

Sitecore Identity Server Upgrade Guide

Sitecore Identity Server 7.0.0

How to upgrade to Sitecore Identity Server 7.0.0

Table of Contents

Chapter 1	Installation and Upgrade.....	3
1.1	Prerequisites.....	4
1.1.1	ASP.NET.....	4
1.1.2	Sitecore Installation Framework.....	4
1.1.3	Define the security database.....	4
1.2	Prepare for the upgrade.....	5
1.2.1	Backup the security database.....	5
1.2.2	Download the files.....	5
1.3	Upgrade the database.....	6
1.4	Install and configure Sitecore Identity Server.....	7
1.4.1	Configure Sitecore Identity Server.....	8
1.4.2	Configure a Sitecore Identity client.....	8
1.5	Configure the Sitecore Identity Client modules.....	9
1.5.1	Horizon.....	9
1.5.2	Tenant service.....	10
1.5.3	Sitecore Experience Commerce.....	10

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2022 Sitecore. All rights reserved.

Chapter 1

Installation and Upgrade

This guide describes how to upgrade to the latest version of Sitecore Identity Server from Sitecore Identity Server 2.0.0 or later.

The instructions in this guide are not valid for containerized solutions.

To upgrade to the latest version of Sitecore Identity Server, you must install the new version of Sitecore Identity Server and then configure your Sitecore XP solution to use it.

Sitecore Identity Server 7.0.0 is compatible with Sitecore XP 9.1 - 10.3.X.

This chapter contains the following sections:

- Prerequisites
- Upgrade the database
- Install and configure Sitecore Identity Server

1.1 Prerequisites

To upgrade to the latest version of Sitecore Identity Server, you must meet all the necessary requirements.

1.1.1 ASP.NET

Sitecore Identity Server is compatible with the client and server operating systems that support ASP.NET Core 6.0.

Sitecore Identity Server requires [ASP.NET Core 6.0 Runtime – Windows Hosting Bundle 6.0.4 or later](#).

1.1.2 Sitecore Installation Framework

To install Sitecore Identity Server, you must use the latest version of Sitecore Installation Framework (SIF). For more information about SIF, see [Sitecore Installation Framework Configuration Guide](#).

1.1.3 Define the security database

The *security* database contains the Sitecore membership objects. You can store these objects in the *core* database or in the *security* database.

You use the connection strings to specify, where the Sitecore membership objects are stored:

- If you are not using the *security* database, the connection strings for the *core* and *security* databases both point to the *core* database and the Sitecore membership objects are stored in the *core* database.
- If you have extracted the *security* database from the *core* database, the connection strings for the *core* and *security* databases are different, and the Sitecore membership objects are stored in the *security* database.

For more information about the *security* database, see the topic [Walkthrough: Moving security data to a separate database](#).

1.2 Prepare for the upgrade

1.2.1 Backup the security database

Before you start the upgrade, make a backup of your security database.

1.2.2 Download the files

[Download](#) the following resources:

- `Sitecore.IdentityServer.7.0.0-rXXXXX.scwdp.zip`
- IdentityServer Deployment Configuration files 7.0.0.zip
- `Sitecore.IdentityServer.UpgradeScripts.7.0.0.zip`

1.3 Upgrade the database

Before you install the new version of Sitecore Identity Server, you must upgrade the *security* database.

To upgrade the security database:

1. Unpack the `Sitecore.IdentityServer.UpgradeScripts.7.0.0.zip` file.
This file contains the `CMS_security_IdentityServer.sql` script.
2. Run the `CMS_security_IdentityServer.sql` script on the *security* database.

The database upgrade script contains the changes that must be applied to the *security* database and modifies some tables to support new functionality.

1.4 Install and configure Sitecore Identity Server

To install Sitecore Identity Server:

1. Create a folder, for example `c:\resourcefiles`.
2. Unpack the `IdentityServer Deployment Configuration 7.0.0.zip` file and copy the extracted files to the `c:\resourcefiles` folder.
3. Copy the `Sitecore.IdentityServer.7.0.0-rXXXXX.scwdp.zip` file to the `c:\resourcefiles` folder.

The folder should now contain the following items:

- o `createcert.json`
 - o `IdentityServer.json`
 - o `Install-IdentityServer.ps1`
 - o `Sitecore.IdentityServer.7.0.0-rXXXXX.scwdp.zip`
4. Save a copy of your Sitecore license file in the `c:\resourcefiles` folder as `license.xml`.
 5. Edit the `Install-IdentityServer.ps1` file, and update the following parameters:

Parameter	Description	Required
<code>\$prefix</code>	Use the name of the CM instance, it is used for the URL of the Identity Server.	
<code>\$CMurl</code>	Update or leave the default value.	
<code>\$IdentityServer</code>	Update or leave the default value, the URL of the Identity Server.	Yes
<code>\$ClientSecret</code>	Must be the same as the <code>sitecoreidentity.secret</code> key that you use in the <code>...\App_Config\ConnectionStrings.config</code> file on the CM instance.	Yes
<code>\$SqlServer</code>	The name of the SQL server.	Yes
<code>\$SqlAdminUser</code>	The name of the SQL server admin.	Yes
<code>\$SqlAdminPassword</code>	The SQL server admin password.	Yes
<code>\$PSScriptRoot</code>	Update this parameter if you use a folder with another name. Default value = <code>c:\resourcefiles</code> .	
<code>\$SitePhysicalRoot</code>	Update this parameter if you use a folder with another name. Default value = <code>C:\inetpub\wwwroot</code>	
<code>\$wdpFileName</code>	Use the default value. Default value = <code>Sitecore.IdentityServer.*.scwdp.zip</code>	

6. Open PowerShell as an administrator, go to the `c:\resourcefiles` folder, and run the following cmdlet:

```
.\Install-IdentityServer.ps1
```

1.4.1 Configure Sitecore Identity Server

After you have installed Sitecore Identity Server, you must configure it.

To configure Sitecore Identity Server:

1. On the Sitecore Identity Server instance, open the `...\Config\production\Sitecore.IdentityServer.Host.xml` file
2. Update the connection string to use the *security* database as the `Data Source`:

```
<ConnectionString>Data Source=.;Initial Catalog=Sitecore.Core;User ID=coreuser;Password=Test12345</ConnectionString>
```

3. Ensure that the `AllowedCorsOriginsGroup1`, and the `PasswordRecoveryUrl` settings contain the URL of a Sitecore instance:

```
<PasswordRecoveryUrl>http://sitecore_instance/sitecore/login?rc=1</PasswordRecoveryUrl>  
<AllowedCorsOriginsGroup1>http://sitecore_instance</AllowedCorsOriginsGroup1>
```

Note

In a scaled environment, these settings must point to a Content Management server.

4. Ensure that `ClientSecret1` is the same as the `sitecoreidentity.secret` key that you use in the `...\App_Config\ConnectionStrings.config` file on the CM instance.
5. Ensure that any configuration changes you have implemented are reconfigured in separate configuration patch files.
6. Add any custom plugins to the `{SI_server_root_folder}\sitecore\` folder.
7. Start Sitecore Identity Server.

1.4.2 Configure a Sitecore Identity client

Since Sitecore XP 9.1.0, every Sitecore instance is a client of Sitecore Identity Server by default.

To configure a Sitecore instance to use Sitecore Identity Server for logging in:

1. To register Sitecore Identity Server on a Sitecore instance, in the `...\App_Config\Sitecore\Owin.Authentication.IdentityServer\Sitecore.Owin.Authentication.IdentityServer.config` file, in the `identityServerAuthority` variable, update the URL of the Sitecore Identity Server:

```
<sc.variable name="identityServerAuthority" value="https://SitecoreIdentityServerHost"/>
```


1.5 Configure the Sitecore Identity Client modules

1.5.1 Horizon

After you have installed Sitecore Identity Server, you must update the Horizon instance.

Sitecore XP 10.0.X

1. On the Sitecore Identity Server instance, open the `...\Config\production\Sitecore.IdentityServer.Host.xml` file.
2. In the `AllowedCorsOrigins` node, add the `HorizonHttpsCorsOrigin` setting and ensure that the URL points to the Horizon instance:

```
<AllowedCorsOrigins>
  <AllowedCorsOriginsGroup1>https://CM_Instance</AllowedCorsOriginsGroup1>
  <HorizonHttpsCorsOrigin>https://Horizon_Instance_</HorizonHttpsCorsOrigin>
</AllowedCorsOrigins>
```

3. On the Horizon instance, open the `your_horizon_instance\sitecore\Sitecore.Horizon.Authoring.Plugin\Config\OpenIdConnect.xml` file.
4. In the `OpenIdConnectOptions` node, ensure that the `Authority` setting contains the URL of the Sitecore Identity Server.
5. In the `BearerAuthenticationOptions` node, ensure that the `Authority` setting contains the URL of the Sitecore Identity Server.
6. Restart the Horizon instance.
7. Restart Sitecore Identity Server.

Sitecore XP 10.1.X, 10.2.X

1. On the Sitecore Identity Server instance, open the `...\Config\production\Sitecore.IdentityServer.Host.xml` file.
2. In the `AllowedCorsOrigins` node, add the `HorizonHttpsCorsOrigin` setting and ensure that the URL points to the Horizon instance:

```
<AllowedCorsOrigins>
  <AllowedCorsOriginsGroup1>https://CM_Instance</AllowedCorsOriginsGroup1>
  <HorizonHttpsCorsOrigin>https://Horizon Instance </HorizonHttpsCorsOrigin>
</AllowedCorsOrigins>
```

3. On the Horizon instance, open the `...\Config\OpenIdConnect.xml` file.
4. In the `OpenIdConnectOptions` node, ensure that the `Authority` setting contains the URL of the Sitecore Identity Server.
5. In the `BearerAuthenticationOptions` node, ensure that the `Authority` setting contains the URL of the Sitecore Identity Server.
6. Restart the Horizon instance.
7. Restart Sitecore Identity Server.

Sitecore XP 9.3

1. On the Sitecore Identity Server instance, open the `...\Config\production\Sitecore.IdentityServer.Host.xml` file

- In the `AllowedCorsOrigins` node, add the `HorizonHttpsCorsOrigin` setting and the `HorizonHttpCorsOrigin` setting and ensure that the URL points to a valid Horizon instance:

```
<AllowedCorsOrigins>
  <AllowedCorsOriginsGroup1>https://CM_Instance</AllowedCorsOriginsGroup1>
  <HorizonHttpsCorsOrigin>https://Horizon_Instance_</HorizonHttpsCorsOrigin>
  <HorizonHttpCorsOrigin>http://Horizon_Instance_</HorizonHttpCorsOrigin>
</AllowedCorsOrigins>
```

- On the Horizon instance, open the `...\Config\OpenIdConnect.xml` file,
- In the `OpenIdConnectOptions` node, ensure that the `Authority` setting contains the URL of the Sitecore Identity Server.
- In the `BearerAuthenticationOptions` node, ensure that the `Authority` setting contains the URL of the Sitecore Identity Server.
- Restart the Horizon instance.
- Restart Sitecore Identity Server.

1.5.2 Tenant service

If the Tenant service uses `auth-endpoint`, you must update it.

To update the `auth-endpoint` of the Tenant service instance:

- On the Tenant service instance, open the `...App_Config\ConnectionStrings.config` file.
- In the `<sitecore>` node, update the endpoint.

```
<add name="sitecore" connectionString="database=master;tenant
id={TenantId};host=https:// CM_Instance;timeout=20;auth
endpoint=https://Sitecore_Identity_Server_instance/;client secret=Client_Secret;client
id=SitecoreCredentials" />
```

1.5.3 Sitecore Experience Commerce

To configure Sitecore Experience Commerce:

Sitecore Experience Commerce (XC) deploys one file to Identity Server – `Sitecore.Commerce.IdentityServer.Host.xml`.

Locate the `Sitecore.Commerce.IdentityServer.Host.xml` file on your original Identity Server installation and make the following modifications

A sample `Sitecore.Commerce.IdentityServer.Host.xml` file with changed content is attached here for reference but you must merge the changes into your customized version.

- Copy the `\Config\production\Sitecore.Commerce.IdentityServer.Host.xml` file from your original Identity Server installation.
- Open the file and
 - Add `<ApiScopes>` :

```
<ApiScopes>
  <EngineApiScope>
    <Name>EngineAPI</Name>
    <DisplayName>Commerce Engine API</DisplayName>
  </EngineApiScope>
  <PostmanApiScope>
    <Name>postman_api</Name>
    <DisplayName>Postman API</DisplayName>
```

```

</PostmanApiScope>
</ApiScopes>

```

- 2) In the <ApiResources> node, add the following <Scopes>:

```

<ApiResources>
  <EngineApiResource>
    <Name>EngineAPI</Name>
    <DisplayName>Engine API</DisplayName>
    <ApiSecrets>
      <ApiSecret1>secret</ApiSecret1>
    </ApiSecrets>
    <UserClaims>
      <UserClaim1>name</UserClaim1>
      <UserClaim2>email</UserClaim2>
      <UserClaim3>role</UserClaim3>
    </UserClaims>
    <Scopes>
      <Scope1>EngineAPI</Scope1>
    </Scopes>
  </EngineApiResource>
  <PostmanApiResource>
    <Name>postman_api</Name>
    <DisplayName>Postman API</DisplayName>
    <ApiSecrets>
      <ApiSecret1>secret</ApiSecret1>
    </ApiSecrets>
    <UserClaims>
      <UserClaim1>name</UserClaim1>
      <UserClaim2>email</UserClaim2>
      <UserClaim3>role</UserClaim3>
    </UserClaims>
    <Scopes>
      <Scope1>postman_api</Scope1>
    </Scopes>
  </PostmanApiResource>
</ApiResources>

```

3. Copy the edited Sitecore.Commerce.IdentityServer.Host.xml file to the \Config\production folder on the new Identity Server installation.
4. If the URL of the Identity Server has been changed for the new installation, you must update the Identity Server URL in the following files:
 - o Commerce Engine:
In the wwwroot\config.json file, change SitecoreIdentityServerUrl
 - o BizFX:
In the assets\config.json file, change IdentityServerUrl.
 - o Postman:
In the existing environments, for example, the Habitat Environment, update the SitecoreIdServerHost variable