” and enable “User.Read.All”. Confirm the selection with the “Add Permission”-Button at the bottom of the menu.

9. Choose “Add a permission” again and select “Microsoft Graph” then “Delegated permissions”. Open the item “Users” and enable “User.Read.All”. Confirm the selection with the “Add Permission”-Button at the bottom of the menu.

10. Make sure to grant Admin consent to all permissions.

11. Go to “Certificates & Secrets” in the submenu on the left.

12. Add a “New Client Secret” and choose an expiration time.

13. Copy the value of the client secret. Be aware that the secret is only shown once! Should it be lost, it can’t be restored, and a new secret has to be defined.

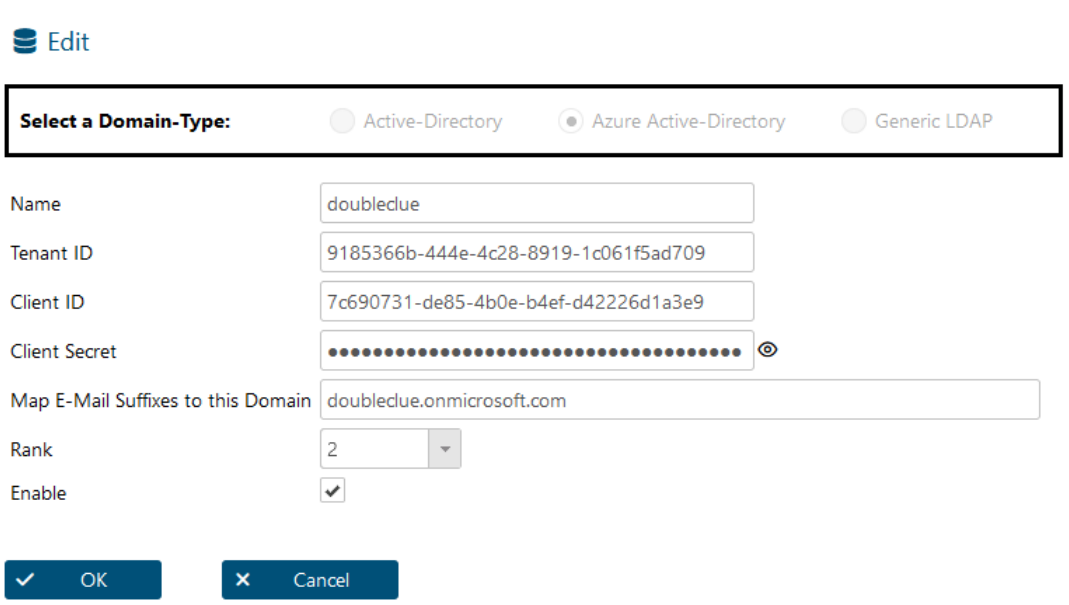
3.2 Connecting with DoubleClue Enterprise Management (DCEM)

1. Log into DCEM as an administrator.

2. Go to “Administration” in the main menu and then to “Domain” in the sub menu.

3. Add a new Domain.

4. Choose “Azure Active-Directory” as Domain Type.



5. Select a meaningful and easily memorable name, like the company name. Note that this name will be the prefix users will need to add to their User ID to identify themselves with DoubleClue.

6. Paste the Tenant ID and Client ID from the value you copied in step 3.1.5 in the respective fields.

7. Paste in the Client Secret from the value you copied in step 3.1.12 in the respective field.

8. In “Map E-Mail Suffixes to this Domain” enter the domain. For example, if user is ‘name.surname@doubleclue.onmicrosoft.com’ then enter ‘doubleclue.onmicrosoft.com’

9. Confirm the input. You now have a successfully connected your DCEM to the Azure app.

3.3 Importing Azure Users to DoubleClue

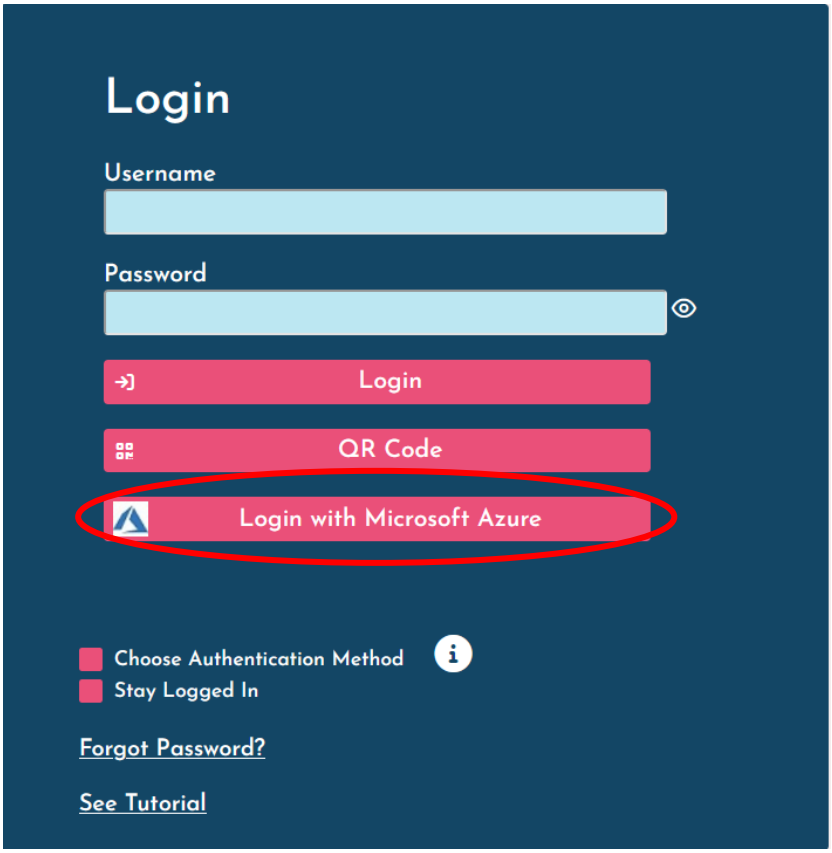
Administrators can import users manually form azure to DCEM, but it isn’t necessary as users will be automatically imported in DCEM when they log in with their Azure credentials.

4. Azure as a Full Identity Provider for DoubleClue

In this scenario, Azure will handle the full user authentication including applying Azure policies and MFA.

This feature must be enabled in DCEM by checking the checkbox ‘Enable Azure Direct Login’ under “Administration” -> “Preferences”. By default, this setting is off.

To log into DCEM using Azure, users will have to select “Login with Microsoft Azure” on the DoubleClue Login Pages (DCEM and UserPortal) or enter the DCEM-URL for azure.



The browser will then automatically redirect the user to the Azure login procedure and apply MFA options as defined in Azure. DoubleClue MFA policies will not be applied. When the Azure login procedure is completed, Azure will redirect the user to the DCEM or UserPortal welcome page and the user has successfully logged in.

4.1 Add the Azure App in DCEM

Add an Azure Domain in DCEM as described in “3.1 Create and configure the Azure Application” and “3.2 Connecting with DoubleClue Enterprise Management (DCEM)”

You need to add the URLs for your DCEM and UserPortal to the Azure-DoubleClue-App as follows:

a) Go to Portal.azure.com
b) Proceed to ‘App Registrations’
c) Select the DoubleClue-App
d) Go to ‘Authentication’
e) Click on ‘Web’
f) Add the DCEM and UserPortal URLS

g) Save the changes

If you use DoubleClue as SAAS, the Dcem URLs for tenants hosted on DoubleClue.online are:

- 'yourTenantName.doubleclue.online/dcem/mgt' for DCEM
- 'yourTenantName.doubleclue.online/dcem/userportal' for UserPortal

4.2 Disable DoubleClue Login for Azure Users

If Azure is configured as Identity Provider for DoubleClue as described above, users can choose whether they want to log in with Azure or use the normal DoubleClue login with their Azure credentials and DoubleClue MFA. You can, however, disable the option to use DoubleClue login and force them login with Azure.

For this, log into DCEM, go to 'Administration' -> 'Preferences' and check the box by 'Use Only Azure Direct Login'. By default, this option is disabled. While DoubleClue login fields will stay enabled in the interface, users will receive an error message if they try to login in this way, instructing them to use Azure login.

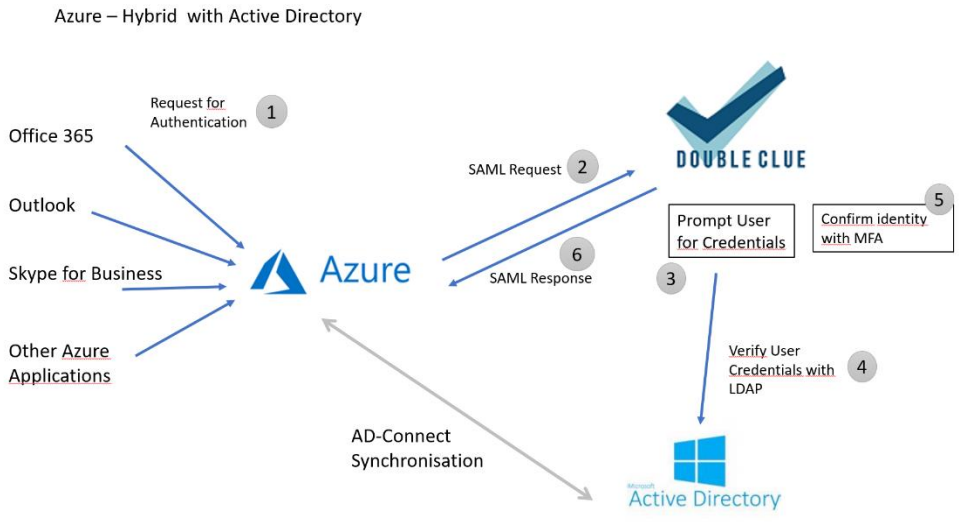
Local users will not be affected by this change.

5. Configure Azure as a Service Provider Hybrid with Active Directory

In this scenario, DCEM will verify the user credentials from an on premisses Active Directory and applies its own MFA policies.

DoubleClue will also applies its MFA policies to all other Azure Apps like Outlook, Teams, Office 365 etc. To configure this scenario, you have to add a federated Domain for Azure.

Within the federation, Azure acts as a SAML Service-Provider and DoubleClue as the SAML Identity-Provider.



The federation uses the SAML protocol for the communication between Azure and DoubleClue. Both parties have first to trust each other by exchanging the SAML metadata before establishing the federation.

5.1 Creating a federated Azure Domain

- The primary domain “tenant.onmicrosoft.com” cannot be configured for federation. You first need to create a new domain.
- The domain must be configured to Azure AD-Connect Sync with the on premises Active Directory.
- Once a domain is federated, the users have to be synchronized with Active Directory or other Identity Providers. You cannot add users or reset user passwords once the domain has been federated.

5.2 Configuring DoubleClue for Azure

DoubleClue comes with a preconfigured Azure metadata. To use it, follow the steps below:

1. Log into DCEM
2. In the main menu, navigate to “SAML” and open the sub menu “Service Providers”
3. Add a new Service Provider and select SP Configuration “Microsoft Azure”
4. Confirm and save the new service provider

5.3 Exporting the DoubleClue Metadata

Azure does not support the SAML standard metadata format. To establish a federation, you therefore require the following data from DoubleClue:

- Logon URL
- Entity ID
- Certificate

Logon URL
The host name of the Logon URL is configured in DCEM. In the main menu, navigate to “SAML” and here to “Preferences”. The host URL is defined in the field “SSO Domain” field, for example: *https://example-url.example-company.com*. The Login URL is composed of the “SSO Domain” and the suffix “dcem/saml”. In our example, the logon URL would be ***https://example-url.example-company.com/dcem/saml***.

Entity ID
Each SAML entity (service providers and identity providers) possess an Entity ID. It is a globally unique identifier exchanged during the software configuration to federate DCEM with Microsoft

Azure domains. The entity ID can be defined in DCEM under SAML -> Preferences. You can choose it freely as long as it is unique.

Certificate
You can download the certificate in DCEM under “SAML” -> “Service Providers”. Click on “Download Idp Metadata” and choose “Download Certificate”.

5.4 Configuring the Azure Domain Federation with Powershell

The Azure Domain federation is configured with Power Shell commands. To use the Windows PowerShell cmdlets, you must first download the Azure Active Directory Modules.

⚠ The code sample in this chapter is meant for administrators who want to federate only one Azure domain with DCEM. If you want to federate several Azure domains with DCEM, please check the further instructions in chapter 4.7 Federating multiple Azure Domains with one DCEM Cluster.

1) Connect to your Azure AD Directory as a tenant administrator with:
   Connect-MsolService

2) Now use **Set-MsolDomainAuthentication** to federate the azure Domain. Use the following sample with the data you exported from DoubleClue as described in chapter 4.3:

```
$dom = " yourazure.domain.com "
$BrandName = "DoubleClue SAML 2.0 IDP"
$LogOnUrl = “https://example-url.example-company.com/dcem/saml”
$LogOffUrl = “https://example-url.example-company.com/dcem/saml”
$EntityId = "example-url.example-company.de"
$MySigningCert = "
MIIC7jCCAdagAwIBAgIQRrjsbFPaXIlOG3GTv50fkjANBgkqhkiG9w0BAQsFADAzMTEwLwYDVQQD
EyhBREZTIFNpZ25pbmcgLSBXUzIwMTJSMi0wLnN3aW5mb3JtZXIuY29tMB4XDTE0MDEyMDE1M
TY0MFoXDTE1MDEyMDE1MTY0MFowMzExMC8GA1UEAxMoQURGUyBTaWduaW5nIC0gV1My
MDEyUjItMC5zd2luZm9ybWVyLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA
Ke+rLVmXy1QwCwZwqgbbp1/kupQVcjKuKLitVDbssFyqbDTjP7WRjlVMWAHBI3kgNT7oE362Gf2
WMJFf1b0HcrsgLin7daRXpq4Qi6OA57sW1YFMj3sqyuTP0eZV3S4+ZbDVob6amsZIdIwxaLP9Zfyw
g2bLsGnVldB0+XKedZwDbCLCVg+3ZWxd9T/jV0hpLIIWr+LCOHqq8n8beJvlivgLmDJo8f+EITnAxW
csJUvVai/35AhHCUq9tc9sqMp5PWtabAEMb2AU72/QlX/72D2/NbGQq1BWYbqUpgpCZ2nSgvlW
DHlCiUo//UGsvfox01kjTFlmqQInsJVfRxF5AcCAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAi8c6C4z
aTEc7aQiUgvnGQgCbMZbhUXXLGRpjvFLKaQzkwa9eq7WLJibcSNyGXBa/SfT5wJgsm3TPKgSehGA
OTirhcqHheZyvBObAScY7GOT+u9pVYp6raFrc7ez3c+CGHeV/tNvy1hJNs12FYH4X+ZCNFIT9tprieR
25NCdi5SWUbPZL0tVzJsHc1y92b2M2FxqRDohxQgJvyJOpcg2mSBzZZIkvDg7gfPSUXHVS1MQs0R
HSbwq/XdQocUUhl9/e/YWCbNNxlM84BxFsBUok1dH/gzBySx+Fc8zYi7cOq9yaBT3RLT6cGmFGVY
ZJW4FyhPZOCLVNsLlnPQcX3dDg9A=="


Set-MsolDomainAuthentication`
-DomainName $dom `
-FederationBrandName $dom`
-Authentication Federated `

-IssuerUri $EntityId `
-PassiveLogOnUri $LogOnUrl `
-LogOffUri $LogOffUrl
-SigningCertificate $MySigningCert `
-PreferredAuthenticationProtocol “Samlp”
```

DomainName specifies the fully qualified domain name (FQDN)

FederationBrandName specifies the name of the string value shown to users when signing in to Azure Active Directory services. We recommend that customers use something that is familiar to them, like their company name

$MysigningCert is the certificate you downloaded from “Exporting the DoubleClue metadata”

$LogOnUrl, $LogOffUrl is the host URL, which is defined in the field “SSO Domain” with the suffix “dcem/saml”.

$EntityId is defined in DCEM under SAML -> Preferences. In case you want to federate multiple Domains, see (4.7).

Authentication specifies the authentication type of the domain (Federated)

Samlp is the authentication protocol

3) To check the configuration, you can execute
```
Get-MsolDomainFederationSettings -domainname "mydomain.com" | format-list *
```

Microsoft references:

https://docs.microsoft.com/en-us/azure/active-directory/hybrid/how-to-connect-fed-saml-idp#configuring-a-domain-in-your-azure-ad-directory-for-federation

https://docs.microsoft.com/en-us/previous-versions/azure/dn194112(v=azure.100)?redirectedfrom=MSDN

Note

If you converted a domain, rather than adding one, it may take up to 24 hours to set up single sign-on. **Before you verify single sign-on, you should finish setting up Active Directory synchronization, synchronize your directories and activate your synced users**.

5.5 Users in federated Domain

Users in federated domain can be created by Azure Ad connect. For more detailed information, see Integrate your on-premises directories with Azure Active Directory.

Windows PowerShell can also be used to automatically add new users to Azure AD and to synchronize changes from the on-premises directory.

Note

The New-MsolUser cmdlet can be used to create a new user in Microsoft Azure Active Directory. In order for this to have access to services, it must have a license (via the parameter"LicenseAssignment").

The following procedure shows how to add a single user to Azure AD.

1. Connect to your Azure AD Directory as a tenant administrator: *Connect-MsolService*.

2. Create a new user principal:
   PowerShellCopy

```
New-MsolUser
 -UserPrincipalName john.smith@example.com
 -ImmutableId ABCDEFG1234567890
 -DisplayName "John Smith"
 -FirstName "John"
 -LastName "Smith "
 -AlternateEmailAddresses "john.smith@something.com"
```

```
-LicenseAssignment "samlp2test:ENTERPRISEPACK"
-UsageLocation "DE"
```

For more information about “New-MsolUser”, see
https://technet.microsoft.com/library/dn194096.aspx

5.6 Verify Access to Office online

You can now log into office.com using DoubleClue as your Identity Provider.

1) Open your browser and enter https://office.com
2) As username, enter the user principal name like name.surname@your-domain-name
3) As your domain is federated, Azure will redirect you to the DoubleClue login page.
4) After a successful DoubleClue authentication, you will be automatically redirected to Office.com as a login user.

5.7 Federating multiple Azure Domains with one DCEM cluster

It is possible to federate several Azure Domains with one DCEM cluster. To establish such a federation, you need to follow the following instructions during the configuration of both – DCEM and Azure.

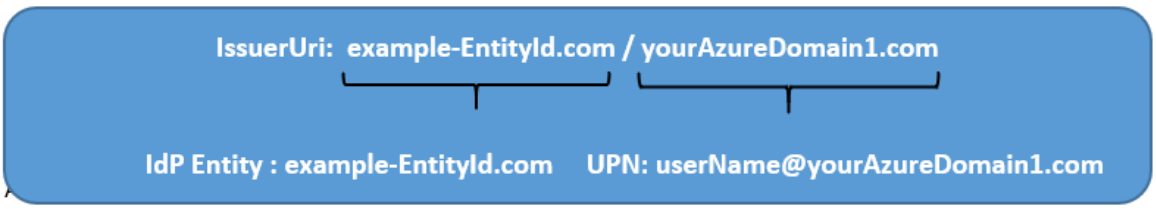
Per default, every Service Provider in DoubleClue needs a unique Entity ID and every Identity Provider in Azure needs a unique Entity ID. This information is defined in the metadata. To federate several azure domains with one DCEM, do the following:

In Azure:

When formulating the Entity ID for your DCEM in Azure, ensure that you compose it according to the following formula – the DCEM entity and the domain name from your UPN separated by a slash:

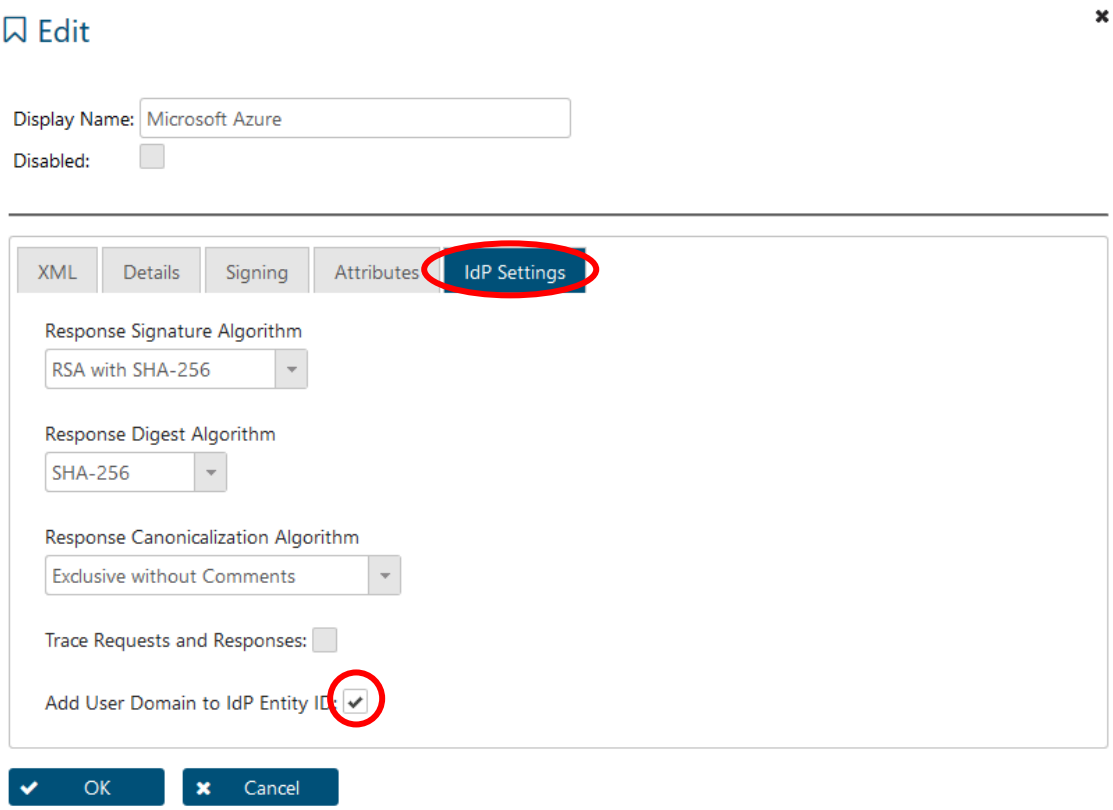
the DCEM entity ID (as defined in the SAML preferences) + / + the domain name from the UPN.

Example:



In DCEM:

All Azure Domains use the same Entity ID. It is therefore not possible, to add Azure as a Service Provider several times. However, one Azure Entry will recognize several domains if you check the **'Add User Domain to IdP Entity'** under IdP Settings.



If the box is checked, DCEM will add the domain name to the Entity ID which is defined in preferences and will send the IdP Entity ID as **EntityID/DomainName** in SAML responses**.** It is thereby recognized as the unique Entity ID by Azure. If a SAML request is send from Azure to DCEM, the Entity ID will be automatically divided into the Entity ID as defined in the preferences and the domain name by DCEM.

Adjusted Powershell Code:

Below you will find samples to configure two Azure domains for one DCEM.

```
$dom = " yourAzureDomain1.com "
$BrandName = "DoubleClue SAML 2.0 IDP"
$LogOnUrl = “https://example-url.example-company.com/dcem/saml”
$LogOffUrl = “https://example-url.example-company.com/dcem/saml”
$EntityId = "example-EntityID.de/yourAzureDomain1.com"
$MySigningCert = "
MIIC7jCCAdagAwIBAgIQRrjsbFPaXIlOG3GTv50fkjANBgkqhkiG9w0BAQsFADAzMTEwLwYDVQQD
EyhBREZTIFNpZ25pbmcgLSBXUzIwMTJSMi0wLnN3aW5mb3JtZXIuY29tMB4XDTE0MDEyMDE1M
TY0MFoXDTE1MDEyMDE1MTY0MFowMzExMC8GA1UEAxMoQURGUyBTaWduaW5nIC0gV1My
MDEyUjItMC5zd2luZm9ybWVyLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA
Ke+rLVmXy1QwCwZwqgbbp1/kupQVcjKuKLitVDbssFyqbDTjP7WRjlVMWAHBI3kgNT7oE362Gf2
WMJFf1b0HcrsgLin7daRXpq4Qi6OA57sW1YFMj3sqyuTP0eZV3S4+ZbDVob6amsZIdIwxaLP9Zfyw
g2bLsGnVldB0+XKedZwDbCLCVg+3ZWxd9T/jV0hpLIIWr+LCOHqq8n8beJvlivgLmDJo8f+EITnAxW
csJUvVai/35AhHCUq9tc9sqMp5PWtabAEMb2AU72/QlX/72D2/NbGQq1BWYbqUpgpCZ2nSgvlW
DHlCiUo//UGsvfox01kjTFlmqQInsJVfRxF5AcCAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAi8c6C4z
aTEc7aQiUgvnGQgCbMZbhUXXLGRpjvFLKaQzkwa9eq7WLJibcSNyGXBa/SfT5wJgsm3TPKgSehGA
OTirhcqHheZyvBObAScY7GOT+u9pVYp6raFrc7ez3c+CGHeV/tNvy1hJNs12FYH4X+ZCNFIT9tprieR
25NCdi5SWUbPZL0tVzJsHc1y92b2M2FxqRDohxQgJvyJOpcg2mSBzZZIkvDg7gfPSUXHVS1MQs0R
HSbwq/XdQocUUhl9/e/YWCbNNxlM84BxFsBUok1dH/gzBySx+Fc8zYi7cOq9yaBT3RLT6cGmFGVY
ZJW4FyhPZOCLVNsLlnPQcX3dDg9A=="


Set-MsolDomainAuthentication`
-DomainName $dom `
-FederationBrandName $dom`
-Authentication Federated `

-IssuerUri $EntityId `
-PassiveLogOnUri $LogOnUrl `
-LogOffUri $LogOffUrl
-SigningCertificate $MySigningCert `
-PreferredAuthenticationProtocol “SamlP”
```