

Sitecore Experience Platform Installation Guide

Sitecore Experience Platform 9.0 Update 2

Installation guide for administrators and developers



sitecore[®]
Own the experience[™]

Table of Contents

Chapter 1	Introduction	4
1.1	Getting Started	5
1.1.1	Preparing to Install Sitecore XP	5
1.1.2	Sitecore Installation Framework	5
Chapter 2	Prerequisites and Requirements	6
2.1	Sitecore Hosting Environment Requirements	7
2.1.1	IIS Requirements	7
2.1.2	Operating System Requirements	7
2.1.3	.NET Framework Requirements	7
2.1.4	Microsoft Visual C++ 2015 Redistributable Requirements	7
2.1.5	Visual Studio Requirements for Custom Solutions	8
2.1.6	Database Requirements	8
2.1.7	Search Indexing Requirements	8
2.1.8	Antivirus Software Considerations	9
2.1.9	Hardware Requirements for a Server Running a Single Sitecore Installation	9
2.2	Sitecore Client Requirements	10
2.2.1	Software Requirements	10
2.2.2	Hardware Requirements	10
2.3	Sitecore Installation Prerequisites	11
2.3.1	File System Permissions	11
2.3.2	Prerequisites for using the Sitecore Installation Framework	12
2.3.3	Enable Contained Database Authentication	13
2.3.4	Install Solr	14
Chapter 3	Prepare the Environment for Deployment	15
3.1	Use and Configure Your Topology	16
3.1.1	Choose Your On-premise Topology	16
3.2	Set Up the Sitecore Installation Framework Module	21
3.2.1	Install the Installation Framework Module Using MyGet	21
3.2.2	Manual Installation	21
3.2.3	Validate the Installation	22
3.2.4	Customize the Sitecore Installation Framework	22
3.2.5	Create the Self-signed Certificates	23
3.3	Running SIF Remotely	24
Chapter 4	Local Environment Setup	25
4.1	Install the Prerequisites	26
4.1.1	Download the SIF Configuration Files	26
4.1.2	Set Up the Sitecore Installation Framework	26
4.2	Install Sitecore XP	27
4.2.1	An Example of How to Install a SIF Configuration File	28
4.2.2	Edit and Run the Installation Script	28
Chapter 5	Production Environment Setup	31
5.1	Secure the Sitecore installation	32
5.1.1	Set Up Server Certificate SSL Authentication on IIS	32
5.1.2	Set Up Client Certificates	33
5.1.3	Set Up a SSL Certificate for Solr	36
5.2	Install a Scaled Sitecore XP	37
5.2.1	Specify the Certificates During Installation	37
Chapter 6	Sitecore XP Post-Installation Steps	39
6.1	Configure MongoDB Provider for xConnect	40
6.1.1	Platform Configuration	40
6.1.2	High Availability	41
6.1.3	Sharded cluster configuration	41
6.1.4	Security	41

Sitecore Experience Platform Installation Guide

6.2	Rebuild the Search Indexes and the Link Databases	43
6.3	Deploy Marketing Definitions	44
6.4	Content Expiration	45
6.5	Configure Tracking	47
6.6	Configure Session State Providers	48
6.7	Warm up the Servers	49
6.8	Security Hardening	50
6.9	Configure Email Experience Manager	51
Chapter 7	Troubleshooting	52
7.1	Common Issues	53
Chapter 8	Appendix	54
8.1	Access Rights	55
8.1.1	Use Windows Authentication with SQL Server	55
8.1.2	Windows Performance Counters	56
8.2	Certificates	57
8.2.1	References to the Certificates	57
8.2.2	Use and Configure a New Client Certificate	57
8.2.3	Configure Sitecore to Use New Server Certificates	59
8.3	Collations	60

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 © 2001-2020 Sitecore. All rights reserved.

Chapter 1

Introduction

This guide describes how to install Sitecore Experience Platform 9.0 rev. 180604 (Update-2).

The document contains the following chapters:

- **Chapter 1 – Introduction**
An introduction to the installation process for Sitecore XP 9.0 Update 2.
- **Chapter 2 – Prerequisites and Requirements**
An outline of the installation requirements for Sitecore XP.
- **Chapter 3 – Prepare the Environment for Deployment**
Preparing your environment for the deployment of Sitecore.
- **Chapter 4 – Local Environment Setup**
For developers installing Sitecore on a local environment.
- **Chapter 5 – Production Environment Setup**
How to install Sitecore in a scaled, multiple server role environment.
- **Chapter 6 – Sitecore XP Post-Installation Steps**
Some procedures that you must perform to finalize the installation.
- **Chapter 7 – Troubleshooting**
Answers to common problems you may encounter while installing Sitecore.
- **Chapter 8 – Appendix**
Additional reference information to help you install Sitecore.

1.1 Getting Started

Sitecore is divided into two distinct product areas. It contains:

- Sitecore Experience Management (XM) – the content management and personalization features.
- Sitecore Experience Platform (XP) – the content management, personalization, marketing and analytics features.

The Sitecore Experience Platform is divided into a number of logical areas:

- Sitecore Experience Database (xDB) – where all experience data of the contact is stored.
- xConnect – an independent service layer that connects the xDB to Experience Applications and allows other channels to add data to the xDB.
- Experience Applications – with applications such as List Manager, Campaign Manager, FXM, and Experience Analytics.
- Experience content management – with applications such as the Experience Editor and Experience Explorer.

You can install the entire Sitecore Experience Platform (XP), or the Experience Management (XM) solution.

For more information about Sitecore XP or XM, refer to the Sitecore Documentation site – <http://doc.sitecore.net>.

For assistance, or to report any discrepancies between this document and the product, please contact <http://support.sitecore.net/helpdesk/>.

1.1.1 Preparing to Install Sitecore XP

The Sitecore Experience Platform is designed to be used in production environments. Sitecore can also be run in a local development environment for the development of Sitecore by a web developer.

For small implementations, including local environments such as developer workstations and testing environments, Sitecore XP and the database server can be installed on a single computer.

For information about installing Sitecore in a local environment, see the chapter *Local Environment Setup*.

For larger implementations, the database server is typically separated from the application server.

The content authoring environment for business users is also frequently separated from the content delivery environment that is accessed by Internet users.

For information about installing Sitecore in a production environment, see the chapter *Production Environment Setup*.

For information about scaling Sitecore and security hardening, see the [Scaling and Architecture Guide](#) and the [Security hardening](#) sections on the Sitecore Documentation site.

For Cloud deployments and installation, refer to the relevant [installation guidelines](#) available on the Sitecore Documentation site – <https://doc.sitecore.net>.

1.1.2 Sitecore Installation Framework

With Sitecore XP 9.0, we introduced the Sitecore Installation Framework (SIF). You must use SIF to install Sitecore. The framework deploys Web Deploy Packages (WDP) by passing parameters to SIF configuration files through a Microsoft® PowerShell module.

If you want to customize your Sitecore configuration, refer to the [Sitecore Installation Framework Configuration Guide](#).

You can download the guide from the Sitecore Downloads page – <https://dev.sitecore.net>.

Chapter 2

Prerequisites and Requirements

This chapter describes the prerequisites, hardware, and software requirements for Sitecore XP 9.0 rev 180604 Update 2.

This chapter contains the following sections:

- Sitecore Hosting Environment Requirements
- Sitecore Client Requirements
- Sitecore Installation Prerequisites

2.1 Sitecore Hosting Environment Requirements

Sitecore XP 9.0 has specific requirements for the operating system, IIS Web Server, .NET Framework, and the database server.

Important

When you configure the Sitecore Experience Database (xDB), you must synchronize all the servers in your solution to a single reliable time source, for example, by means of the Network Time Protocol (NTP). The aggregation of engagement automation states depends on the system time, and changing this can lead to incorrect aggregation results or loss of data.

2.1.1 IIS Requirements

Sitecore XP can be hosted on the following IIS versions:

- IIS 10.0
- IIS 8.5

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 XP does not officially support any other ASP.NET web servers such as IIS Express, or Mono Web Server, and it neither supports nor allows multiple IIS website definitions to point to the same Sitecore web root.

Important

If you plan to use one or more processing, dedicated publishing, and/or indexing servers that do not handle requests, you must use [Application Initialization](#) to successfully start Sitecore after you recycle the application pool. If you do not do this, Sitecore will not launch and its application pool can shut down due to inactivity.

2.1.2 Operating System Requirements

Sitecore XP 9.0 is only compatible with the client and server operating systems that support .NET Framework 4.6.2 or later.

Sitecore XP can be hosted on the following Microsoft operating systems:

- Windows Server 2016
- Windows Server 2012 R2 (64-bit)
- Windows 10 (32/64-bit)
- Windows 8.1 (32/64-bit)

Important

Run Windows Update and install all the appropriate service packs and security updates on all of your Sitecore XP host and client computers.

2.1.3 .NET Framework Requirements

Sitecore XP 9.0 requires .NET Framework 4.6.2 or later.

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

2.1.4 Microsoft Visual C++ 2015 Redistributable Requirements

Sitecore XP 9.0 Update-1 introduced a new prerequisite for the Microsoft Visual C++ 2015 Redistributable. For more information, see <https://www.microsoft.com/en-us/download/details.aspx?id=53587>.

Note

This redistributable may already be installed with Microsoft Windows. Without it, Sitecore XP 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.*

2.1.5 Visual Studio Requirements for Custom Solutions

Sitecore XP 9.0 requires Microsoft Visual Studio 2015 or later.

2.1.6 Database Requirements

Sitecore 9.0 supports the following database servers:

- Microsoft SQL Server 2016 SP1
This version supports the XM databases and is the required and only supported version for the Experience Database (xDB).
- Microsoft SQL Server 2014 SP2
This version only supports XM databases and does not support the Experience Database (xDB).
- MongoDB Server 3.6.6
This is the required and only supported version of MongoDB for the Experience Database (xDB). In this release we use Mongo session state provider 2.6.1.

Note

Sitecore XP 9.0.2 does not support the MMAPv1 storage engine because it does not support retryable writes. For more information about retryable writes, see the section *High Availability*.

Important

Sitecore XP 9.0 Update 2 does not currently support Oracle databases for the Experience Database (xDB). Support will be added in future versions of Sitecore.

Microsoft SQL Server Drivers and Utilities

You must also install:

- [Microsoft ODBC Driver 13 for SQL Server](#)
- [Microsoft Command Line Utilities 13 for SQL Server](#)

2.1.7 Search Indexing Requirements

Sitecore XP 9.0 supports several index providers. For information about installing or managing them in a Sitecore context, see to the [Sitecore Documentation site](#).

Sitecore supports the following index providers:

- Solr 6.6.2
For the XP Single (XP0), XP Scaled (XP1) and XM Scaled (XM1) topologies, Solr is the default search provider.
- Azure Search
The Azure Search provider is supported and recommended for Azure Cloud PaaS deployments only.
- Lucene

Note

Lucene only supports Content Search and does not support xConnect.

Sitecore Experience Platform Installation Guide

The Sitecore Content Search API uses the native Microsoft Windows IFilter interface to extract text from media files so that Sitecore Content Search can index it.

However, to enable the Sitecore Content Search API crawlers to properly index the content in Adobe PDF files, you must install Adobe PDF IFilter on every content management and content delivery server.

Currently, the only supported version of Adobe PDF IFilter is version 9.

You can install Adobe PDF IFilter as a standalone IFilter or as part of Adobe Acrobat Reader. You can download Adobe PDF IFilter version 9 from:

`ftp://ftp.adobe.com/pub/adobe/acrobat/win/9.x/`.

Note

Adobe has published a known issue about running Adobe PDF IFilter version 9 on Microsoft Windows 8. For more information see: <https://helpx.adobe.com/acrobat/kb/pdf-search-breaks-110-install.html>.

2.1.8 Antivirus Software Considerations

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

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 that is 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.

2.1.9 Hardware Requirements for a Server Running a Single Sitecore Installation

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

- 4 core processor
- 16GB 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.

2.2 Sitecore Client Requirements

This section describes the software and hardware requirements for Sitecore when accessed by a client, such as a desktop computer or a mobile device.

2.2.1 Software Requirements

Browser

Sitecore XP clients are browser-based user interfaces. Sitecore XP 9.0 has been tested and can run on the following browsers:

- Microsoft Internet Explorer 11
- Mozilla Firefox
- Google Chrome
- Microsoft Edge
- Apple Safari 9+

Note

Sitecore XP 9.0 Update 2 supports all the current stable versions of these browsers. However, it does not support the Compatibility view in Internet Explorer 11.

Although Sitecore XP 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](#).

2.2.2 Hardware Requirements

Sitecore XP 9.0 has the following hardware requirements for the client device:

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

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

2.3 Sitecore Installation Prerequisites

Before you install Sitecore, you must fulfill all the requirements for the platform and the installation framework.

2.3.1 File System Permissions

File System Permissions for ASP.NET Requests

Sitecore XP 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.

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

IIS Version	Default ASP.NET account name
8.5	NETWORK SERVICE
10	NETWORK SERVICE

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 the following [article](#).

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 XP application requires access to multiple system files and folders that are not distributed as a part of Sitecore XP, but are installed as a part of the Windows Operating System and the .NET framework. Microsoft has more information about [built in groups and accounts in IIS](#).

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 XP, you must assign the <i>Modify</i> access rights to the <code>\temp</code> 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.

Sitecore Experience Platform 9.0 Update 2

These variables have the following default values:

Variable	Default value
%WINDIR%	C:\Windows
%PROGRAMDATA%	C:\ProgramData for IIS 7 and later

Note

The Sitecore Installation Framework specifies the required permissions for certificates under the \Crypto folder.

UNC Share is Not Supported

You must install Sitecore XP 9.0 on a local drive, not a Universal Naming Convention share.

Sitecore Cannot Operate from a Virtual Directory

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

2.3.2 Prerequisites for using the Sitecore Installation Framework

To use the Sitecore Installation Framework to install Sitecore XP in an on-premises environment you must download and install:

- [Microsoft PowerShell® version 5.1 or later](#)
- [Web Platform Installer 5.0.](#)
- Microsoft [sqlcmd Utility](#).

The sqlcmd Utility is part of SQL Server. If you do not have SQL Server installed on the computer that is running SIF, you must download and install the sqlcmd Utility.

Note

The Sitecore Installation Framework does not validate the prerequisite software. You must ensure that you install the correct versions.

The Sitecore Installation Framework requires:

Requirement	Feature	Details
WebAdministration module	Supports IIS management.	When you configure a computer with IIS, the WebAdministration module is automatically installed.
Web Deploy 3.6 for Hosting Servers	Supports the installation of Web Deploy Packages.	You install the tool using the Web Platform Installer.
URL Rewrite 2.1	Supports URL rewrites for Sitecore when installed as a Web Deploy Packages.	You install the tool using the Web Platform Installer.

Requirement	Feature	Details
Microsoft SQL Server Data-Tier Application Framework (DacFx) version 2016	Supports the installation of .dac files in Web Deploy Packages.	<p>You can install DacFx from here: https://www.microsoft.com/en-us/download/details.aspx?id=53013</p> <p>To ensure that DacFx works correctly, you must install its system requirements including SQLSysCLRTypes.msi.</p> <p>If you are running an x64 environment, must install both the x64 and x86 versions of DacFx and SQLSysCLRTypes.</p> <p>Note If DACFx fails to install, you can see the following error message when using the framework: <i>The SQL provider cannot run with dacpac option because of a missing dependency. Please make sure that DACFx is installed.</i> To resolve this error, refer to the following Sitecore Knowledge Base article.</p>

Clear the Web Platform Installer download cache

If the Web Platform Installer hangs or freezes during the installation, you must restart and clear the download cache.

To clear the Web Platform Installer download cache:

1. Launch the Web Platform Installer.
2. In the bottom pane, click **Options**.
3. In the **Change Options** dialog box, scroll down to the **Installer cache** section and click **Delete installer cache folder**.
4. Click **OK**.

2.3.3 Enable Contained Database Authentication

When you use Web Deploy Packages, you must ensure that the target SQL Server is configured correctly.

To configure the target SQL Server to allow users and logins to be contained at the database level:

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

```
sp_configure 'contained database authentication', 1;  
GO  
RECONFIGURE;  
GO
```

Note

Microsoft has more information about the [contained database authentication option](#).

2.3.4 Install Solr

The Sitecore Experience Platform 9.0 rev. 180604 (Update-2) supports Solr, Lucene, and Azure Search as search providers. If you want to use a search index that works in both analytics and content search, we recommend that you use Solr.

Note

If you want to use a search index that works in both analytics and content search on PaaS deployments, you should use Azure search.

Before you run the Sitecore Installation Framework, you must:

- [Enable and set up SSL](#)
- [Install Solr](#) and configure it to run as a Windows service.

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

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](#).

Install the Solr Certificate

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

- Content Management
- xConnect Collection Search

Note

The new Dedicated Dispatch Server (DDS) role that was introduced with Sitecore XP 9.0. Update 1 can only be configured on a CM server and therefore requires the Solr certificate.

For more information about [configuring a Dedicated Dispatch Server](#), see the EXM documentation.

Chapter 3

Prepare the Environment for Deployment

This chapter describes how to prepare your environment for a local or remote deployment of the Sitecore Experience Platform.

This chapter contains the following sections:

- Use and Configure Your Topology
- Set Up the Sitecore Installation Framework
- Running SIF Remotely

3.1 Use and Configure Your Topology

Before you install Sitecore, you must choose the topology or the instance that you want to install.

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

- XP Single (XPo)
- XM Scaled (XM1)
- XP Scaled (XP1)

Important

You can configure the topology to match your business needs. For more information about scaling, see the [Scaling and Architecture Guide](#).

Note

If you want to install the XPo topology, you only need to install the CM instance from the XM1 topology and then in the `web.config` file, specify the following setting:

```
<add key="role:define" value="Standalone" />
```

3.1.1 Choose Your On-premise Topology

The following table describes the three types of deployment topologies available: XP Single (XPo), XM Scaled (XM1), and XP Scaled (XP1).

Note

Azure Cloud supports different deployment topologies. For more information, see the documentation about [Sitecore configurations and topology for Azure](#).

To deploy Sitecore XP in Azure Cloud, you must use Sitecore Azure Toolkit and the appropriate Sitecore Azure WDP.

Deployment topology	Description
XP Single (XPo)	<p>The Sitecore Experience Platform, running as two single instances: Sitecore and xConnect. The Experience Database (xDB) is partially included in both instances.</p> <p>Use this topology for local development and testing. You can also use this topology in a production environment that combines the server roles.</p> <p>Note For security and scalability reasons, in production environments, it is best practice to use the XM Scaled (XM1) or XP Scaled (XP1) configuration.</p>
XM Scaled (XM1)	<p>The Sitecore Experience Management configuration (CMS-only mode) running both the Content Delivery (CD) and Content Management (CM) roles.</p> <p>Use this topology when you are not planning to use the Analytics and Marketing features of the Sitecore Experience Platform.</p> <p>Note When you select this topology, xDB and xConnect are not available.</p>

Deployment topology	Description
XP Scaled (XP1)	<p>The Sitecore Experience Platform configuration running the following separated server roles:</p> <ul style="list-style-type: none">• Content Delivery,• Content Management,• Content Management + DDS (optional),• Processing, Reporting,• xConnect Collection,• xConnect Collection Search,• xDB Reference Data,• xDB Automation Operations,• xDB Automation Reporting. <p>Use this topology when you are planning a fully featured Sitecore Experience Platform installation.</p> <p>Use this topology only if you are installing dedicated server roles.</p>

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*.

Scalability

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 scalability, read the [Scaling and Architecture Guide](#).

Download the Web Deploy Packages and SIF Configuration Files

After you select your deployment topology, you must download the corresponding web deploy packages from the Sitecore Downloads page – <https://dev.sitecore.net>. The web deploy packages also contain the SIF configuration files.

XP Single (XP0)

The following Web Deploy Packages are required for XP Single (XP0) topologies:

- `sitecore 9.0.2 rev. 180604 (OnPrem)_single.scwdp.zip`
- `sitecore 9.0.2 rev. 180604 (OnPrem)_xp0xconnect.scwdp.zip`

The following SIF configuration files are required for XP Single (XP0) topologies:

- `sitecore-solr.json`
- `xconnect-solr.json`
- `xconnect-createcert.json` (for developer environments)

Important

When you install the xConnect configurations on IIS 8.5 - Windows 2012 R2 or Windows 8.1, you cannot use the `xconnect-createcert.json` to generate a self-signed client certificate for xConnect. You must provide a certificate for the installation. For more information, see to section *Secure the Sitecore installation*.

- `sitecore-XP0.json`
- `xconnect-xp0.json`

XM Scaled (XM1)

The following Web Deploy Packages are required for XM Scaled (XM1) topologies:

- `sitecore 9.0.2 rev. 180604 (OnPrem)_cd.scwdp.zip`
- `sitecore 9.0.2 rev. 180604 (OnPrem)_cm.scwdp.zip`

The following SIF configuration files are required for XM Scaled (XM1) topologies:

- `sitecore-solr.json`
- `sitecore-XM1-cd.json`
- `sitecore-XM1-cm.json`

XP Scaled (XP1)

The following Web Deploy Packages are required for XP Scaled (XP1) topologies:

- `sitecore 9.0.2 rev. 180604 (OnPrem)_cd.scwdp.zip`
- `sitecore 9.0.2 rev. 180604 (OnPrem)_cm.scwdp.zip`
- `sitecore 9.0.2 rev. 180604 (OnPrem)_prc.scwdp.zip`
- `sitecore 9.0.2 rev. 180604 (OnPrem)_rep.scwdp.zip`
- `sitecore 9.0.2 rev. 180604 (OnPrem)_xplcollection.scwdp.zip`
- `sitecore 9.0.2 rev. 180604 (OnPrem)_xplcollectionsearch.scwdp.zip`
- `sitecore 9.0.2 rev. 180604 (OnPrem)_xplmarketingautomation.scwdp.zip`
- `sitecore 9.0.2 rev. 180604 (OnPrem)_xplmarketingautomationreporting.scwdp.zip`
- `sitecore 9.0.2 rev. 180604 (OnPrem)_xplreferencedata.scwdp.zip`
- `sitecore 9.0.2 rev. 180604 (OnPrem)_dds.scwdp.zip`

The following SIF configuration files are required for XP Scaled (XP1) topologies:

- `sitecore-solr.json`
- `xconnect-solr.json`
- `xconnect-createcert.json` (for local test environments)

Important

When you install the xConnect configurations on IIS 8.5 - Windows 2012 R2 or Windows 8.1, you cannot use the `xconnect-createcert.json` to generate a self-signed client certificate for xConnect. You must provide a certificate for the installation. For more information, see to section *Secure the Sitecore installation*.

- `sitecore-XP1-cd.json`
- `sitecore-XP1-cm.json`
- `sitecore-XP1-prc.json`
- `sitecore-XP1-rep.json`
- `sitecore-XP1-dds.json`
- `sitecore-XP1-cm-dds-patch.json`
- `sitecore-XP1-cm-dds-patch.ps1`
- `xconnect-xpl-collection.json`

- `xconnect-xp1-collectionsearch.json`
- `xconnect-xp1-marketingautomation.json`
- `xconnect-xp1-marketingautomationreporting.json`
- `xconnect-xp1-referencedata.json`

Note

This document does not describe how to configure a DDS server. For more information, see the [Sitecore Email Campaign Manager documentation](#).

Configure Parameters in the SIF Configuration Files

The SIF configuration files are templates that provide 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 chosen 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. For each parameter, if it has a default value, check whether you need to change it. 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 will prompt you for that information.

Important

The `SolrZookeeperUrl` parameter has been deprecated. Do not change its default value.

Note

In the configuration files, each parameter has a `description` property that explains what it is used for. For example, the following screenshot shows a snippet of a `Parameters` block from a configuration file.

```
{
  "Parameters": {
    // Parameters are values that may be passed when Invoke-SitecoreInstall is called.
    // Parameters must declare a Type and may declare a DefaultValue and Description.
    // Parameters with no DefaultValue are required when Invoke-SitecoreInstall is called.

    "Package": {
      "Type": "string",
      "Description": "The path to the Web Deploy package to deploy."
    },
    "LicenseFile": {
      "Type": "string",
      "Description": "The path to the Sitecore license file."
    },
    "SqlDbPrefix": {
      "Type": "string",
      "Description": "The prefix used for all Sql databases."
    },
    "SiteName": {
      "Type": "string",
      "DefaultValue": "SitecoreCD",
      "Description": "The name of the site to be deployed."
    },
    "SqlCoreUser": {
      "Type": "string",
      "DefaultValue": "sa",
      "Description": "The Sql user for the Core connection string in Sitecore."
    }
  }
}
```

3.2 Set Up the Sitecore Installation Framework Module

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

Because the Sitecore Experience Platform is designed to be secure-by-default, for developer environments there is a Sitecore Fundamentals module that sets up all the required self-signed certificates for you.

Sitecore Fundamentals is a PowerShell module that is used to configure security certificates and transport level security for Sitecore websites. Sitecore Fundamentals is automatically installed when you install SIF.

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.

3.2.1 Install the Installation Framework Module Using MyGet

The Sitecore Gallery is a public MyGet feed that is used to download and install PowerShell modules created by Sitecore. SIF is available through the Sitecore Gallery.

To set up SIF:

1. In Windows, launch PowerShell as an administrator.
2. To register the repository, in a PowerShell command line, run the following cmdlet:

```
Register-PSRepository -Name SitecoreGallery -SourceLocation https://sitecore.myget.org/F/sc-powershell/api/v2
```

3. Install the PowerShell module by running the following cmdlet:

```
Install-Module -Name SitecoreInstallFramework -RequiredVersion 1.2.1
```

4. When prompted to install, press Y, and then press ENTER.

```
PS C:\> Install-Module -Name SitecoreInstallFramework -RequiredVersion 1.2.1
Untrusted repository
You are installing the modules from an untrusted repository. If you trust this repository, change its
InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from
'SitecoreGallery'?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): y
PS C:\>
```

Important

You must install Sitecore 9.0.2 with the Sitecore Installation Framework 1.x. You cannot install Sitecore 9.0.2 with Sitecore Installation Framework 2.x.

3.2.2 Manual Installation

SIF is also available as a .zip package. To manually install SIF, you must also download and install the Sitecore Fundamentals package.

You can download the Sitecore Fundamentals and the SIF packages from the Sitecore Downloads page – <https://dev.sitecore.net>.

Note

When you download the packages, it is possible that the .zip packages are marked as blocked by Microsoft Windows. To install SIF, you must first unblock the .zip packages.

To unblock a .zip package

1. In Windows Explorer, navigate to the folder where you downloaded the .zip packages, and right-click the relevant .zip file
2. Click **Properties**.
3. In the **Properties** dialog box, on the **General** tab, click **Unblock**.
4. Click **OK**.

Extract the Sitecore Installation Framework

The installation path depends on the location where you want to install the SIF. You can install it for all users (global path), for a specific user, or to a custom location:

Usage	Path
All users (<i>global path</i>)	C:\Program Files\WindowsPowerShell\Modules
Specific user	C:\Users\ <i><user></i> \Documents\WindowsPowerShell\Modules
Custom location	Any path

Important

If you want to install SIF to a custom location, after the installation you must import the module and specify the path to the file by running the following cmdlet:

```
Import-Module <custompath>\SitecoreInstallFramework
```

However, if you added SIF to an *All users* or *Specific user* path, you do not have to import the module, because this is done automatically.

For example, if you want to make SIF available to all users:

- Extract the Sitecore Fundamentals .zip package to the following path:

```
C:\Program Files\WindowsPowerShell\Modules\SitecoreFundamentals
```

- Extract the Sitecore Install Framework .zip package to the following path:

```
C:\Program Files\WindowsPowerShell\Modules\SitecoreInstallFramework
```

3.2.3 Validate the Installation

After you install SIF, you can validate its installation to confirm that it is available for use.

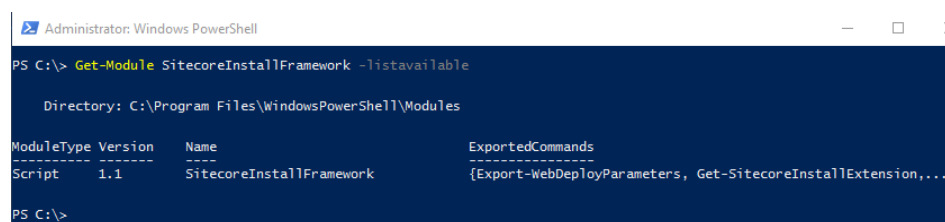
Note

This validation only works if you have installed SIF to the *All users* (global) path.

To validate the installation:

- In a PowerShell command line, run the following cmdlet:

```
Get-Module SitecoreInstallFramework -ListAvailable
```



```
Administrator: Windows PowerShell
PS C:\> Get-Module SitecoreInstallFramework -listavailable

Directory: C:\Program Files\WindowsPowerShell\Modules

ModuleType Version Name ExportedCommands
-----
Script 1.1 SitecoreInstallFramework {Export-WebDeployParameters, Get-SitecoreInstallExtension,...}

PS C:\>
```

3.2.4 Customize 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, read the *Customize the Sitecore Installation Framework* section in the Sitecore Installation Framework Configuration Guide. You can download the guide from the Sitecore Downloads page – <https://dev.sitecore.net>.

3.2.5 Create the Self-signed Certificates

SIF depends on the Sitecore Fundamentals module for local environment installations because it helps create self-signed certificates for developers so they can quickly get started on their workstations.

Note

To secure communication, for example with XConnect, Sitecore uses certificates to encrypt data. In production environments, these certificates must be provided and signed by an authorized provider. However, in development environments, the certificates can be generated and signed locally.

However, if you have installed SIF to a custom location, you must first import the Sitecore Fundamentals package, and then import SIF using its full path.

- To import Sitecore Fundamentals, in a PowerShell command line, run the following cmdlet:

```
Import-Module <custom location>\SitecoreFundamentals
```

- To import SIF, in a PowerShell command line, run the following cmdlet:

```
Import-Module <custom location>\SitecoreInstallFramework
```

3.3 Running SIF Remotely

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

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. The MSDN website has more details on [securing or configuring a computer for remote access](#).

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, as follows:

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

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

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

4. Run the following cmdlet to start the installation:

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

Note

For more details about the `Invoke-Command` cmdlet, see the [PowerShell documentation](#) on the MSDN website.

Chapter 4

Local Environment Setup

This chapter describes how to install Sitecore XP 9.0 rev. 180604 (Update-2) in a local environment for development or evaluation purposes.

This gives you a working instance of Sitecore XP with the XP Single topology.

This chapter contains the following sections:

- Install the Prerequisites
- Install Sitecore XP

4.1 Install the Prerequisites

Before you install your local Sitecore XP environment, ensure that you have installed all the prerequisites and prepared your environment as described in Chapters 2 and 3.

4.1.1 Download the SIF Configuration Files

You must download the relevant SIF configuration files and the Web Deploy Packages (WDPs) that contain the XP Single (XPo) topology from the Sitecore Downloads page – <http://dev.sitecore.net>:

Download the following SIF configuration files:

- `sitecore-solr.json`
- `xconnect-solr.json`
- `xconnect-createcert.json` (for developer environments)
- `sitecore-XP0.json`
- `xconnect-xp0.json`

Download the following WDPs:

- Sitecore 9.0.2 rev. 180604 (OnPrem)_single.scwdp.zip
- Sitecore 9.0.2 rev. 180604 (OnPrem)_xp0xconnect.scwdp.zip

4.1.2 Set Up the Sitecore Installation Framework

You must set up SIF before you can install Sitecore XP.

For more information about setting up SIF, see the section *Set Up the Sitecore Installation Framework Module*.

4.2 Install Sitecore XP

The Sitecore installation is a combination of SIF configuration files, WDP packages, and databases.

SIF uses the SIF configuration files to configure the environment and install the application and databases using the WDP packages.

The predefined XP Single topology configures:

- The Sitecore stand-alone website that handles content management, content delivery, reporting, and processing.
- The xConnect and xDB web services.
- Search indexes on the Solr search engine.
- A Windows service that runs the Marketing Automation engine.
- A Windows service that runs the xConnect indexer.
- The Sitecore content and xDB databases.
- A self-signed client certificate for secure communication between Sitecore and xConnect.
- A self-signed server certificate for running HTTPS on the xConnect and xDB web services.

This topology is implemented by installing the SIF configuration files in the following order:

- `xconnect-createcert.json`

This SIF configuration file creates the self-signed client certificate which is passed to the `xconnect-xp0.json` and `sitecore-XP0.json` SIF configuration files.

Important

When you install the xConnect configurations on IIS 8.5 - Windows 2012 R2 or Windows 8.1, you cannot use the `xconnect-createcert.json` configuration file to generate a self-signed client certificate for xConnect. You must provide a certificate for the installation. For more information, see to section *Secure the Sitecore installation*.

- `xconnect-solr.json`

This SIF configuration file creates the Solr indexes that are used by xConnect.

- `xconnect-xp0.json`

This SIF configuration file sets-up the xDB and xConnect web services on IIS, the xDB databases on SQL Server, and secures them with the provided certificates.

This configuration uses the following WDP package:

- `Sitecore 9.0.2 rev. 180604 (OnPrem)_xp0xconnect.scwdp.zip`

- `sitecore-solr.json`

This SIF configuration file creates the Solr indexes that are used by Sitecore.

- `sitecore-XP0.json`

This SIF configuration file sets-up the stand-alone Sitecore website on IIS and the content databases on SQL Server.

This configuration uses the following WDP package:

- `Sitecore 9.0.2 rev. 180604 (OnPrem)_single.scwdp.zip`

Each of these SIF configuration files requires a separate set of parameters that must be passed to them during the installation process. For more information, open the files in a text editor and examine the `Parameters` section.

Note

If you want to provide a signed certificate, you must install it on the local server and the name can then be passed as a parameter to the `xconnect-xp0.json` and `sitecore-XP0.json` SIF configuration files. You can then ignore the `xconnect-createcert.json` SIF configuration file.

For more information, see the section *Secure the Sitecore installation*.

4.2.1 An Example of How to 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. To start the installation, run the `Install-SitecoreConfiguration` cmdlet, and specify the path to your SIF configuration file.

For example, using the `sitecore-XP0.json` file:

```
Install-SitecoreConfiguration -Path <configurationpath>\sitecore-XP0.json
```

Optionally, parameters declared in 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-XP0.json -SqlDbPrefix SC.
```

In a PowerShell command line, additional parameters can be passed to control the installation process. For example, running the `Verbose` cmdlet increases the information logged, and the `-Skip <taskname>` cmdlet skips one or more tasks.

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

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

Note

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

4.2.2 Edit and Run the Installation Script

To simplify your installation, you can use a PowerShell script to install the XP Single (XPo) topology.

To edit and run the installation script:

1. Create a folder called `c:\resourcefiles`.
2. Download and save the following WDP packages and SIF resource files in this folder:
 - o `sitecore 9.0.2 rev. 180604 (OnPrem)_single.scwdp.zip`
 - o `sitecore 9.0.2 rev. 180604 (OnPrem)_xp0xconnect.scwdp.zip`
 - o `sitecore-solr.json`
 - o `xconnect-solr.json`
 - o `xconnect-createcert.json` (for developer environments)
 - o `sitecore-XP0.json`
 - o `xconnect-xp0.json`
3. Save your Sitecore license file in the `c:\resourcefiles` folder as `license.xml`.
4. Edit the highlighted sections of the following script:

```
#define parameters  
$prefix = "xp0"  
$PSScriptRoot = "C:\resourcefiles"
```

```
$XConnectCollectionService = "$prefix.xconnect"
$sitecoreSiteName = "$prefix.sc"
$SolrUrl = "https://localhost:8989/solr"
$SolrRoot = "C:\Solr-6.6.2"
$SolrService = "Solr-6.6.2"
$SqlServer = ".\SQL2016"
$SqlAdminUser = "sa"
$SqlAdminPassword="Password12345"

#install client certificate for xconnect
$certParams = @{
    Path = "$PSScriptRoot\xconnect-createcert.json"
    CertificateName = "$prefix.xconnect_client"
}
Install-SitecoreConfiguration @certParams -Verbose

#install solr cores for xdb
$solrParams = @{
    Path = "$PSScriptRoot\xconnect-solr.json"
    SolrUrl = $SolrUrl
    SolrRoot = $SolrRoot
    SolrService = $SolrService
    CorePrefix = $prefix
}
Install-SitecoreConfiguration @solrParams

#deploy xconnect instance
$xconnectParams = @{
    Path = "$PSScriptRoot\xconnect-xp0.json"
    Package = "$PSScriptRoot\Sitecore 9.0.2 rev. 180604
(OnPrem)_xp0xconnect.scwdp.zip"
    LicenseFile = "$PSScriptRoot\license.xml"
    SiteName = $XConnectCollectionService
    XConnectCert = $certParams.CertificateName
    SqlDbPrefix = $prefix
    SqlServer = $SqlServer
    SqlAdminUser = $SqlAdminUser
    SqlAdminPassword = $SqlAdminPassword
    SolrCorePrefix = $prefix
    SolrURL = $SolrUrl
}
Install-SitecoreConfiguration @xconnectParams

#install solr cores for sitecore
$solrParams = @{
    Path = "$PSScriptRoot\sitecore-solr.json"
    SolrUrl = $SolrUrl
    SolrRoot = $SolrRoot
    SolrService = $SolrService
    CorePrefix = $prefix
}
Install-SitecoreConfiguration @solrParams

#install sitecore instance
$xconnectHostName = "$prefix.xconnect"
$sitecoreParams = @{
    Path = "$PSScriptRoot\sitecore-XP0.json"
    Package = "$PSScriptRoot\Sitecore 9.0.2 rev. 180604
(OnPrem)_single.scwdp.zip"
    LicenseFile = "$PSScriptRoot\license.xml"
    SqlDbPrefix = $prefix
    SqlServer = $SqlServer
    SqlAdminUser = $SqlAdminUser
    SqlAdminPassword = $SqlAdminPassword
    SolrCorePrefix = $prefix
    SolrUrl = $SolrUrl
    XConnectCert = $certParams.CertificateName
    SiteName = $sitecoreSiteName
    XConnectCollectionService = "https://$XConnectCollectionService"
}
Install-SitecoreConfiguration @sitecoreParams
```

5. Save the script as `install.ps1` to the `c:\resourcefiles` folder.

Sitecore Experience Platform 9.0 Update 2

6. In a PowerShell command line, navigate to the `c:\resourcefiles` folder and run the following command:

```
.\install.ps1
```

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

Chapter 5

Production Environment Setup

This chapter describes how to install Sitecore XP 9.0 rev. 180604 (Update-2) in a scaled environment for production or test purposes.

This gives you a working instance of the Sitecore XP with the XP Scaled topology.

This chapter contains the following sections:

- Secure the Sitecore installation
- Install a Scaled Sitecore XP

5.1 Secure the Sitecore installation

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

For local testing and development, you can use a self-signed certificate. For more information, see the chapter *Local Environment Setup*.

Server Certificate Authentication

All communication between Sitecore instances, including xConnect web services, and the Solr search provider occurs over the default HTTPS configuration. 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 unsecure network. It does not identify who the client is and the server authentication alone does not determine who can connect to the server.

Client Certificate Authentication

The xConnect server roles support an additional layer of security, referred to as SSL Client Certificate Authentication. SSL Client Certificate Authentication validates that the individual HTTP client is authorized to connect to the HTTP server. SSL Client Certificate Authentication requires that the HTTP client device is configured with a specific client certificate and private key, or thumbprint, which is used to connect to the protected SSL server.

Because xConnect web services use server-to-server communication and are non-interactive, the client certificate allows the Content Management server role and other server roles to connect securely to WebAPI JSON services.

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. Microsoft has more information on [how to set up SSL in IIS](#).

Server Certificates

The following table shows the full set of server authentication certificates for each topology:

XM Scaled (XM1)	XP Single (XP0)	XP Scaled (XP1)
Content Management	xConnect	Content Management Reporting Processing xConnect Collection xConnect Collection Search xDB Reference Data xDB Automation Operations xDB Automation Reporting

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.

Install Server Certificates

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

Sitecore Experience Platform Installation Guide

To install 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 certificate, see [MSDN](#).

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 (XM₁), and XP Single (XP₀) topologies, it is assumed that there is only one SSL certificate for each IIS instance that covers multiple application roles. For XP Scaled (XP₁), there is a dedicated role per server in a distributed setup, and you must obtain and install a certificate for each server role.

For XP Scaled (XP₁), after you obtain all the server certificates, you must install them on the required servers:

XP Scaled (XP₁)

Role Name	Server Certificate
Content Management	Processing Reporting xConnect Collection xConnect Collection Search xDB Reference Data xDB Automation Operations xDB Automation Reporting
Content Delivery	Content Management xConnect Collection xDB Reference Data xDB Automation Operations
Reporting	<i>None required</i>
Processing	xConnect Collection
xConnect Collection	<i>None required</i>
xConnect Collection Search	<i>None required</i>
xDB Reference Data	<i>None required</i>
xDB Automation Operations	xConnect Collection xDB Reference Data
xDB Automation Reporting	<i>None required</i>
EXM Dedicated Dispatch Server	Processing Reporting xConnect Collection xConnect Collection Search xDB Reference Data xDB Automation Operations xDB Automation Reporting

5.1.2 Set Up Client Certificates

You must obtain and install the client certificates before running SIF.

Sitecore Experience Platform 9.0 Update 2

The client certificate is typically installed on the Windows Server that connects to the server where xConnect is deployed. The client certificate is stored in the certificate store for either a specific user or the entire server.

The thumbprint of the client certificate is specified on the server that you are connecting to (the destination), in this case the xConnect server, and only clients with the correct certificate and matching thumbprint are allowed to connect.

In production environments, different client certificates are used for different application roles with the aim of isolating the servers, in the event of a key being compromised.

For development purposes, you can use a single client certificate to validate that authentication will work as expected once you move to a production environment.

Client Certificates

The following table shows the full set of client authentication certificates for each topology:

XM Scaled (XM1)	XP Single (XP0)	XP Scaled (XP1)
None	xConnect	xConnect Collection xConnect Collection Search xDB Reference Data xDB Automation Operations

Install the Client Certificate

After you have obtained the certificates, you must install them.

To install the client certificate:

- Install the client authentication certificate, including the private key, in the `Certificates (Local Computer) \ Personal` folder for each required role.

For information about how to install a private key certificate, see [MSDN](#).

Important

When you import the client certificate, you must select the **Allow private key to be exported** option.

- If you created a self-signed certificate, you must install the self-signed authority certificate used to create the client authentication certificate in the following folder:

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

The thumbprint for the certificates that were installed in the previous step must be added to the *xConnect Collection*, *xConnect Collection Search*, *xDB Reference Data*, *xDB Automation Operations*, and *xDB Automation Reporting* roles.

In the `/App_Config/AppSettings.config` file, add the thumbprint to the `<add key="validateCertificateThumbprint" value="YOUR_CERTIFICATE_THUMBPRINT" />` setting.

This defines which client certificate is used for authentication.

The following tables provide details about the client certificates required for each role:

XP Single (XP0)

Role Name	Client Certificates	Associated connection strings containing client thumbprint
Sitecore	xConnect Client	xconnect.collection.certificate xdb.referencedata.client.certificate xdb.marketingautomation.reporting.client.certificate xdb.marketingautomation.operations.client.certificate

Sitecore Experience Platform Installation Guide

Role Name	Client Certificates	Associated connection strings containing client thumbprint
xConnect	xConnect Client	xdb.referencedata.client.certificate xconnect.collection.certificate

XP Scaled (XP1)

Role Name	Client Certificate	Associated connection strings containing client thumbprint
Content Management	xConnect Collection Search	xconnect.collection.certificate
	xDB Reference Data	xdb.referencedata.client.certificate
	xDB Automation Operations	xdb.marketingautomation.reporting.client.certificate
	xDB Automation Reporting	xdb.marketingautomation.operations.client.certificate
Content Delivery	xConnect Collection	xconnect.collection.certificate
	xDB Reference Data	xdb.referencedata.client.certificate
	xDB Automation Operations	xdb.marketingautomation.operations.client.certificate
Reporting	xConnect Collection	xconnect.collection.certificate
Processing	xConnect Collection	xconnect.collection.certificate
xConnect Collection	<i>None required*</i>	-
xConnect Collection Search	<i>None required*</i>	-
xDB Reference Data	<i>None required*</i>	-
xDB Automation Operations	xConnect Collection	xconnect.collection.certificate
xDB Automation Reporting	<i>None required*</i>	-

* These four roles do not require a client certificate because they do not make calls to other roles.

Note

You must also ensure that client certificate private keys permissions and read access are granted to the users under which your services are running.

By default, these users are:

The *ApplicationPoolIdentity* for the web sites.

The *Local Service* account for Windows services.

SIF does this automatically.

5.1.3 Set Up a SSL Certificate for Solr

As described in the *Install Solr* section, if you want to use the Experience Database (xDB) and xConnect, you must enable SSL for Solr.

Note

In production environments, the Solr certificate must be provided and signed by an authorized provider. However, in development environments, the certificates can be generated and signed locally.

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

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

You must install the Solr SSL certificate for the following server roles:

XM Scaled (XM1)	XP Single (XP0)	XP Scaled (XP1)
<i>None required</i>	xConnect Sitecore	Content Management xConnect Collection Search

5.2 Install a Scaled Sitecore XP

Once you have obtained the required certificates, you can run SIF and install Sitecore. 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.

For more information about scaling, read the [Sitecore Scaling Guide](#).

Important

You must first install the `sitecore-solr.json` and the `xconnect-solr.json` deployment configurations. Then you must install the rest of the Sitecore server roles deployment configurations and these can then be installed in any order.

To run SIF and install Sitecore:

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

```
Import-Module SitecoreInstallFramework
```

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

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

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

```
Install-SitecoreConfiguration -Path  
"C:\SitecoreInstaller\Configurations\Platform\XP1\sitecore-XP1-cm.json"  
  
Install-SitecoreConfiguration -Path  
"C:\SitecoreInstaller\Configurations\Platform\XP1\sitecore-XP1-cd.json"  
  
Install-SitecoreConfiguration -Path  
"C:\SitecoreInstaller\Configurations\Platform\XP1\sitecore-XP1-prc.json"  
  
Install-SitecoreConfiguration -Path  
"C:\SitecoreInstaller\Configurations\Platform\XP1\sitecore-XP1-rep.json"  
  
Install-SitecoreConfiguration -Path  
"C:\SitecoreInstaller\Configurations\xConnect\XP1\xconnect-xp1-collection.json"  
  
Install-SitecoreConfiguration -Path  
"C:\SitecoreInstaller\Configurations\xConnect\XP1\xconnect-xp1-collectionsearch.json"  
  
Install-SitecoreConfiguration -Path  
"C:\SitecoreInstaller\Configurations\xConnect\XP1\xconnect-xp1-  
MarketingAutomationReporting.json"  
  
Install-SitecoreConfiguration -Path  
"C:\SitecoreInstaller\Configurations\xConnect\XP1\xconnect-xp1-  
MarketingAutomation.json"  
  
Install-SitecoreConfiguration -Path  
"C:\SitecoreInstaller\Configurations\xConnect\XP1\xconnect-xp1-ReferenceData.json"
```

For information about setting up EXM, see the [EXM documentation](#).

5.2.1 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 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 as parameters the names or thumbprints of the certificates that you created and installed earlier in this guide. For example:

- For the *client* authentication certificate:

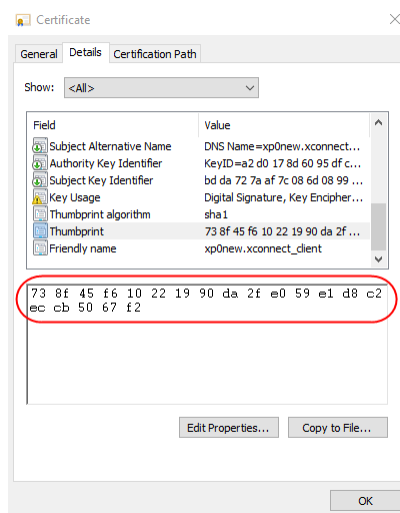
```
-XConnectCert "xConnect_client"
```

or

```
-XConnectCert "738F45F610221990DA2FE059E1D8C2ECCB5067F2"
```

Note

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



- For the *server* authentication certificate, for example, for an instance with the name "CM_test":

```
-SSLCert "CM_test"
```

or

```
-SSLCert "2205a94867ee99e3b29ea7a9ac5a7646d43fd88b"
```

Chapter 6

Sitecore XP Post-Installation Steps

This chapter describes the steps required to complete the installation after you use SIF to install Sitecore XP 9.0 rev. 180604 (Update 2).

This chapter contains the following sections:

- Configure MongoDB Provider for xConnect
- Rebuild the Search Indexes and the Link Databases
- Deploy Marketing Definitions
- Content Expiration
- Configure Tracking
- Configure Session State Providers
- Warm up the Servers
- Security Hardening
- Configure Email Experience Manager

6.1 Configure MongoDB Provider for xConnect

Important

If you are not going to use MongoDB, you can skip this section.

Important

You must use MongoDB Server 3.6.6 or later as it contains important fixes that are essential for the xConnect MongoDB data provider.

6.1.1 Platform Configuration

The xConnect platform is installed with the SQL Provider for the *collection* database by default.

If you want to use Mongo DB as the provider for the *collection* database you must configure a Mongo DB server.

Note

In Sitecore XP 8.1 – 8.2, the *collection* database was called the *analytics* database.

To enable the MongoDB Provider for different roles depending on the topology:

1. Enable the `sc.Xdb.Collection.Data.MongoDb.xml.disabled` configuration file for all the roles that you use in your topology.
2. Disable `sc.Xdb.Collection.Data.Sql.xml` configuration file by adding the `.disabled` file extension for all the roles that you use in your topology.
3. Update the connection string for the *collection* database to point to the MongoDB instance for all the roles that you use in your topology.

The connection string for a configured replica set and with least privilege users should look like this:

```
mongodb://sa:12345@10.45.111.102:57017,10.45.111.102:57018,10.45.111.102:57019/collection?replicaSet=testReplicaSet&retryWrites=true
```

The connection string for a configured sharded cluster and with least privilege users should look like this:

```
mongodb://sa:12345@127.0.0.1:27017/collection?retryWrites=true
```

Note

The collection database is automatically created on the Mongo DB provider the first time you run the Mongo server.

Note

After you switch the data provider from SQL to Mongo, you can delete the SQL *collection* database.

4. Rebuild the [xDB search indexes in Solr](#).

Note

Update the platform configuration for the indexer job (xConnect role).

XP Single (XP0)	XP Scaled (XP1)
xConnect	xConnect Collection xConnect Collection Search

6.1.2 High Availability

You must configure the MongoDB [replica sets](#) and [retryable writes](#) features to ensure high availability.

Replication is a group of MongoDB instances that are configured into a single replica set that maintains the same data set for automatic failover and node recovery.

Note

We recommend that you configure replication and retryable writes in your production environment.

The [minimum replication configuration](#) is a replica set with two members that hold the data and an [arbiter](#).

We also recommend that you change the [write concern](#) option of the default replica set to a number greater than 1. The write concern option specifies the number of replica set nodes for request acknowledgement.

Whereas replication ensures data availability, the current operation must be successfully completed during failover. You configure this in the [connection string](#) with the [retryWrites](#) option.

Note

Retryable writes require a replica set or sharded cluster and do not support standalone instances.

Retryable writes allow the MongoDB driver to retry a write operation if there is a network problem or if the primary node is not healthy. Retryable reads are not supported by the MongoDB driver and were implemented as part of the xConnect MongoDB provider.

The combination of these features provides high availability.

6.1.3 Sharded cluster configuration

[Sharding](#) is a method for scaling databases that distributes data across multiple machines.

Note

MongoDB uses sharding to support deployments with very large data sets and high level of throughput operations.

MongoDB uses shard keys to partition data by collection. The shard key consists of an immutable field or fields that exist in every document in the target collection.

The following table lists the collections and their required shard keys:

Collection	Shard key
Contacts	{_id: 1}
ContactFacets	{_id: 1}
Interactions	{_id: 1}
InteractionFacets	{_id: 1}
DeviceProfiles	{_id: 1}
DeviceProfileFacets	{_id: 1}
ContactIdentifierIndex	{_id:'hashed'}
Changes	{_id:'hashed'}

6.1.4 Security

Important

Follow the MongoDB [security checklist](#) to protect your MongoDB installation.

Sitecore Experience Platform 9.0 Update 2

We recommend that you create least privilege users who can access MongoDB.

The following table contains a list of the privilege actions for least privilege users:

Privilege Sections	Privilege Actions
Query and Write Actions	find; insert; remove; update
Database Management Actions	createIndex
Deployment Management Actions	-
Replication Actions	-
Sharding Actions	-
Server Administration Actions	-
Session Actions	-
Diagnostic Actions	listIndexes; listCollections

6.2 Rebuild the Search Indexes and the Link Databases

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

To rebuild all the indexes:

1. On 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 Deploy Marketing Definitions

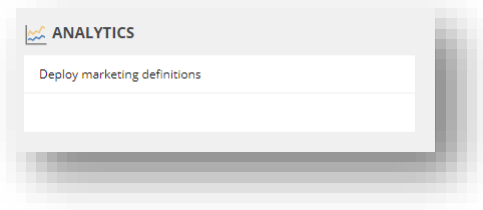
If you want to use the Sitecore Experience Marketing functionality, you must deploy marketing definitions.

Note

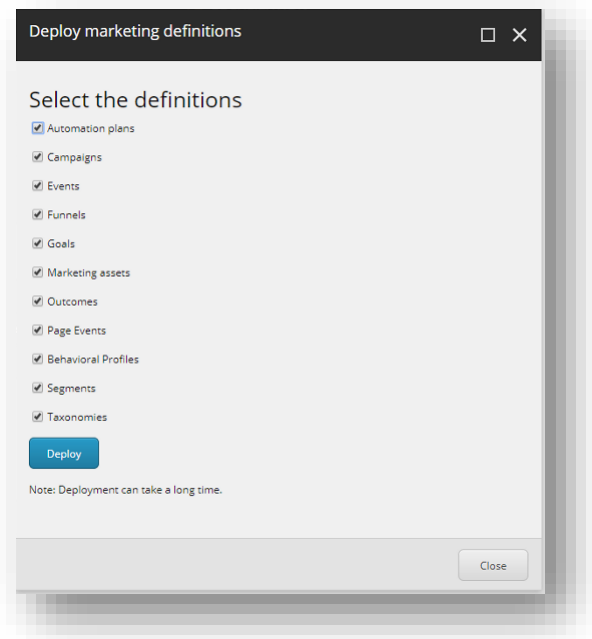
If you do not plan to use the Sitecore Experience Database (xDB), you do not have to perform these steps.

To deploy marketing definitions:

1. On the **Sitecore Launchpad**, click **Control Panel, Analytics**, and then click **Deploy Marketing Definitions**.



2. In the **Deploy marketing definitions** dialog box, select all the definitions and taxonomies and click **Deploy**.



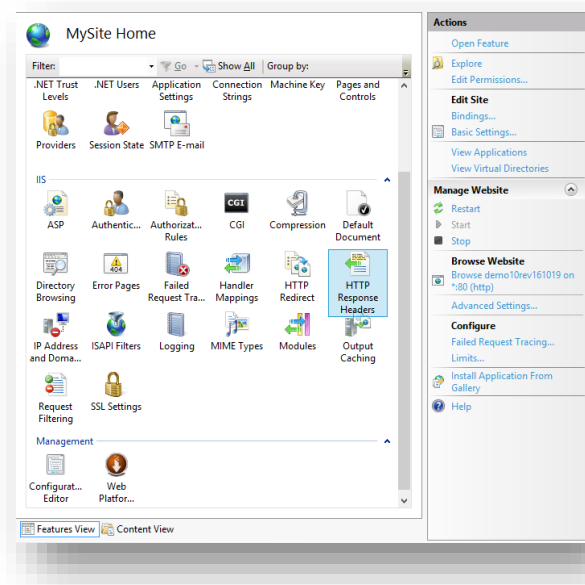
6.4 Content Expiration

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.

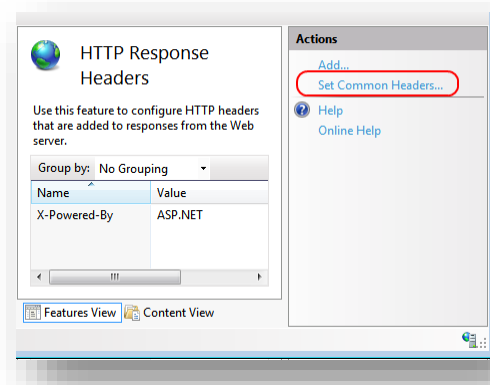
By setting Expire Web content to something other than *immediately*, you can reduce second-access load times by 50 –70%. This setting does not affect dynamically-generated content.

To enable the *Expire Web content* header in IIS:

1. In Windows, open the IIS Manager.
2. Select the site that you want to enable the *Expire Web content* header for.
3. In the **IIS** section, double click **HTTP Response Headers**.



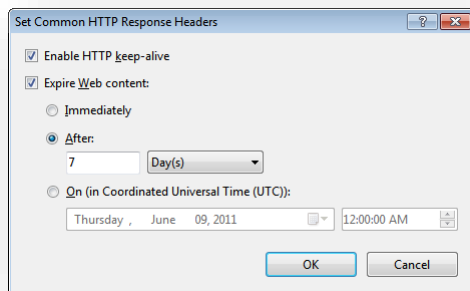
4. In the **Actions** panel, click **Set Common Headers...**



5. Select the **Expire Web content** check box.

Sitecore Experience Platform 9.0 Update 2

6. Select **After**, and set the number of days to a longer expiration time. For example, 7 days.



7. Click **OK**.

6.5 Configure Tracking

Tracking enables you to identify contacts and track their activity on your website.

In certain cases, you might not want to track Geo IP data, or due to restrictions in certain legal jurisdictions, you might not be able or want to store IP addresses. In these situations, you can configure tracking.

Note

The following procedure is optional.

To configure tracking:

1. If you do not want to track Geo IP data:
 - o In the `Website\App_Config\Sitecore\Marketing.Tracking\Sitecore.Analytics.Tracking.config` file, set the `Analytics.PerformLookup` setting to *False*.

Note

The default value of the `Analytics.PerformLookup` setting is *True* and you must not change it in a single-instance environment.

2. If you do not want to store IP addresses in the xDB:
 - o In the `\App_Config\Sitecore\Marketing.Tracking\Sitecore.Analytics.Tracking.config` file, update the `RedactIpAddress` setting.

This change will hash the IP addresses before they are stored in the xDB.

Important

To ensure that hashing is secure, in the `Sitecore.Analytics.Tracking.config` file, in the `geoIpManager` section, change the default salt value.

3. Restart IIS.

6.6 Configure Session State Providers

In the Sitecore Experience Database, you can use a session state server to share all your contact sessions across different browsers and devices. Configuring session state is particularly important if you have deployed a multi-server, fully scalable environment with clusters of content delivery or processing 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.

To configure any *OutOfProc* session state providers, see the [Sitecore documentation](#).

The *Sitecore ASP.NET Session State Provider for SQL Server* enables you to use SQL Server as your session state store. This provider supports the `SessionEnd` event that the xDB needs to track website visits.

For more information about deploying the Session database, see the [Sitecore Documentation](#).

If you use the MongoDB Session State Provider, you must set the `uuidRepresentation` parameter in the connection string, for example:

```
mongodb://localhost/session?uuidRepresentation=standard
```


6.7 Warm up the Servers

To ensure that your Sitecore websites are available at all times, even after restarting a server, you should 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](#).

6.8 Security Hardening

Sitecore recommends 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 may require additional security-related coding and configuration.

For more information about security hardening, see the [security hardening](#) documentation.

6.9 Configure Email Experience Manager

To use EXM you must configure the delivery process.

For more information about EXM and about configuring the delivery process, see the [EXM documentation](#).

Chapter 7

Troubleshooting

This chapter can help you resolve the most common problems that you may encounter when you install Sitecore XP 9.0.

- Common Issues

7.1 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, `sc90_xconnect`.
- Ensure you pasted your thumbprint into a PowerShell command line window, and that you removed the hidden character at the start of the string.
- Make sure 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

- In the `\xConnect\App_data\jobs\continuous\IndexWorker\App_data\Logs` folder, check the indexer's log files:

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 also see the following error: *The SQL provider cannot run with dacpac option because of a missing dependency*. To resolve this, see the following [Knowledge Base article](#).

My Sitecore installation failed while I was using Skype

- If you use Skype while you are installing Sitecore 9.0, 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](#).

Chapter 8

Appendix

This chapter contains supplementary instructions that help you configure your environment such as file permissions, performance counters, and other information.

This chapter contains the following sections:

- Access Rights
- Certificates

8.1 Access Rights

8.1.1 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.

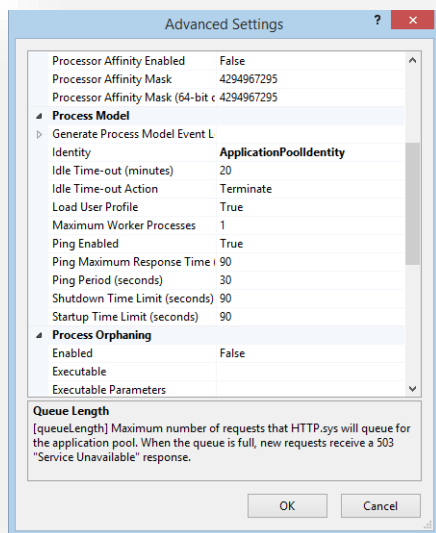
Note

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

xDB and *xConnect* only support Certificate Authentication as described in the chapter *Production Environment Setup*.

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 XP, add the domain user to the *IIS_IUSRS* group.
For more information about changing the permissions for the *IIS_IUSRS* group, see the section *File System Permissions*.
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=Sandbox6_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" />
  <add name="reporting" connectionString="Data
Source=<Data-Source>;Database=Sandbox6_Analytics;Trusted_Connection=True" />
</connectionStrings>
```

Note

If you use the Sitecore Experience Database (xDB), the configuration is the same for the *Reporting* database.

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

8.1.2 Windows Performance Counters

Sitecore XP 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*.

8.2 Certificates

8.2.1 References to the Certificates

Client certificates

When you have installed Sitecore XP 9.0, you can see the thumbprint values of the `XConnectCert` parameters in the following connection strings in the `\App_config` folder:

- `ConnectionStrings.config` for Sitecore and xConnect roles:

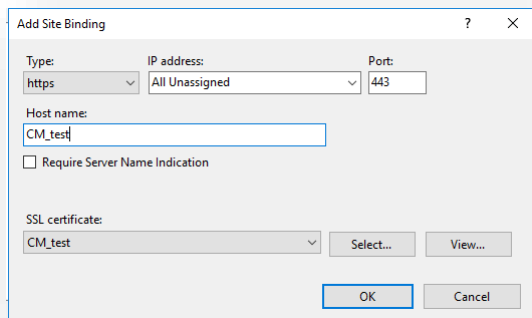
```
<add name="xconnect.collection.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;FindValue=738F45F610221990DA2FE059E1D8C2ECCB5067F2" />
```

- `AppSettings.config` file for the xConnect roles:

```
<add key="validateCertificateThumbprint" value="738F45F610221990DA2FE059E1D8C2ECCB5067F2" />
```

Server certificates

When you have installed Sitecore XP 9.0, you have an IIS site named `CM_test` with HTTPS binding and an associated SSL certificate called `CM_test`.



8.2.2 Use and 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 Client 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**.

Sitecore Experience Platform 9.0 Update 2

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.

For more information about the appropriate roles and the role certificate thumbprints that must be updated, read the section *Set Up Client Certificates*.
11. Update the thumbprint values in all of the certificate connection strings on the following instances:

XP Single (XPo)	XP Scaled (XP1)
Sitecore	Content Delivery
xConnect	Content Management
	Processing
	Marketing Automation Engine
	xDB Automation Operations

For example, on each XP Single (XPo) Sitecore instance, update the thumbprint value in these connection strings:

```
<add name="xconnect.collection.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;FindValue=859E88DC0692BA1583145223C455F186937C0D62" />

<add name="xdb.referencedata.client.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;FindValue=859E88DC0692BA1583145223C455F186937C0D62" />

<add name="xdb.marketingautomation.reporting.client.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;FindValue=859E88DC0692BA1583145223C455F186937C0D62" />

<add name="xdb.marketingautomation.operations.client.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;FindValue=859E88DC0692BA1583145223C455F186937C0D62" />
```

12. In the xConnect root folder, in the `\App_Config\AppSettings.config` file, update the thumbprint value in the following setting:

```
<add key="validateCertificateThumbprint" value="859E88DC0692BA1583145223C455F186937C0D62" />
```

You must update this setting on the following servers:

XP Single (XPo)	XP Scaled (XP1)
xConnect	xConnect Collection
	XConnect Search
	xDB Reference Data

	xDB Automation Reporting
	xDB Automation Operations

13. Restart IIS on every computer that you configured to use a new client certificate.

8.2.3 Configure Sitecore to Use New Server Certificates

1. Replace all of the old server certificates with new server certificates on each server with the XP Scaled (XP1) role.

Note

The common name field (CN) must be the same as your instance name.

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

8.3 Collations

All of the databases in Sitecore XP use the *SQL_Latin1_General_CP1_CI_AS* collation except the *Reference Data* database that uses the case sensitive *Latin1_General_CS_AS* collation.

This is because comparisons within the *Reference Data* database are case sensitive and they are not case sensitive in the other databases.