Installation Guide for the XM Scaled Topology

A guide to installing the Sitecore XM scaled topology

June 5, 2024
Sitecore Experience Platform 10.3.X

Table of Contents

1. Choosing a topology

Before you install Sitecore Experience Platform, you must choose the topology or the type of instance that you want to install.

Sitecore supports the following topologies for on-premise installations by default:

- XP Single Developer (XP0)
- XM Scaled (XM1)
- XP Scaled (XP1)

This guide describes how to install the XM Scaled topology.

If you want to install one of the other topologies, download the installation guide for that topology.

You can configure the topology to match your business needs. There are several scalability options that you can use to achieve better performance, cope with greater website demand, and manage large amounts of website traffic. For more information about scaling, see Scaling options.

Sitecore also provides a number of cloud-based offerings. For more information, see our Cloud Services.

> **NOTE**
> If you want to install an XM Single topology, you must install the CM instance from the XM Scaled topology and then in the `web.config` file, specify the following setting:
>
> ```
> <add key="role:define" value="Standalone" />
> ```

1.1. On-premise topology options

The following topologies are available:

Deployment topology	Description
XP Single Developer (XP0)	Use this topology for local development and testing. The Sitecore Experience Platform, runs as three single instances: Sitecore, xConnect, and Sitecore Identity server. The Experience Database (xDB) is partially included in the Sitecore and xConnect instances. **NOTE** For security and scalability reasons, in production environments, it is best practice to use the XM Scaled (XM1) or XP Scaled (XP1) configuration. This option runs all the services on a single server.

Deployment topology	Description
XM Scaled (XM1)	Use this topology if you are not planning to use the analytics and marketing features in the Sitecore Experience Platform. The Sitecore Experience Manager configuration (similar to CMS-only mode) runs the Content Delivery (CD), Content Management (CM) server roles and the Sitecore Identity server. **NOTE** When you select this topology, xDB and xConnect are not available. This option runs the services on one or more servers.
XP Scaled (XP1)	Use this topology if you are planning a fully featured Sitecore Experience Platform installation. The Sitecore Experience Platform configuration runs the following separated server roles: • Content Delivery • Content Management • Email Experience Dedicated Dispatch Server (optional) • Sitecore Identity • Processing • xConnect Collection • xConnect Collection Search • xDB Reference Data • xDB Automation Operations • xDB Automation Reporting • Sitecore Cortex™ Processing Engine • Sitecore Cortex™ Reporting service This option runs the services on one or more servers.

There are different ways to install Experience Manager (XM). You can use:

- Sitecore XM Scaled topology packages.
- Sitecore XP Single or the XP Scaled topology packages and then configure the installation to run in CMS-only mode.
- Sitecore containers.

In a Sitecore Experience Manager installation, the functionality that you can use is different depending on how you installed Sitecore XM.

NOTE
In a scaled environment, you must consider how to configure your session state provider. For more information, see the section Configure session state providers.

NOTE
This document does not describe how to configure the Sitecore Email Experience Manager. For more information, see the the Sitecore Email Campaign Manager documentation.

2. Sitecore Installation Framework

Before you can install Sitecore Experience Platform, you *must* set up the latest Sitecore Installation Framework (SIF).

The Sitecore Installation Framework (SIF) is a Microsoft® PowerShell module that supports local and remote installations of Sitecore Experience Platform.

SIF deploys Web Deploy Packages (WDP) by passing parameters to SIF configuration files through a Microsoft® PowerShell module and is fully extensible.

The Sitecore Experience Platform is designed to be secure-by-default. For developer environments all the required self-signed certificates are created automatically if you do not provide any.

In a production environment, you can provide your own certificates. In a non-production environment, you can choose to have the module generate the certificates for you.

2.1. Customizing the Sitecore Installation Framework

SIF lets you customize your installation within Microsoft PowerShell to add more tasks and features as required. For example, you can add steps to unpack a `.zip` archive of content, download files from other sources, or make a web request to call another service.

For more information about how to extend the installation framework, see the *Customize the Sitecore Installation Framework* section in the latest Sitecore Installation Framework Configuration Guide .

2.2. Run SIF remotely

PowerShell Remoting lets you run SIF configurations on a remote computer.

2.2.1. Enable PowerShell remoting

To enable PowerShell remoting:

- On the remote computer, in a PowerShell command line, run the `Enable-PSRemoting` cmdlet.

NOTE
You must enable PowerShell remoting for the user that completes the installation, and this user must have administrator rights to perform the deployment. For more information about securing or configuring a computer for remote access, see Microsoft's documentation.

SIF uses SSL to create a remote PowerShell session. You must configure Windows Remote Management (WinRM) to work over HTTPS. For more information, see Microsoft Support.

2.2.2. Start a remote installation

To start a remote deployment:

1. Install SIF on the remote computer.
2. In a PowerShell command line, create a new remote session:

```
$session = New-PSSession -ComputerName <RemoteComputerName>
```

3. To copy all the required packages and SIF configuration files to the remote computer, specify the path and then run the following cmdlet:

```
Copy-Item -Path <sourcefile> -Destination -<remotePath> -ToSession $session
```

4. To start the installation, run the following cmdlet:

```
$session = New-PSSession -ComputerName  <RemoteComputerName>

Invoke-Command -Session $session { Import-Module SitecoreInstallFramework }

Invoke-Command -Session $session { Install-SitecoreConfiguration -Path
<configurationpath> }
```

NOTE
For more information about the `Invoke-Command` cmdlet, see the PowerShell documentation.

2.3. Install a SIF configuration file

Here is an example of how to use SIF to install a SIF configuration file on a local server.

To install a SIF configuration file on a local instance:

1. Launch PowerShell as an administrator.
2. Set up the latest Sitecore Installation Framework.

3. To start the installation, run the `Install-SitecoreConfiguration` cmdlet, and specify the path to your SIF configuration file.
   For example, using the `sitecore-XM1.json` file:

   ```
   Install-SitecoreConfiguration -Path <configurationpath>\sitecore-XM1.json
   ```

   Optionally, the parameters declared in the SIF configuration files can be passed in at the command line by prefixing their name with a dash "-". For example:

   ```
   Install-SitecoreConfiguration -Path <configurationpath>\sitecore-XM1.json -SqlDbPrefix SC.
   ```

   In a PowerShell command line, you can pass additional parameters to control the installation process. For example:

Cmdlet	Description
`-Verbose`	Increases the amount of information that is logged.
`-Skip <taskname>`	Skips one or more tasks.

   For more information about the parameters that can be passed to the `Install-SitecoreConfiguration` cmdlet, run the following cmdlet:

   ```
   Get-Help Install-SitecoreConfiguration
   ```

NOTE
You can also use the `scinst` alias to run the `Install-SitecoreConfiguration` cmdlet.

3. Installation requirements

Before you can install Sitecore Experience Platform, you must ensure that you have all of the server and client requirements and prerequisites in place.

3.1. Client requirements

3.1.1. Browser requirements

Sitecore Experience Platform clients are browser-based user interfaces. Sitecore Experience Platform has been tested and can run on the following browsers:

- Mozilla Firefox
- Google Chrome
- Microsoft Edge
- Apple Safari (Mac only)

> **NOTE**
> Sitecore Experience Platform supports all the current stable versions of these browsers.
>
> Although Sitecore Experience Platform supports the tested versions of the listed browsers, newer browser revisions are continually released. Sitecore will support the latest revisions of these browsers.

For more information about browser compatibility, see the Sitecore compatibility table.

3.1.2. Client hardware requirements

Sitecore Experience Platform has the following minimum client hardware requirements:

- Processor: Intel Pentium 4, 2 GHz or faster
- RAM: 512 MB minimum, 1 GB – recommended
- TCP/IP connection at 512 Kbps or faster to the Sitecore XP host
- 1024 x 768 or greater screen resolution required for advanced operations

In general, the client computer should meet the hardware requirements of the browser that you use.

You do not have to install any additional software on the Sitecore Experience Platform clients that access Sitecore Experience Platform servers.

3.2. Server requirements

Before installing Sitecore Experience Platform, you must ensure that you have all of the requirements and prerequisites in place.

3.2.1. Hardware requirements for a server running a single Sitecore installation

To run a single Sitecore installation, the minimum configuration requirements are:

- 4 core processor
- 16 GB of RAM

> **NOTE**
> The recommended hardware requirements are for running the software on a single computer. For more information about running Sitecore on different kinds of hardware, consult your Sitecore partner or technical sales representative.

3.2.2. IIS requirements

You must use the version of IIS that your operating system supports. For more information about IIS and operating systems, see Microsoft's documentation.

Sitecore Experience Platform does not officially support any other ASP.NET web servers such as IIS Express, or Mono Web Server.

Sitecore Experience Platform neither supports nor allows multiple IIS website definitions to point to the same Sitecore web root.

3.2.3. Operating system requirements

Sitecore Experience Platform is only compatible with the client and server operating systems that support .NET Framework 4.8.0.

Sitecore Experience Platform can be hosted on the following Microsoft operating systems:

- Windows Server 2022
- Windows Server 2019
- Windows 11 (64-bit)
- Windows 10 (64-bit)

IMPORTANT
If the Transport Layer Security (TLS) protocol version 1.2 is not activated by default, you must enable it on all of your Sitecore Experience Platform content management and Dedicated Dispatch servers (DDS).

For Windows 10, TLS 1.2 is enabled by default.

You can check that it is enabled in the registry - `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Client`.

For Windows 11 and Windows Server 2022, TLS protocol version 1.3 is enabled by default.

You must disable the TLS 1.3 on all of your Sitecore Experience Platform content management and Dedicated Dispatch servers(DDS) and enable TLS 1.2.

For more information about enabling TLS 1.2, see Microsoft's documentation.

IMPORTANT
Run Windows Update and install all the appropriate service packs and security updates on all of your Sitecore Experience Platform server and client computers.

3.2.4. .NET requirements

Sitecore Experience Platform requires .NET Framework 4.8.0.

Sitecore Identity server requires the latest .NET Core 6.0 Windows Hosting Bundle.

You must apply any available updates to the .NET Framework on every Sitecore installation.

3.2.5. Microsoft Visual C++ 2015 redistributable requirements

Sitecore Experience Platform 9.0 Update-1 introduced a new prerequisite for the Microsoft Visual C++ 2015 Redistributable. For more information, see Microsoft's documentation.

NOTE
This redistributable may already be installed with Microsoft Windows. Without it, Sitecore Experience Platform will fail to start up with the message:

Could not load file or assembly 'ChilkatDotNet46.dll' or one of its dependencies. The specified module could not be found.

3.2.6. Database requirements

Sitecore Experience Platform supports the following database servers:

- Microsoft SQL Server 2017
- Microsoft SQL Server 2019
- Microsoft Azure SQL
  You must create the Azure SQL instance in advance and pass the address and the server administrator credentials in the corresponding parameters when you run the installation.

For more information, see the Sitecore XP documentation.

3.2.7. Enable Contained Database Authentication

When you use Web Deploy Packages, you must ensure that the target SQL Server is configured to allow users and logins to be contained at the database level.

To configure the target SQL Server:

1. Launch Microsoft SQL Server Management Studio and log in as an administrator.
2. Run the following new query:

```
EXEC sp_configure 'contained', 1;
RECONFIGURE;
```

> **NOTE**
> For more information about the contained database authentication option, see Microsoft's documentation.

3.2.8. Search indexing requirements

Sitecore Experience Platform supports Solr 8.11.2. Solr is the default search provider and supports both content search and analytics search. To use the Sitecore Installation Framework (SIF), you must have Solr installed.

> **NOTE**
> Support for Lucene was removed in Sitecore XP 9.3.0.
>
> Support for Azure Search was removed in Sitecore XP 10.2.0.

For more information about how to install and manage these index providers in Sitecore Experience Platform, see the Sitecore documentation.

3.2.9. Installing Solr

> **NOTE**
> If you plan to use the Sitecore Installation Assistant to install Sitecore XP, it can install Solr for you and you can skip this section.
>
> If you plan to use the Sitecore Installation Framework (SIF) to install Sitecore XP, you must follow the steps described in this section.

The standard Sitecore Experience Platform configuration requires Solr. The Sitecore Experience Platform is secure by default, you must, therefore, enable SSL for Solr.

Before you run the SIF, you must:

- Install Solr and configure it to run as a Windows service.
- Enable and set up SSL for Solr.

The `Solr-Singledeveloper.json` deployment configuration file installs and configures Solr with SSL.

For local testing and development, you can set up a self-signed certificate. The Apache Solr Reference guide has more information about creating a self-signed certificate.

For more information about installing Solr, see the section Install a search provider in a scaled solution.

3.2.10. Antivirus software considerations

Some antivirus software can have a detrimental effect on the performance of ASP.NET applications, including Sitecore. We recommend that you use only antivirus scanners that are certified for your operating system.

For more information about the certified products, see the Windows Server Catalog website.

For optimal performance, ensure that the following folders are *not* scanned by your antivirus software:

- The site root folder
- The data folder defined in the `web.config` file
- The folder that contains the actual Sitecore database files
- The `C:\Windows\Temp or {app_pool user profile}\Temp` folder

> **NOTE**
> Active file scans from antivirus tools can significantly impact the performance of search indexing software. This can lead to poor user experience or slow system performance. Consider turning off any antivirus tools or modifying antivirus settings on the search index server to exclude the application data folders from scans. For more information about your search indexing software, consult the related documentation.

3.2.11. Prerequisites for using the Sitecore Installation Framework

To use the Sitecore Installation Framework to install Sitecore Experience Platform in an on-premise environment, you must download and install:

- Windows PowerShell® version 5.1

3.3. Server file system requirements

Before you install Sitecore Experience Platform, you must fulfill all the requirements.

3.3.1. File system permissions for ASP.NET requests

Sitecore Experience Platform executes requests for ASP.NET resources and all the .NET code running within the application with the permissions of the account configured as an identity for the website's application pool. This account requires Modify permissions for all the files, folders, and subfolders under the `\wwwroot\<YourWebsiteFolder>` folder.

> **NOTE**
> The Sitecore Installation Framework automatically sets all the required permissions to your website folder. If you deploy Sitecore through a manual configuration, such as a PowerShell script or similar, you must set the correct file system permissions.

The default account that is used to process ASP.NET requests in the different versions of IIS:

IIS version	Default ASP.NET account name
10	ApplicationPoolIdentity

If you select a different user account to process the ASP.NET requests, you must also grant this account the *Modify* permissions.

For more information about application pool identities and specifically about assigning rights to the *AppPoolIdentity* account, see Microsoft's documentation.

3.3.2. File system permissions for system folders

To load the .NET runtime and ASP.NET resources that are used to process the ASP.NET requests, the worker process that hosts the Sitecore Experience Platform application requires access to multiple system files and folders that are not distributed as a part of Sitecore Experience Platform, but are installed as a part of the Windows Operating System and the .NET framework. For more information about built-in groups and accounts in IIS, see Microsoft's documentation.

Most of these permissions are granted by IIS to all ASP.NET applications, automatically making the application pool identity account a member of the *IIS_IUSRS* security group.

However, in certain environments, you must manually grant permissions for the Application Pool Identity to the following system locations:

Default location	Required permissions	Comments
`%WINDIR%\temp\`	Modify	To install Sitecore Experience Platform, you must assign the *Modify* access rights to the `\temp` folder for the ASP.NET and/or IUSR accounts.
`%WINDIR%\Globalization\`	Modify	Required for registering custom languages by the .NET Framework correctly.
`%PROGRAMDATA%\Microsoft\Crypto`	Modify	Required for storing cryptographic keys used for encrypting/decrypting data.

These variables have the following default values:

Variable	Default value
`%WINDIR%`	`C:\Windows`
`%PROGRAMDATA%`	`C:\ProgramData` for IIS 10 and later

> **NOTE**
> The Sitecore Installation Framework specifies the required permissions for certificates under the `\Crypto` folder.

3.3.3. UNC share is not supported

You must install Sitecore Experience Platform on a local drive, not a Universal Naming Convention share.

3.3.4. Sitecore cannot operate from a virtual directory

Sitecore Experience Platform does not support operating from a virtual directory.

4. Install the prerequisites

You must install the prerequisites. You can install them automatically or manually.

4.1. Automated installation of prerequisites

A predefined SIF configuration file `Prerequisites.json` is distributed in the configuration packages. This file downloads and installs most of the prerequisites.

This file does not install:

- Microsoft SQL Server
- Solr

For more information about how to use this file, see the section Install a SIF configuration file.

> **NOTE**
> SIF does not install any of the prerequisite software if it is already installed. You must ensure that you install the correct versions.

4.2. Manual installation of the prerequisites

To install most of the Sitecore server roles, you must have the following prerequisites:

- IIS, ASP.NET 4.8, and the Web administration PowerShell Module
- SQL PowerShell Module

The Sitecore server roles require:

Requirement	Feature	Details
WebAdministration module	Supports IIS management.	When you configure a computer with IIS, the *WebAdministration* module is installed automatically.
Web Deploy 3.6 for Hosting Servers	Supports the installation of Web Deploy Packages.	Download and install Web Deploy v3.6.
URL Rewrite 2.1	Supports URL rewrites for Sitecore when installed as a Web Deploy Packages.	Download and install URL Rewrite 2.1.

Requirement	Feature	Details
Microsoft SQL Server Data-Tier Application Framework (DacFx) version 2017 (x86 and x64)	Supports the installation of .dac files in Web Deploy Packages	Download and install DacFx x86. Download and install DacFX x64. This must be installed on servers that have been assigned a Sitecore server role and where you are going to install DAC packages: In the XM Scaled topology: • Content Management To ensure that DacFx works correctly, you must install its system requirements including Microsoft System CLR Types for SQL Server 2017. If you are running an x64 environment, you must install both the x64 and x86 versions of DacFx and SQLSysCLRTypes. **NOTE** If DACFx fails to install, you can see the following error message when you use the framework: *The SQL provider cannot run with dacpac option because of a missing dependency. Please make sure that DACFx is installed.* For information about how to resolve this error, see this Sitecore Knowledge Base article.
Microsoft ODBC Driver 13 for SQL Server Microsoft Command Line Utilities 13 for SQL Server	Required by the Sitecore Installation Framework	You must also install these utilities on the xConnect application server before running the Sitecore Installation Framework installation template for xConnect or the Single Developer workstation.

NOTE
If you don't have SQL Server installed, you need to install `SharedManagementObjects.msi` and its dependency `SQLSysClrTypes.msi`. They both can be taken from Microsoft® SQL Server® 2016 Service Pack Feature Pack. It is required for the SQL Server database deployments used by Web Deploy.

4.3. Install a search provider in a scaled solution

Sitecore Experience Platform supports Solr as the search provider for both content search and analytics search.

- In PaaS solutions, you can use Solr, or SolrCloud in Azure.

- In on-prem solutions, you can use Solr or SolrCloud. In on-premise solutions, the Sitecore Installation Framework requires Solr to deploy.

> **NOTE**
> Support for Lucene was removed in Sitecore Experience Platform 9.3.0.
>
> Support for Azure Search was removed in Sitecore Experience Platform 10.2.0.

For local testing and development, you can set up a self-signed certificate. For more information about creating a self-signed certificate, see the Apache Solr Reference guide.

4.3.1. Install the Solr Certificate

You must install the Solr certificate on the servers that perform the following roles:

- Content Management

5. Set up a production environment

This chapter describes how to install Sitecore Experience Platform XM scaled topology for production.

5.1. Set up the certificates

Sitecore Experience Platform is designed to be secure by default. You must therefore implement HTTPS across the platform.

Server Certificate Authentication

All communication between Sitecore instances occurs over the default HTTPS configuration. This includes the Sitecore Identity server, and the Solr search provider. HTTPS requires that you obtain and set up certificates for the Secure Sockets Layer (SSL) before you install the platform.

Server authentication uses a server-side certificate and a private key to encrypt traffic between the HTTP client and the HTTP server application. This type of authentication prevents unencrypted content from traveling over an unsecured network. It does not identify who the client is and the server authentication alone does not determine who can connect to the server.

> **IMPORTANT**
> In local developer environments, self-signed certificates can be used to develop Sitecore solutions. Due to potential security concerns, you must not use self-signed certificates in production environments.

5.1.1. Set up server certificate SSL authentication on IIS

You must obtain and install the server certificates before you run SIF. For more information about how to set up SSL in IIS, see Microsoft's documentation.

The following table lists the full set of server authentication certificates for this topology:

XM Scaled (XM1)
Content Management
Sitecore Identity server

For each certificate, you must use the site name in the common name **CN** field in the certificate. For example, if the name that you want to use for the Content Management IIS site is *CM_test*, you must use this name when you create the Content Management certificate.

> **NOTE**
> Starting with Sitecore 10.0.1, the Content Delivery server role comes with the HTTPS protocol enabled by default.
>
> If you need to deploy a CD server role that uses the HTTP protocol, you can change the protocol. To use the HTTP protocol, before you deploy the CD server role, in the `sitecore-XP1-cd.json` file, set the *DisableHttpsForCD* parameter to *true*.

Install the server certificates

After you obtain the relevant certificates, you must install them.

To install the server certificates:

1. Install the server authentication certificate in the system certificate store folder:

   ```
   Certificates (Local Computer)\Personal
   ```

   For information about how to install a private key certificates, see PowerShell Import-Certificate from Microsoft.

2. If you created a self-signed certificate, install the self-signed authority certificate for the SSL certificate in the following folder:

   ```
   Certificates (Local Computer)\Trusted Root Certification Authorities
   ```

> **NOTE**
> For the XM Scaled topology, it is assumed that there is only one SSL certificate for each IIS instance that covers multiple application roles. For XM Scaled (XM1), there is a dedicated role per server in a distributed setup, and you must obtain and install a certificate for each server role.

5.2. Install the Sitecore XM scaled topology

Once you have obtained the required certificates, you can run SIF and install the Sitecore XM scaled topology. You can install any of the configurations for dedicated server roles, on single or multiple servers.

The server roles are defined as a part of your desired scaling configuration.

> **IMPORTANT**
> You must first install the `sitecore-solr.json` deployment configuration on your Solr search server. Then you must install the rest of the Sitecore server roles deployment configurations.

5.2.1. Use SIF to install the Sitecore XM Scaled topology

To run SIF and install the Sitecore XM Scaled topology:

1. If you have not already done so, as an administrator, in a PowerShell command line, run the following cmdlet:

   ```
   Import-Module SitecoreInstallFramework
   ```

2. Download the OnPrem XM Scaled WDP archive and unzip to a folder of your choice.

3. To install the Solr cores, run the following cmdlets with the required parameters for your server roles:

```
Install-SitecoreConfiguration –Path
"C:\SitecoreInstaller\Configurations\Platform\Solr\sitecore-solr.json"
```

4. To install the server roles, run the following cmdlets with the required parameters for your server roles:

```
Install-SitecoreConfiguration –Path
"C:\SitecoreInstaller\Configurations\IdentityServer\IdentityServer.json"

Install-SitecoreConfiguration –Path
"C:\SitecoreInstaller\Configurations\Platform\XM1\sitecore-XM1-cm.json"

Install-SitecoreConfiguration –Path
"C:\SitecoreInstaller\Configurations\Platform\XM1\sitecore-XM1-cd.json"
```

Specify the certificates during installation

To install Sitecore with your pre-installed certificates, when you run the `Install-SitecoreConfiguration` cmdlet, you must provide the certificates as parameters.

SIF searches for the certificates in the following path, by default:

```
Cert:\Localmachine\My
```

You can change the storage location.

Change the default location of the certificates

To change the default location of the certificates used for the deployment:

- In a text editor, open the relevant `.json` file, and in the `Variables` section, change the default store value:

```
"Security.CertificateStore": "Cert:\\Localmachine\\My"
```

Specify the names or thumbprints of the certificates

You must specify the names or thumbprints of the certificates that you created and installed earlier in this guide as parameters.

- For the SSL authentication certificate, for example, of an instance with the name `"CM_test"`:

```
-SSLCert "CM_test"
```

or

```
-SSLCert "2205A94867EE99E3B29EA7A9AC5A7646D43FD88B"
```

> **NOTE**
> In the PowerShell command line parameter, you must specify the client certificate thumbprint in capital letters.

Skip database deployment when you install a server role

When you install a server role, you do not have to install the databases when you deploy the WDP packages.

A new the *SkipDatabaseInstallation* parameter has been added to the following deployment configuration file:

- `sitecore-XM1-cm.json`

The *SkipDatabaseInstallation* parameter is set to *false* by default and all the databases are installed when you deploy the WDP packages.

> **NOTE**
> The *SkipDatabaseInstallation* parameter is not supported by the SingleDeveloper or the other XM Scaled deployment configuration files.

To deploy a Sitecore server role without installing the databases:

1. Ensure that you already have already installed the databases that you need for the server role.
2. Ensure that the database names contain the prefixes that you are going to use for this deployment.
3. In the Powershell deployment command, enter all the required parameters.
4. In the `c:\resourcefiles\Role-Remote.json` file, ensure that the `SIFVersion` parameter matches the latest version of SIF.
5. In the Powershell deployment command, enter following parameter in the corresponding config files:
   - `SkipDatabaseInstallation: true`
6. Pass the database user names and passwords, for example:

```
"SqlCoreUser" : "mycoreuser"

"SqlCorePassword" : "mycorepassword"
```

Install to a custom website folder

Since Sitecore XP 10.0.0, you can specify the folder where the website is installed.

A new *SitePhysicalRoot* parameter has been added to the deployment configuration files for each server role.

When you set this parameter, the website is installed in the *[SitePhysicalRoot]\[SiteName]* folder.

If you leave this parameter with the default value, the website is installed in the default IIS *wwwroot* folder.

> **NOTE**
> This parameter is supported by the SingleDeveloper and the Distributed deployment configuration files and the *SitePhysicalRoot* path that you provide is created for every website.

5.3. Configure the parameters in the SIF configuration files

The SIF configuration files are templates that are the basis for deploying various Sitecore Experience Platform configurations with support for:

- Creating an IIS Application Pool.
- Creating an IIS website.
- Installing Web Deploy Packages.
- Configuring File Permissions.

You must review and configure the default parameters in each of the SIF configuration files for your topology.

To configure the parameters in a SIF configuration file:

1. In a text editor, open the relevant SIF configuration file, for example `sitecore-solr.json`, and find the `Parameters` section.
2. Check whether you need to change the default value of each parameter, if it has one. If there is no default value, consider if you want to add one. When you run the installation, any parameter that does not have a default value prompts you for that information.
   The configuration files contain a description for each parameter.
   The following screenshot shows a snippet of a `Parameters` block from a configuration file:

```
{
    "Parameters": {
        "Package": {
            "Type": "string",
            "Description": "The path to the Web Deploy package to deploy.",
            "DefaultValue": ""
        },
        "SqlDbPrefix": {
            "Type": "string",
            "Description": "The prefix used for all Sql databases.",
            "DefaultValue": ""
        },
        "SitecoreIdentityCert": {
            "Type": "string",
            "Description": "The certificate to use for encryption. Provide the name or the thumbprint.",
            "DefaultValue": ""
        },
        "LicenseFile": {
            "Type": "string",
            "Description": "The path to the Sitecore license file.",
            "DefaultValue": ".\\License.xml"
        },
        "SiteName": {
            "Type": "string",
            "DefaultValue": "IdentityServer",
            "Description": "The name of the site to be deployed."
        },
        "SqlCoreUser": {
            "Type": "string",
            "DefaultValue": "coreuser",
            "Description": "The user to create and use in Core connection string."
```

5.4. Distributed installation script for the Sitecore XM Scaled topology

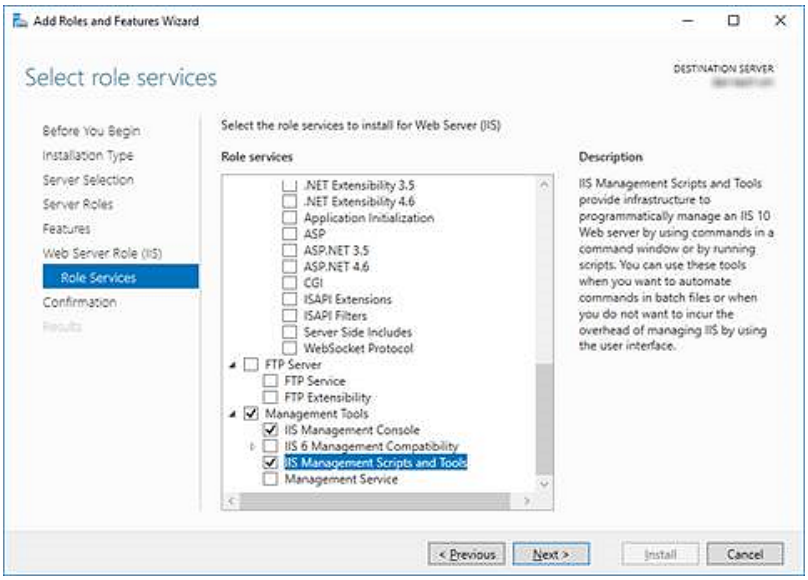
To simplify your scaled installation on multiple servers, you can use a PowerShell script to install the Sitecore XM Scaled topology. For more information, see Architecture overview in Sitecore documentation.

> **NOTE**
> If you use this script to install the Sitecore XM Scaled topology, each server role must be installed on a separate computer and there must be no previous Sitecore installation on any of these computers.

5.4.1. Distributed installation script prerequisites

To prepare the servers, you must perform the following steps on each server:

1. Enable PowerShell Remoting.
   For more information about PowerShell Remoting, see the section *Run SIF Remotely*.
2. Install all the prerequisites.
   For more information about the prerequisites, see the section *Install the Prerequisites*.
3. Create the `c:\resourcefiles` folder on every machine that will be assigned a Sitecore server role or will run Solr.
4. Configure the Web Server Role (IIS).



5.4.2. Run the distributed installation script for the XM Scaled topology

To edit and run the XM topology installation script:

1. Create a folder called `c:\resourcefiles`.
2. On the Sitecore Experience Platform download site, download the Packages for XM Scaled topology. This file is listed in the Download options for On Premises deployment section. Extract to `c:\resourcefiles` folder all `scwdp.zip` files and all configuration files.

3. Save your Sitecore license file in the `c:\resourcefiles` folder as `license.xml`.
4. In the `c:\resourcefiles` folder, edit the `XM1-Distributed.ps1` file script and update each line with the appropriate settings for your environment.
5. In a PowerShell command line, navigate to the `c:\resourcefiles` folder and run the following command:

```
.\XM1-Distributed.ps1
```

After you have edited and run the installation script, you must complete the post-installation steps described in the chapter *Post-installation steps*.

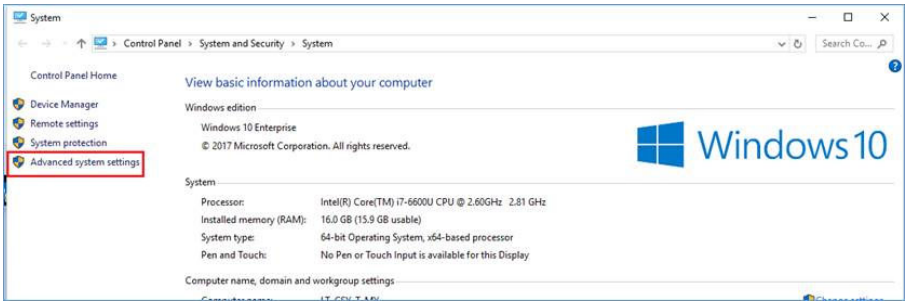
5.5. Enable telemetry reporting in production

The Sitecore telemetry reporting feature gathers information that helps Sitecore understand how customers use our products. The environment type (production or non-production) helps us to associate the features used with the appropriate use-cases.

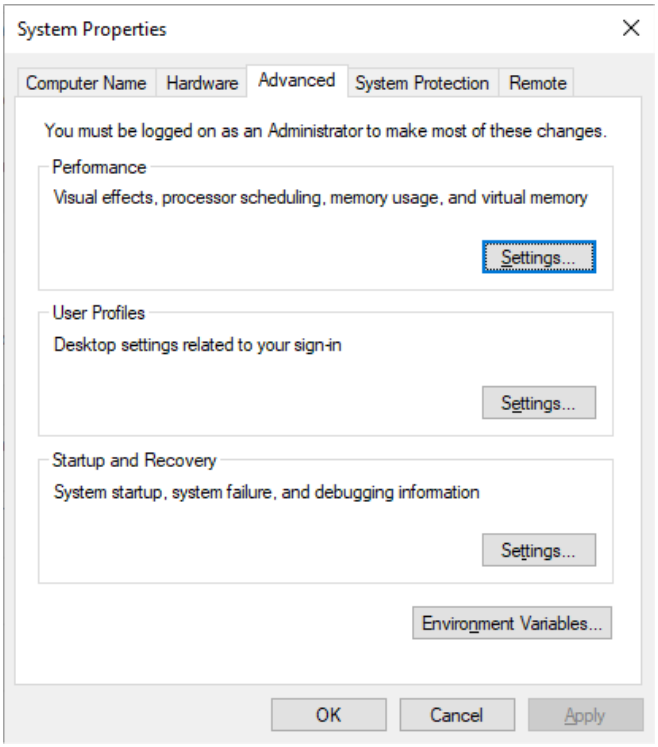
The telemetry system assumes Sitecore Experience Platform is running in a non-production environment by default. When you set up a production environment, you must therefore define a system variable to report the environment type.

To define a system variable:

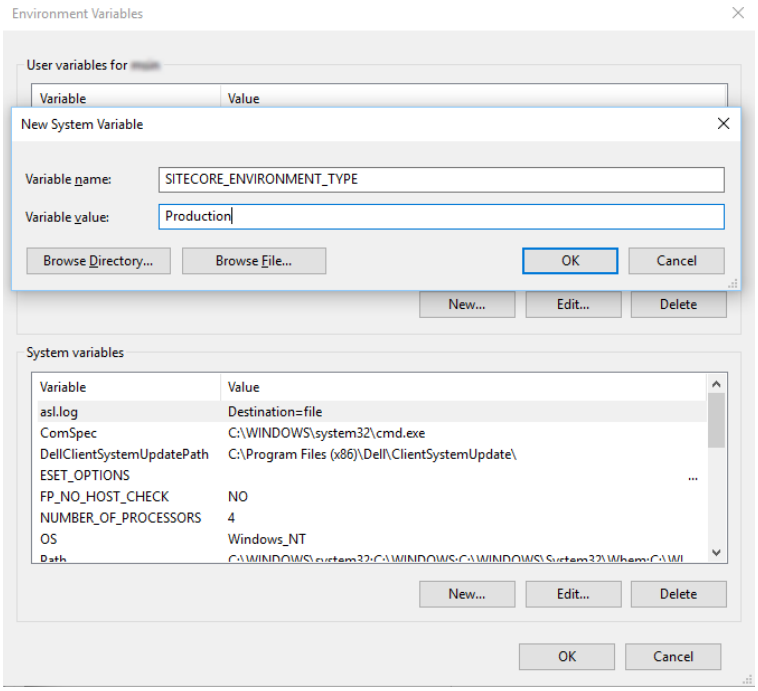
1. In the **Control Panel**, click **System**, **Advanced system settings**.



2. In the **System Properties** dialog, click **Environment Variables**.



3. In the **Environment Variables** dialog, in the **System variables** section, click **New**.
4. In the **New System Variable** dialog, in the **Variable name** field, enter SITECORE_ENVIRONMENT_TYPE and in the **Variable value** field, enter *Production*.



5. Restart the computer.

6. Post-installation steps

After you use SIF to install Sitecore XP, you must perform the following post-installation steps.

6.1. Configuring Sitecore Identity server

The Sitecore Identity server only works with HTTPS, and you must generate a certificate for it.

The Sitecore Identity server configuration requires the following additional parameters:

- `allowedCorsOrigins` – a pipe-separated list of instances (URLs) that are allowed to login via Sitecore Identity. This can be a Sitecore instance in the XP0 topology, or all the CM/CD servers in a scaled environment.
- `ClientSecret` – a random string value that must be identical on both the client and server side.
  - On the client side, it is stored in the connection strings on the CM server – `sitecoreidentity.secret`.
  - On the server side, it is stored in the `<IdentityServer folder>\ Config\production\Sitecore.IdentityServer.Host.xml` file, in the `ClientSecrets` node.
- `PasswordRecoveryUrl` – the client URL (CM server).
  If a user forgets their password, they are redirected to the appropriate Sitecore server to fill in the form for password recovery.

You must also register the Identity server on the client side. The Identity server is configured in the `\App_Config\Sitecore\Owin.Authentication.IdentityServer\Sitecore.Owin.Authentication.IdentityServer.config` configuration file `–sc.variable "identityServerAuthority"`.

For more information see Sitecore Identity documentation.

6.2. Rebuild the search indexes and the Link database

After you install Sitecore Experience Platform, you must rebuild the search indexes and rebuild the *Link* databases.

To rebuild all the indexes:

1. On a Sitecore CM or Index Manager instance, open the Sitecore Launchpad, click **Control Panel**, and in the **Indexing** section, click **Indexing manager**.

2. In the **Indexing Manager** dialog box, click **Select all**, and then click **Rebuild**.

To rebuild the Link databases for the *Master* and *Core* databases:

1. On the **Sitecore Launchpad**, click **Control Panel**, and in the **Database** section, click **Rebuild Link Databases**.
2. Select the *Master* and *Core* databases and then click **Rebuild**.

6.3. Content expiration

Microsoft IIS uses the Expire Web content header (located in common HTTP Response Headers) to determine whether to return a new version of the requested web page if the request is made after the web page content has expired. IIS marks each web page before it is sent, using the settings that you provide for content expiration. The website visitor's browser translates the expiration mark. You can set the IIS Expire Web content header to improve performance.

By setting Expire Web content to something other than *immediately*, you can reduce second-access load times by 50 –70%. This setting does not affect dynamically-generated content.

For more information about content expiration, see Client Cache in the Microsoft IIS documentation.

6.4. Configuring session state providers

In the Sitecore Experience Platform, configuring session state is particularly important if you have deployed a multi-server, fully scalable environment with clusters of content delivery servers.

Sitecore is deployed with an *InProc* session state provider by default but we recommend that you use *OutOfProc* session state providers if you deploy more than one CD server.

Sitecore Experience Platform supports the following session state providers:

- Microsoft SQL Server
- Redis

To configure any *OutOfProc* session state providers, see the Sitecore documentation.

6.5. Warm up the servers

To ensure that your Sitecore websites are available at all times, even after restarting a server, you must enable the IIS auto-start feature for the application pools on all the servers that you have configured.

For more information about the auto-start feature, see Microsoft's documentation.

6.6. Security hardening

We recommend that you follow all the security hardening instructions described in our documentation. In addition, the way you implement your Sitecore solution has a significant effect on the security of your website and it might require additional security-related coding and configuration.

For more information about security hardening, see the Security Guide.

6.7. Import the client translations

The user interface in Sitecore Experience Platform are only available in English by default. If you want to use another language, you must import the client translation. Download the latest translations from the Client Translations section of the .

Client translations are available in:

- Danish (da-DK)
- German (de-DE)
- Japanese (ja-JP)
- Chinese (zh-CN)

For more information about how to import a client translation, see the Sitecore Experience Platform Client Translations.

7. Uninstall the Sitecore XM Scaled topology

This chapter describes how to uninstall Sitecore XM Scaled topology.

7.1. Uninstall a Sitecore instance using SIF

You can use SIF to uninstall a Sitecore instance/role from a local server.

To uninstall a Sitecore instance:

1. Launch PowerShell as an administrator.
2. Run the `Uninstall-SitecoreConfiguration` cmdlet, and specify the path to your SIF configuration file.
   For example, using the `sitecore-XM1-cm.json` file:

   ```
   Uninstall-SitecoreConfiguration -Path <configurationpath>\sitecore-XM1-cm.json
   ```

   Alternatively, you can pass in the parameters declared in the SIF configuration files by prefixing their name with a dash "–" in the command line.
   For example:

   ```
   Uninstall-SitecoreConfiguration -Path <configurationpath>\sitecore-XM1-cm.json
   -SqlDbPrefix SC.
   ```

   In a PowerShell command line, you can pass additional parameters to control the uninstall process.
   For example, running the `Verbose` cmdlet increases the amount of information that is logged, and the `-Skip <taskname>` cmdlet skips one or more tasks.

To correctly uninstall a SIF configuration, you must pass the same parameters that were used during the installation.

The uninstall is performed by a separate list of tasks within the configuration file. For more information, see the SIF documentation.

7.2. Uninstalling the XM Scaled topology in a distributed environment

Centralized uninstallation of distributed deployments is not supported yet. If you need to re-install a scaled deployment you must remove everything that was installed during the previous installation from each individual computer.

8. Appendix

This chapter contains answers to some issues that can arise during the installation as well as some supplementary instructions that help you configure your environment, such as, file permissions, performance counters, and certificates.

8.1. Common issues

I get a 403.16 Forbidden error

- Check that your root certificate is in the Trusted Root Certificates Authority store of the *Local Computer*, not the current user and that the *Issued To* and *Issued By* properties of the root certificate match.
- Ensure you imported the certificates into the *Local Computer's* certificate store, not the current user's certificate store.
- Ensure the certificate you created in IIS has a name that matches the site.
  For example, `CM_test`.
- Ensure you pasted your thumbprint into a PowerShell command line window, and that you removed the hidden character at the start of the string.
- Ensure your thumbprint is in uppercase letters.
  For example: `3D703B5198D6D3CEE1D0C1B1BC9ECB6D34989BA4`.
  You can find the thumbprint in the following locations:
  - `Sitecore\App_Config\ConnectionStrings.config`
  - `XConnect\App_Config\ConnectionStrings.config`
  - `XConnect\App_Data\jobs\continuous\AutomationEngine\App_Config\ConnectionStrings.config`
- Ensure the self-signed certificate you created in IIS has the same name as your xConnect instance.
- Ensure the client authentication certificate (under the local machine's *Personal* store) has *read* permissions for the *IIS_IUSR* group and the *NETWORK SERVICE* group.

My Solr index is empty

If you have an error that says your remote certificate is invalid according to the validation procedure, ensure that the indexer's `ConnectionStrings.config` file is using `localhost` rather than `127.0.0.1` for the Solr core URL.

Microsoft.SqlServer.TransactSql.ScriptDom.dll is not installed

If this happens, you can see the following error: *The SQL provider cannot run with dacpac option because of a missing dependency.* To resolve this issue, see this Knowledge Base article.

My Sitecore installation failed while I was using Skype

If you use Skype while you are installing Sitecore Experience Platform, it is possible that your xConnect installation may fail. This occurs because Skype and Sitecore xConnect both use port 443, which interferes with the installation.

If this happens, change your Skype configuration as described in Microsoft's documentation.

Failed installations

If an installation fails for any reason, you must clean up the partial installation before attempting another installation.

To clean up a partial installation, run the configurations again with the `Uninstall-SitecoreConfiguration` command with the same parameters as you used for the installation.

The `createcert.json` configuration file does not contain uninstallation tasks that remove certificates because it is highly likely that the certificates, particularly the root certificates, are used elsewhere.

To remove an incorrect certificate:

1. To open the **Certificate Management** console, in the Windows command prompt, enter `certlm.msc` and press **Enter**.
   To open the **Certificate Management** console for the Current User, enter `certmgr.msc`.
2. In the left pane, in the tree, expand the *Personal* node and select **Certificates**.
3. Select the incorrect certificate, right click it and then click **Delete**.
4. To remove the Root certificates, in the console, in the left hand pane, expand *Certificates*, *Trusted Root Certification Authorities*, *Certificates* and select the incorrect certificate, right click it and then click **Delete**.

After you have removed the failed installation, correct the errors in the launch script or configuration files before attempting a new installation.

The Indexing Manager shows no results after rebuilding the indexes

This can happen if the Solr managed schema is not populated properly during deployment.

To solve this problem, re-populate the Solr schema.

My Solr Schema is not populated

To populate the Solr schema:

1. On the **Sitecore Launchpad**, click **Control Panel**, and in the **Indexing** section, click **Populate Solr Managed Schema**.
2. In the **Schema Populate** dialog box, click **Select all**, and then click **Populate**.

8.2. Access rights

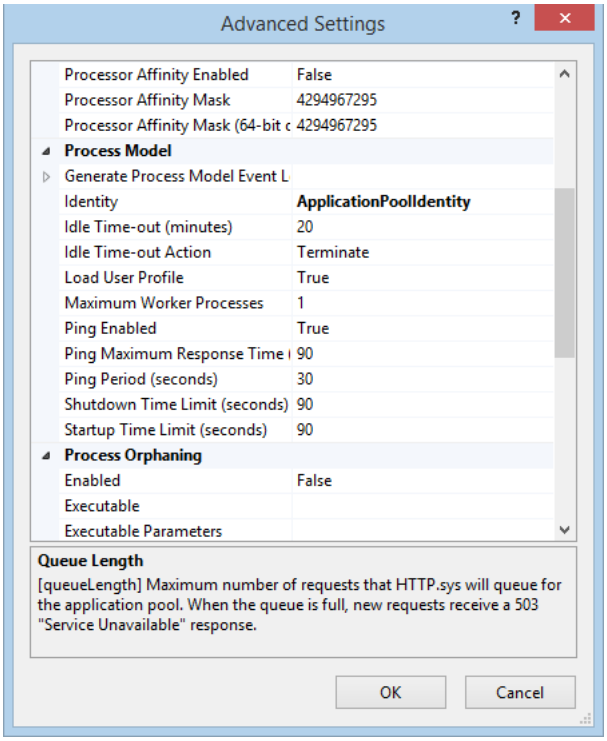
8.2.1. Use Windows Authentication with SQL Server

You can configure Sitecore to use Windows Authentication for a SQL connection and remove the user name and password from the `connectionStrings.config` file.

> **NOTE**
> This only applies to the *Core*, *Master*, *Web*, and *Reporting* SQL databases, and not to xDB and xConnect.

To configure Sitecore to use Windows Authentication:

1. In Windows, launch the IIS Manager.
2. Select the application pool that Sitecore is running under, click **Advanced Settings** and in the **Identity** field, set the identity to the domain user.



3. In SQL Server, register the domain user and grant the appropriate security permissions to the Sitecore databases for the domain user.
4. On the computer that hosts Sitecore Experience Platform, add the domain user to the *IIS_IUSRS* group.
   For more information about changing the permissions for the *IIS_IUSRS* group, see the section Server file system requirements.
5. In a text editor, edit the `\App_Config\ConnectionStrings.config` file and replace the user id and password parameters with `trusted_connection=Yes`.

```
<?xml version="1.0" encoding="utf-8"?>
<connectionStrings>
<add name="core" connectionString="Data
Source=.\sql2016;Database=sc9_Core;Trusted_Connection=True"
```

```
/>
<add name="master" connectionString="Data
Source=.\sql2016;Database=Sandbox6_Master;Trusted_Connection=True"
/>
<add name="web" connectionString="Data
Source=.\sql2016;Database=Sandbox6_Web;Trusted_Connection=True"
/>
</connectionStrings>
```

6. Prepare your identity so that it can be used as a service account with the `aspnet_regiis.exe` file and the `-ga` switch.

8.2.2. Use Windows performance counters

Sitecore Experience Platform contains built-in functionality that reads and updates the Windows performance counters that you can use to monitor and troubleshoot the Sitecore application. This functionality requires access to Windows registry keys. You can grant access by making the application pool identity a member of the built-in *Performance Monitor Users* group.

For more information, see Microsoft's documentation about Application Pool Identity.

If the required registry permissions are not granted, whenever the application attempts to access Windows performance counters, the *Access to the registry key 'Global'* is denied error is registered in the Sitecore log files.

To avoid this error, you must prevent Sitecore from updating the performance counters.

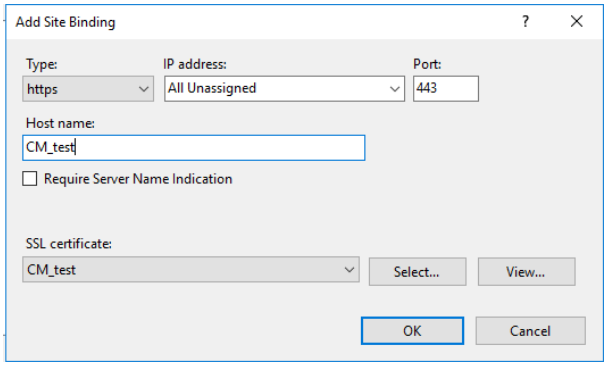
To prevent Sitecore from updating the performance counters:

- In a text editor, open the `\App_Config\Sitecore.config` file and set the `Counters.Enabled` setting to *false*.

8.3. Certificates

8.3.1. Server certificates

After you install Sitecore XP, your IIS site will have an HTTPS binding and an associated SSL certificate with the same name. For example, if the site is named `CM_test` , the HTTPS binding and associated SSL certificate are also named `CM_test`.



Configure a new client certificate

If your client certificate has expired, you must configure Sitecore to use a new client certificate.

To configure Sitecore to use a new client certificate:

1. Install the new client certificate on every computer on which you have installed the xConnect client and ensure that the authority that issued the certificate is in the *Trusted Authorities* list. For more information about the appropriate role and the certificate that you must install, see the section *Set up the certificates*.
2. To grant the appropriate permissions to the certificate, open the Microsoft Management Console, click **File**, and then click **Add/Remove Snap-in**.
3. In the **Add or Remove Snap-ins** dialog box, in the **Available snap-ins** field, select **Certificates** and then click **Add**.
4. In the **Certificates snap-in** dialog box, select **Computer account** and then click **Next**.
5. In the **Select Computer** dialog box, select **Local computer** and then click **Finish**.
6. In the **Add or Remove Snap-ins** dialog box, click **OK**.
7. In the **Console** window, in the left-hand pane, navigate to the *Certificates (Local Computer)/Personal/Certificates* folder.
8. In the center pane, right-click the new certificate, click **All Tasks**, **Manage Private Keys**.
9. In the **Permissions** dialog box, add the accounts that you want to grant permissions to, based on the following criteria:
   - For virtual accounts that were created for each Sitecore application pool identity, add for example:
     IIS AppPool\<AppPoolName> – for virtual accounts.
     NETWORK SERVICE account – only if the Sitecore website application pools run under the NetworkService identity.
     LOCAL SERVICE account – the Marketing Automation Engine runs under this account.
   - For virtual accounts that were created for the xConnect application pool identity for the website hosting the *xDB Automation Operations* role, add for example:
     IIS AppPool\<AppPoolName> – for virtual accounts.
10. In the `\App_Config\connectionstrings.config` file, in the appropriate connection strings, replace the old thumbprint parameter value with the new client certificate thumbprint.
11. Restart IIS on every computer that you configured to use a new client certificate.

8.3.2. Configure Sitecore XP to use new server certificates

To configure Sitecore XP to use new server certificates:

1. On each IIS instance, in the **Site Bindings** window, select the new server certificate.
2. Restart IIS on every computer that you configured to use the new server certificates.