

Sitecore Experience Platform Installation Guide

Sitecore Experience Platform 9.0

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	How to Use This Installation Guide	5
1.1.3	Installation Process Overview	6
Chapter 2	Prerequisites and Requirements	7
2.1	Sitecore Hosting Environment Requirements	8
2.1.1	IIS Requirements	8
2.1.2	Operating System Requirements	8
2.1.3	.NET Framework Requirements	8
2.1.4	Visual Studio Requirements for Custom Solutions	9
2.1.5	Database Requirements	9
2.1.6	Search Indexing Requirements	9
2.1.7	Antivirus Software Considerations	9
2.1.8	Hardware Requirements for a Server Running a Single Sitecore Installation	10
2.2	Sitecore Client Requirements	11
2.2.1	Software Requirements	11
2.2.2	Hardware Requirements	11
2.3	Sitecore Installation Prerequisites	12
2.3.1	File System Permissions	12
2.3.2	Prerequisites for the Sitecore Installation Framework	13
2.3.3	Enable Contained Database Authentication	14
2.3.4	Installing Solr	14
Chapter 3	Preparing the Environment for Deployment	16
3.1	Use and Configure Your Topology	17
3.1.1	Choose Your Topology	17
3.2	Set Up the Sitecore Installation Framework Module	21
3.2.1	Installing the Installation Framework Module Using MyGet	21
3.2.2	Manual Installation	21
3.2.3	Validate the Installation	22
3.2.4	Customizing the Sitecore Installation Framework	23
3.2.5	Sitecore Fundamentals Dependency	23
3.3	Install a Local Instance of Sitecore	24
3.4	Install an Instance of Sitecore Remotely	25
Chapter 4	Production Environment Setup	26
4.1	Sitecore XP Security Requirements: SSL	27
4.1.1	Set Up Server Certificate SSL Authentication on IIS	27
4.1.2	Set Up Client Certificates	28
4.1.3	Set Up a SSL Certificate for Solr	31
4.2	Install Sitecore XP	32
4.2.1	Specifying the Certificates during Installation	33
Chapter 5	Local Environment Setup	34
5.1	Install Prerequisites	35
5.1.1	Download the Configuration Files	35
5.1.2	Set Up the Sitecore Installation Framework	35
5.2	Install Sitecore XP	36
5.2.1	Edit and Run the Installation Script	36
Chapter 6	Sitecore XP Post-Installation Steps	38
6.1	Add a Recognized User to the xDB Shard Databases	39
6.2	Rebuild the Search Indexes and the Link Databases	41
6.3	Deploy Marketing Definitions	42
6.4	Content Expiration	43
6.5	Tracking Configuration	45

Sitecore Experience Platform Installation Guide

6.6	Warm up the Servers.....	46
Chapter 7	Troubleshooting	47
7.1	Common Issues	48
Chapter 8	Appendix	49
8.1	Windows Authentication for Sitecore.....	50
8.1.1	References to the Certificates	51
8.1.2	Administrator Permissions	52
8.1.3	Azure Cloud Deployment Topologies.....	52

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. 171002 (Initial Release).

The document contains the following chapters:

- **Chapter 1 – Introduction**
An introduction to the Sitecore Experience Platform 9.0 installation process.
- **Chapter 2 – Prerequisites and Requirements**
An outline of the installation requirements for Sitecore XP.
- **Chapter 3 – Preparing the Environment for Deployment**
Preparing your environment for the deployment of Sitecore.
- **Chapter 4 – Production Environment Setup**
Steps to install Sitecore onto a scaled out, multiple server role environment.
- **Chapter 5 – Local Environment Setup**
For developers installing Sitecore on a local environment.
- **Chapter 6 – Sitecore XP Post-Installation Steps**
Information about necessary actions after installation.
- **Chapter 7 – Troubleshooting**
Answers to common problems you may encounter while installing Sitecore.
- **Chapter 8 – Appendix**
Additional referential 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

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 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 scaling Sitecore and security hardening, see the [Scaling and Architecture Guide](#) and the [Security hardening](#) sections on the Sitecore Documentation site – <https://doc.sitecore.net/>.

1.1.2 How to Use This Installation Guide

To install the Sitecore Experience Platform successfully, ensure that you meet all the prerequisites listed in this manual, have installed all the required software, and configured your environment accordingly.

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.

If you are installing Sitecore to a production environment, you must follow the chapter Production Environment Setup.

If you are installing Sitecore to a local environment, you must follow the chapter Local Environment Setup.

Starting with Sitecore 9.0, you must use the Sitecore Installation Framework to install Sitecore. The framework deploys Web Deploy Packages (WDP) by passing parameters to configuration files through a Microsoft® PowerShell module.

This installation guide explains how to use the Sitecore Installation Framework to install Sitecore in a production or local environment. If you want to customize your configuration, refer to the Sitecore Installation Framework Configuration Guide.

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

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

1.1.3 Installation Process Overview

This section gives an overview of the installation process using the Sitecore Installation Framework.

To install an instance of Sitecore:

- Ensure you meet the software and hardware requirements as described in the chapter *Prerequisites and Requirements*.
- Install the Sitecore Installation Framework Microsoft PowerShell® module in the computer where you will install Sitecore, as described in the section *Set Up the Sitecore Installation Framework*.
- Locate the web deploy packages and configuration files for your chosen topology as described in the *Download the Web Deploy Packages and Environment Configuration Files* section.
- Configure the installation parameters as described in the *Configure Parameters in the Environment Configuration Files* section.
- If necessary, you can customize your configuration as described in the Sitecore Installation Framework Configuration Guide.
- Deploy Sitecore to either a local or a remote computer.

Chapter 2

Prerequisites and Requirements

This chapter describes the prerequisites, hardware, and software requirements for Sitecore XP 9.0. In this document, the term "server" refers to a Windows OS Web server and the term "host" refers to an installation of Sitecore XP on either a server or a workstation.

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

Important

If you use IIS 8.5, you must create your certificates manually, as Sitecore Fundamentals does not support 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 pointing 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

Ensure that you 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 Visual Studio Requirements for Custom Solutions

Sitecore XP 9.0 requires Microsoft Visual Studio 2015 or later, targeting .NET Framework 4.6.2.

2.1.5 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).

Important

Sitecore XP 9.0 rev. 171002 does not currently support MongoDB or Oracle databases for the Experience Database (xDB). Support will be added in future versions of Sitecore.

Note

You can also use Windows authentication to connect to the SQL server. For more information about connecting to the SQL server, see the *Windows Authentication for Sitecore* section.

2.1.6 Search Indexing Requirements

Sitecore XP 9.0 supports several indexing providers. For information about installing or managing them in a Sitecore context, refer to the Sitecore Documentation site – <https://doc.sitecore.net>.

Sitecore supports the following indexing providers:

- Solr 6.6.2
For the XP0 and XP1 topologies, Solr is the default search provider.
- Lucene
For the XM1 topology, Lucene is the default search provider.

Note

Lucene only supports Content Search and does not support xConnect.

- Azure Search
The Azure Search provider is supported for Azure Cloud PAAS deployments only.

2.1.7 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.

To ensure optimal performance, exclude the following folders from scanning 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.

2.1.8 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 54, 55, 56.
- Google Chrome 60, 61.
- Microsoft Edge 40.
- Apple Safari 9, 10, 11.

Note

Sitecore XP 9.0 rev. 171002 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, due to bug fixes and vulnerabilities, newer browser revisions are continually released. Sitecore will support the latest revisions of these browsers.

For more information about configuring a browser, see the [Browser Configuration Reference](#) document on the Sitecore Developer Network.

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 begin your installation of Sitecore, ensure you have fulfilled 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 required rights 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 following table contains the default account used for processing 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 be able to load .NET runtime and ASP.NET resources used for processing the ASP.NET requests, the worker process hosting 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. The Microsoft site 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 Installation Guide

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 the Sitecore Installation Framework

This section describes the prerequisites to install the Sitecore Experience Platform in an on-premises environment using the Installation Framework. To use the Sitecore Installation Framework, you must download and install:

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

Note

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

The Sitecore Installation Framework has the following requirements:

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 installation of Web Deploy Packages.	You install the tool using the Web Platform Installer.
URL Rewrite 2.1	Supports the use of the Sitecore Installation Framework.	You install the tool using the Web Platform Installer.

Requirement	Feature	Details
Microsoft SQL Server Data-Tier Application Framework (DAC Fx) version 17.1	Supports installation of .dac files in Web Deploy Packages.	<p>You install the framework using the Web Platform Installer.</p> <p>Note If the 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>
Microsoft SQL Server Transact-SQL ScriptDom	ScriptDom provides Microsoft SQL Server-specific Parser and Script Generator support.	You install ScriptDom using the installer for Visual Studio .

Clear the Web Platform Installer download cache

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

To clear the Web Platform Installer download cache:

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

2.3.3 Enable Contained Database Authentication

To enable the SQL server to allow the creation of users when using the Sitecore Installation Framework, 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

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

2.3.4 Installing Solr

The Sitecore Experience Platform 9.0 rev. 171002 supports Solr, Lucene, and Azure Search as search providers. If you want to use a search index that works in both Analytics and ContentSearch, we recommend that you use Solr.

Sitecore Experience Platform Installation Guide

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.

By default, the Sitecore Experience Platform is secure, therefore you must 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](#).

Installing the Solr Certificate

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

- Content Management
- xConnect Collection Search

Chapter 3

Preparing the Environment for Deployment

This chapter describes all the steps necessary 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
- Install a Local Instance of Sitecore
- Install an Instance of Sitecore Remotely

3.1 Use and Configure Your Topology

Before you run the installation of Sitecore, you must choose the topology or the instance that you want to install. By default, Sitecore provides XP0, XM1, and XP1 topologies.

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 XM0 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 Topology

The following table describes the three types of deployment topologies available: XP0, XM1, and XP1.

Note

Azure Cloud has different deployment topology names. For more information, see the section *Azure Cloud Deployment Topologies* in the appendix.

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

Deployment topology	Description
XP0	<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 <u>combines the server roles</u>.</p> <p>Note For security and scalability reasons, in production environments, it is best practice to use the XM1 or XP1 configuration.</p>
XM1	<p>The Sitecore Experience Management configuration (CMS-only mode) running both the Content Delivery (CD) and Content Management (CM) roles.</p> <ul style="list-style-type: none">Use this environment when you are not planning to use the Analytics and Marketing features of the Sitecore Experience Platform. <p>Note When you select this topology, xDB and xConnect are not available.</p>

Deployment topology	Description
XP1	<p>The Sitecore Experience Platform configuration running the following separated server roles: Content Delivery, Content Management, Processing, Reporting, xConnect Collection, xConnect Collection Search, xDB Reference Data, xDB Automation Operations, xDB Automation Reporting.</p> <p>Use this environment 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

For both Private and Shared Session states, Sitecore is deployed with an InProc session state provider by default. 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 [Server Role Configuration Reference Guide](#).

Moreover, the Sitecore Installation Framework does not automatically deploy the Sessions database. To deploy the Sessions database, follow the steps in the [Configuring a Private Session State Database Using the SQL Server Provider](#) topic.

Scalability

To achieve better performance, cope with greater website demand and large amounts of website traffic, you have several scalability options. For more information on these options, read the [Scaling and Architecture Guide](#).

Download the Web Deploy Packages and Environment Configuration Files

After selecting your deployment topology (XP0, XM1, or XP1), you must download the corresponding web deploy packages which also contain the environment configuration files from the Sitecore Downloads page – <https://dev.sitecore.net>.

XP0

The following Web Deploy Packages are required for XP0 topologies:

- `sitecore 9.0.0 rev. 171002 (OnPrem)_single.scwdp.zip`
- `sitecore 9.0.0 rev. 171002 (OnPrem)_xp0xconnect.scwdp.zip`

The following environment configuration files are required for XP0 topologies:

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

XM1

The following Web Deploy Packages are required for XM1 topologies:

- `sitecore 9.0.0 rev. 171002 (OnPrem)_cd.scwdp.zip`
- `sitecore 9.0.0 rev. 171002 (OnPrem)_cm.scwdp.zip`

Sitecore Experience Platform Installation Guide

The following environment configuration files are required for XM1 topologies:

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

XP1

The following Web Deploy Packages are required for XP1 topologies:

- `sitecore 9.0.0 rev. 171002 (OnPrem)_cd.scwdp.zip`
- `sitecore 9.0.0 rev. 171002 (OnPrem)_cm.scwdp.zip`
- `sitecore 9.0.0 rev. 171002 (OnPrem)_prc.scwdp.zip`
- `sitecore 9.0.0 rev. 171002 (OnPrem)_rep.scwdp.zip`
- `sitecore 9.0.0 rev. 171002 (OnPrem)_xplcollection.scwdp.zip`
- `sitecore 9.0.0 rev. 171002 (OnPrem)_xplcollectionsearch.scwdp.zip`
- `sitecore 9.0.0 rev. 171002 (OnPrem)_xplmarketingautomation.scwdp.zip`
- `sitecore 9.0.0 rev. 171002 (OnPrem)_xplmarketingautomationreporting.scwdp.zip`
- `sitecore 9.0.0 rev. 171002 (OnPrem)_xplreferencedata.scwdp.zip`

The following environment configuration files are required for XP1 topologies:

- `sitecore-solr.json`
- `xconnect-solr.json`
- `sitecore-XP1-cd.json`
- `sitecore-XP1-cm.json`
- `sitecore-XP1-prc.json`
- `sitecore-XP1-rep.json`
- `xconnect-xpl-collection.json`
- `xconnect-xpl-collectionsearch.json`
- `xconnect-xpl-marketingautomation.json`
- `xconnect-xpl-marketingautomationreporting.json`
- `xconnect-xpl-referencedata.json`

Configure Parameters in the Environment Configuration Files

The environment configuration files are templates that provide a basis for deployment of various Sitecore Experience Platform configurations with support for:

- Creating an IIS Application Pool
- Creating an IIS website
- Installation of Web Deploy Packages
- Configuration of File Permissions

If necessary, you can configure the default parameters within each of the configuration files for your chosen topology.

To configure parameters in an environment configuration file:

1. In a text editor, open the relevant configuration file and find the `Parameters` block.
2. For each parameter, if it has a default value, check whether you must 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 result in a prompt for that information.

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 is a Microsoft® PowerShell module that supports local and remote installations of Sitecore, and it 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 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. It is automatically installed when you install the Sitecore Installation Framework.

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.

Important

If you use IIS 8.5, you must create your certificates manually, as Sitecore Fundamentals does not support IIS 8.5.

3.2.1 Installing 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. The Sitecore Installation Framework is available through the Sitecore Gallery.

To set up the Sitecore Installation Framework:

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

Sitecore 9.0 must be installed with the Sitecore Installation Framework 1.x. You cannot install Sitecore 9.0 with Sitecore Installation Framework 2.x.

3.2.2 Manual Installation

The Sitecore Installation Framework is also provided as a .ZIP package. When you manually install the Sitecore Installation Framework, you must also download and install the Sitecore Fundamentals package.

You can download the Sitecore Fundamentals and the Sitecore Installation Framework 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 continue the installation of the Sitecore Installation Framework, 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**.

Extracting the Sitecore Installation Framework

The installation path that you must use depends on the location where you want to install the Sitecore Installation Framework. 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 plan to install the Sitecore Installation Framework module 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 the Sitecore Installation Framework to either 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 the Sitecore Installation Framework 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 the Sitecore Installation Framework, you can validate its installation to confirm that it is available for use. This procedure is optional.

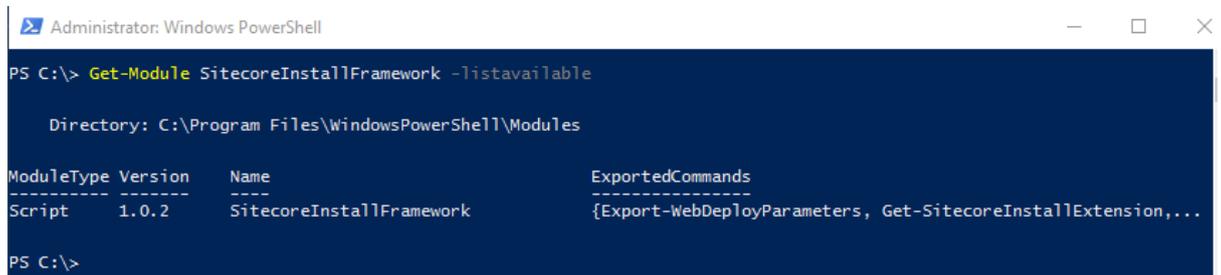
Note

This validation only works if you have installed the Sitecore Installation Framework 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      1.0.2       SitecoreInstallFramework          {Export-WebDeployParameters, Get-SitecoreInstallExtension, ...}

PS C:\>
```

3.2.4 Customizing the Sitecore Installation Framework

The Sitecore Installation Framework 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 you can extend the installation framework, read the *Customize the Sitecore Installation Framework* section in the Sitecore Installation Framework Configuration Guide.

3.2.5 Sitecore Fundamentals Dependency

The Sitecore Installation Framework 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.

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

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

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

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

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

3.3 Install a Local Instance of Sitecore

The following is an example of a local instance installation performed with the Sitecore Installation Framework.

To install Sitecore to 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 configuration. For example, using the `sitecore-XP0.json` configuration:

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

Optionally, parameters declared in configurations 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 details and further guidance on the parameters that can be passed to the `Install-SitecoreConfiguration` cmdlet, you can run the following cmdlet in a PowerShell command line: `Get-Help Install-SitecoreConfiguration`.

Note

You can also run the `Install-SitecoreConfiguration` cmdlet by using the alias `scinst`.

3.4 Install an Instance of Sitecore Remotely

In addition to a local deployment, PowerShell Remoting lets you deploy your new Sitecore instance to 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 Instance Installation

To start a remote instance deployment:

1. Install the Sitecore Installation Framework 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 required packages and configurations to the remote computer, and then in a PowerShell command line 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

Production Environment Setup

This chapter outlines the steps required to install Sitecore XP 9.0 rev. 171002 in a production environment.

This chapter contains the following sections:

- Sitecore XP Security Requirements: SSL
- Install Sitecore XP

4.1 Sitecore XP Security Requirements: SSL

The Sitecore Experience Platform is designed to be secure by default. This means that you must implement HTTPS across the Platform.

Server Certificate Authentication

All communication between Sitecore instances, including xConnect web services, and the Solr search provider occurs over 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 discriminate who can connect to the server.

Client Certificate Authentication

xConnect server roles support an additional layer of security, referred to as SSL Client Certificate Authentication. Client Certificate Authentication validates that the individual HTTP client is authorized to connect to the HTTP server. 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 Content Management 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, do not use self-signed certificates in production environments.

4.1.1 Set Up Server Certificate SSL Authentication on IIS

You must obtain and install server certificates before you run the Sitecore Installation Framework. The Microsoft site 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 (XM1, XP0, or XP1):

XM1	XP0	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.

To install server certificates:

1. Install the server authentication certificate in the system certificate store `Certificates (Local Computer) \Personal` folder. For information on how to install the certificate, read the *To import a private key certificate* section on [MSDN](#).
2. If you created a self-signed certificate, install the self-signed authority certificate for the SSL certificate in the `Certificates (Local Computer) \Trusted Root Certification Authorities` folder.

Note

For XM₁, and XP₀ topologies, it is assumed that there is only one SSL certificate for each IIS instance that covers multiple application roles. For 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 XP₁, after you obtain all server certificates, you must install them on the required servers according to the following table:

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>

4.1.2 Set Up Client Certificates

You must obtain and install client certificates before running the Sitecore Installer Framework.

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 and purposes to isolate servers, in the event of a key being compromised.

Sitecore Experience Platform Installation Guide

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:

XM1	XPo	XP1
None	xConnect	xConnect Collection xConnect Collection Search xDB Reference Data xDB Automation Operations xDB Automation Reporting

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 on how to do this, see the *To import a private key certificate* section on [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 `Certificates (Local Computer)\Trusted Root Certification Authorities` folder.

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:

XPo

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
xConnect	xConnect Client	xdb.referencedata.client.certificate xconnect.collection.certificate

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 Reference Data	xdb.referencedata.client.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.

4.1.3 Set Up a SSL Certificate for Solr

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

You must install Solr SSL certificates according to the following table:

XM1	XP0	XP1
<i>None required</i>	xConnect Sitecore	Content Management xConnect Collection Search

4.2 Install Sitecore XP

Once you have obtained the required certificates, you can run the Sitecore Installation Framework and install Sitecore. You can install any of the configurations into 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](https://doc.sitecore.net) in the Sitecore Documentation site – <https://doc.sitecore.net>.

Important

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

To run the installation framework 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. Run the Solr cores installation by executing the following cmdlets:
 - o `Install-SitecoreConfiguration -Path "C:\SitecoreInstaller\Configurations\XConnect\Solr\xconnect-solr.json"`
 - o `Install-SitecoreConfiguration -Path "C:\SitecoreInstaller\Configurations\Platform\Solr\sitecore-solr.json"`
3. Run the server roles installation by executing the following cmdlets with the required parameters for your server roles:
 - o `Install-SitecoreConfiguration -Path "C:\SitecoreInstaller\Configurations\Platform\XP1\sitecore-XP1-cm.json"`
 - o `Install-SitecoreConfiguration -Path "C:\SitecoreInstaller\Configurations\Platform\XP1\sitecore-XP1-cd.json"`
 - o `Install-SitecoreConfiguration -Path "C:\SitecoreInstaller\Configurations\Platform\XP1\sitecore-XP1-prc.json"`
 - o `Install-SitecoreConfiguration -Path "C:\SitecoreInstaller\Configurations\Platform\XP1\sitecore-XP1-rep.json"`
 - o `Install-SitecoreConfiguration -Path "C:\SitecoreInstaller\Configurations\xConnect\XP1\xconnect-xp1-collection.json"`
 - o `Install-SitecoreConfiguration -Path "C:\SitecoreInstaller\Configurations\xConnect\XP1\xconnect-xp1-collectionsearch.json"`
 - o `Install-SitecoreConfiguration -Path "C:\SitecoreInstaller\Configurations\xConnect\XP1\xconnect-xp1-MarketingAutomationReporting.json"`
 - o `Install-SitecoreConfiguration -Path "C:\SitecoreInstaller\Configurations\xConnect\XP1\xconnect-xp1-MarketingAutomation.json"`

- o `Install-SitecoreConfiguration -Path "C:\SitecoreInstaller\Configurations\xConnect\XP1\xconnect-xp1-ReferenceData.json"`

4.2.1 Specifying the Certificates during Installation

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

By default, the Sitecore Installation Framework searches for the certificates in the following path:

- `Cert:\Localmachine\My`

If required, you can change the storage location.

Change the default location of 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 certificates

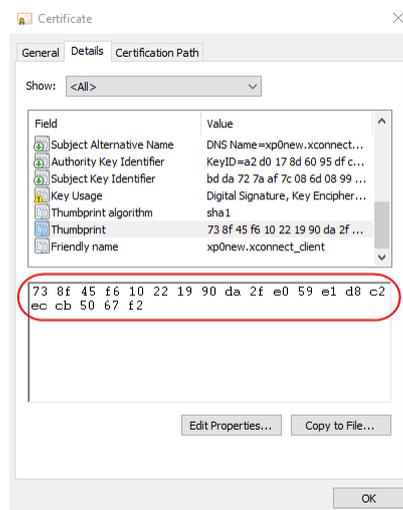
You must specify as parameters the names or thumbprints of certificates that you created and installed in previous steps in this guide. For example:

- For the *client* authentication certificate:

```
-XConnectCert "xConnect_client"
```

or

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



Note

You must specify the client certificate thumbprint parameter 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 5

Local Environment Setup

This chapter outlines the steps required to install Sitecore XP 9.0 rev. 171002 in a developer environment.

This chapter contains the following sections:

- Install Prerequisites
- Install Sitecore XP

5.1 Install Prerequisites

- Before you begin installing your local environment, ensure you read Chapters 2 and 3 in this guide to install all the prerequisite software and to prepare your environment.

Important

The Sitecore Fundamentals module does not support IIS 8.5. To install Sitecore while running IIS 8.5, you must install certificates manually.

5.1.1 Download the Configuration Files

You must download the relevant configuration files and the Web Deploy Packages (WDPs) that contain the XPo topology.

From the Sitecore Downloads page – <http://dev.sitecore.net>:

- Download the following configuration files:
 - Sitecore-XP0.json
 - XConnect-XP0.json
 - Sitecore-Solr.json
 - XConnect-Solr.json
 - XConnect-CreateCert.json
- Download the following WDPs:
 - Sitecore 9.0.0 rev. 171002 (OnPrem)_single.scwdp.zip
 - Sitecore 9.0.0 rev. 171002 (OnPrem)_xp0xconnect.scwdp.zip

5.1.2 Set Up the Sitecore Installation Framework

To set up the Sitecore Installation Framework:

1. Run a Microsoft PowerShell console as an administrator.
2. Add the Sitecore PowerShell repository as described in the *Installing the Installation Framework Module Using MyGet* section.
3. Install the Sitecore Installation Framework module by running the following cmdlet:

```
Install-Module SitecoreInstallFramework
```
4. Import the Sitecore Installation Framework by running the following cmdlet:

```
Import-Module SitecoreInstallFramework
```

5.2 Install Sitecore XP

To simplify your installation, you can install the Sitecore XPo topology automatically by running a PowerShell script. After you have edited and run the installation script, you must complete some post-installation steps.

You can download a .txt file containing the installation script from the Sitecore Downloads page – <https://dev.sitecore.net>.

5.2.1 Edit and Run the Installation Script

To edit and run the installation script:

1. 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 = "$PSScriptRoot\Sitecore 9.0.0 rev. 171002
(OnPrem)_xp0xconnect.scwdp.zip"
    LicenseFile = "$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 = "$PSScriptRoot\Sitecore 9.0.0 rev. 171002
(OnPrem)_single.scwdp.zip"
    LicenseFile = "$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
```

2. Save the script as `install.ps1` to the `C:\resourcefiles` folder.
3. In a PowerShell command line, run the following command:

```
C:\resourcefiles\install.ps1
```

Chapter 6

Sitecore XP Post-Installation Steps

This chapter describes the steps required to complete the installation after you install Sitecore XP 9.0 rev. 171002 with the Sitecore Installation Framework.

This chapter contains the following sections:

- Add a Recognized User to the xDB Shard Databases
- Rebuild the Search Indexes and the Link Databases
- Deploy Marketing Definitions
- Content Expiration
- Tracking Configuration

6.1 Add a Recognized User to the xDB Shard Databases

xConnect has three collection databases that require a user to be recognized and authenticated: the xDB Collection Shard Map Manager database, and two Shard databases. The shards are used to scale out the system as required by the solution being implemented.

After the Sitecore Installation Framework installs the databases, you must set up the correct user roles for authentication. To help you with this, Sitecore has created an example SQL script that you can use to provide a simple solution for the setup of a user role in this release.

The SQL script creates a SQL login and a user with the minimum privileges for each database.

Note

You must run this example SQL script after you have completed the installation.

To run the script:

1. Launch MS SQL Server Management Studio and connect to the database server as an administrator.
2. Open a new query window, and paste the following script:

```
:SETVAR DatabasePrefix xp1
:SETVAR UserName collectionuser
:SETVAR Password Test12345
:SETVAR ShardMapManagerDatabaseNameSuffix _Xdb.Collection.ShardMapManager
:SETVAR Shard0DatabaseNameSuffix _Xdb.Collection.Shard0
:SETVAR Shard1DatabaseNameSuffix _Xdb.Collection.Shard1
GO

IF(SUSER_ID('${UserName}') IS NULL)
BEGIN
    CREATE LOGIN [${UserName}] WITH PASSWORD = '${Password}';
END;
GO

USE [${DatabasePrefix}${ShardMapManagerDatabaseNameSuffix}]

IF NOT EXISTS (SELECT * FROM sys.database_principals WHERE name = N'${UserName}')
BEGIN
    CREATE USER [${UserName}] FOR LOGIN [${UserName}]
    GRANT SELECT ON SCHEMA :: __ShardManagement TO [${UserName}]
    GRANT EXECUTE ON SCHEMA :: ShardManagement TO [${UserName}]
END;
GO

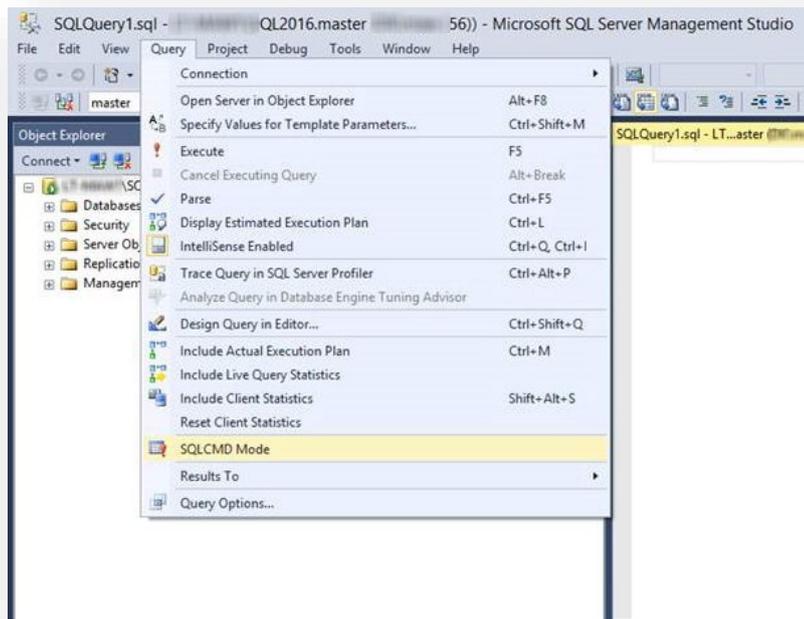
USE [${DatabasePrefix}${Shard0DatabaseNameSuffix}]

IF NOT EXISTS (SELECT * FROM sys.database_principals WHERE name = N'${UserName}')
BEGIN
    CREATE USER [${UserName}] FOR LOGIN [${UserName}]
    EXEC [xdb_collection].[GrantLeastPrivilege] @UserName = '${UserName}'
END;
GO

USE [${DatabasePrefix}${Shard1DatabaseNameSuffix}]

IF NOT EXISTS (SELECT * FROM sys.database_principals WHERE name = N'${UserName}')
BEGIN
    CREATE USER [${UserName}] FOR LOGIN [${UserName}]
    EXEC [xdb_collection].[GrantLeastPrivilege] @UserName = '${UserName}'
END;
GO
```

3. In the application bar, on the **Query** tab, click **SQLCMD Mode**.



4. In the script:
 - o Set the `DatabasePrefix` variable to the prefix that you used to install the Collection database. The default value is `XP1`.
 - o Set the `UserName` variable to the value that you used for the Collection Database Application User Name parameter. The default value is `collectionuser`.
 - o Set the `Password` variable to the value that you used for the Collection Database Application User Password parameter.
5. Run the script.

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

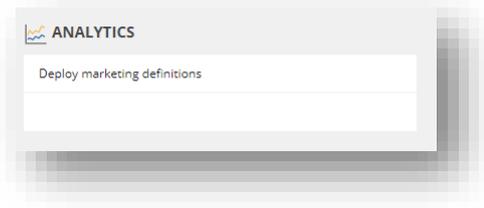
If you want to use the Sitecore Experience Marketing functionality, you must deploy marketing definitions.

Note

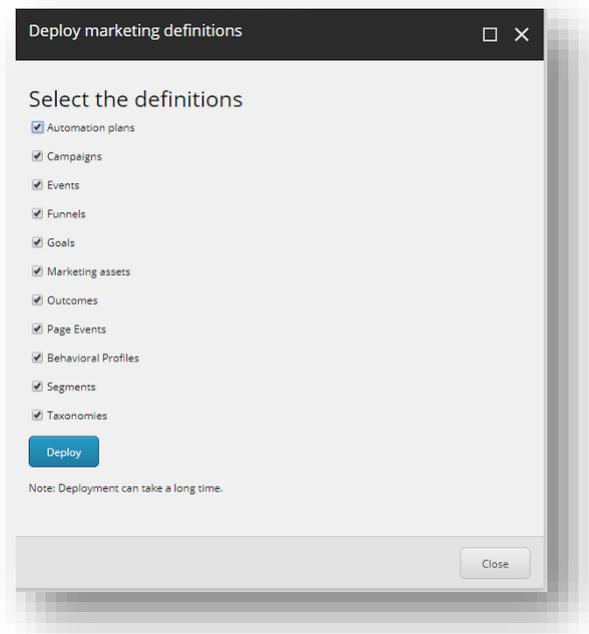
If you do not plan to use the Sitecore Experience Database (xDB), you do not have to perform these steps.

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**.



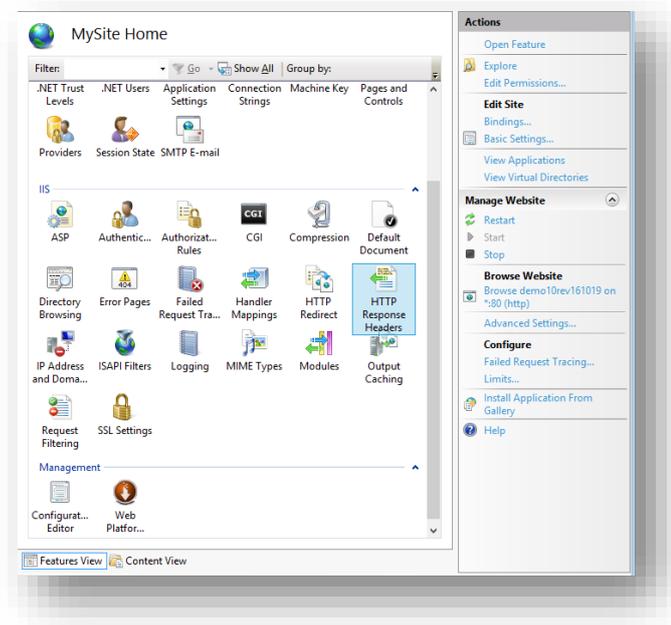
6.4 Content Expiration

The Expire Web content header (located in common HTTP Response Headers) is how IIS determines 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.

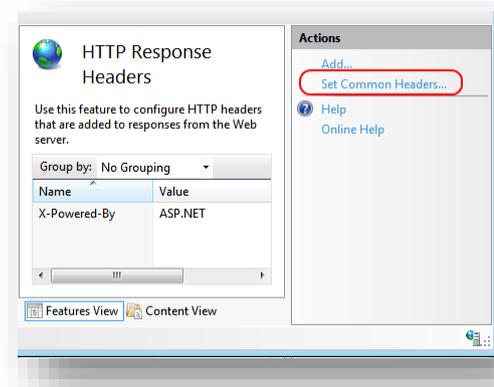
By setting Expire Web content to something other than *immediately*, you can reduce second-access load times by 50 –70%. This setting does not affect dynamically-generated content.

To enable the Expire Web content header in IIS:

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

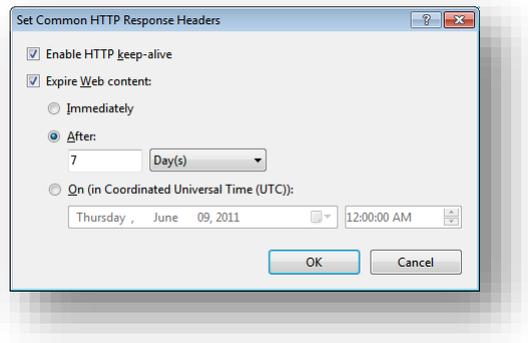


4. In the **Actions** panel, click **Set Common Headers...**



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

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



7. Click **OK**.

6.5 Tracking Configuration

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.

For information about security hardening, read the Security hardening section on the Sitecore Documentation site.

On the Sitecore Documentation site, you can read more about [security hardening](#) after you install Sitecore.

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, update the `Analytics.PerformLookup` setting and change the value to *False*.

Note

The `Analytics.PerformLookup` setting has a default value of *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, you must change the default salt value in the `Sitecore.Analytics.Tracking.config` file, in the `geoIpManager` section.

3. You must restart IIS.

6.6 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](#).

Chapter 7

Troubleshooting

This chapter can help you resolve the most common problems encountered during your Local or Production environment installation of Sitecore XP 9.0.

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 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 have pasted your thumbprint into a PowerShell command line window, and that you have 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 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 that 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

- When using Skype while you are installing Sitecore 9.0, it is possible that your xConnect installation may fail. This is due to Skype and Sitecore xConnect using port 443, which interferes with the installation. If this happens, you must change your Skype configuration as outlined in the following [article](#).

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:

- Windows Authentication for Sitecore

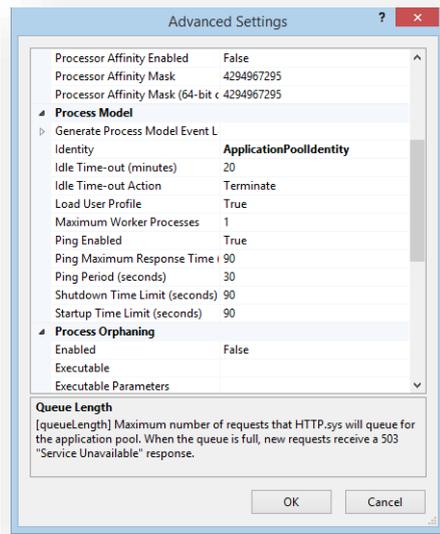
8.1 Windows Authentication for Sitecore

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

This only applies to the Core, Master, Web and Reporting SQL Databases, and not to the xDB and xConnect. xDB and xConnect only support Certificate Authentication as described in the *Production Environment Setup* chapter.

To configure Sitecore to use Windows Authentication:

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



2. In SQL Server, register the domain user and grant the appropriate security permissions to the Sitecore databases for the domain user.
3. On the computer that hosts Sitecore XP, add the domain user to the `IIS_IUSRS` group.

Note

You must change the permissions for the `IIS_IUSRS` group according to the instructions in the

File System Permissions for ASP.NET Requests and *File System Permissions for System Folders* sections in this guide.

4. 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.

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

Windows Performance Counters

Sitecore XP contains a 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.

The Microsoft site describes more about [Application Pool Identity](#).

Note

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

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

To prevent the update of the performance counters:

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

8.1.1 References to the Certificates

Client certificates

After you have completed the installation of Sitecore XP 9.0, the thumbprint values of the `-XConnectCert` parameters are present in the following connection strings in the `\App_config` folder:

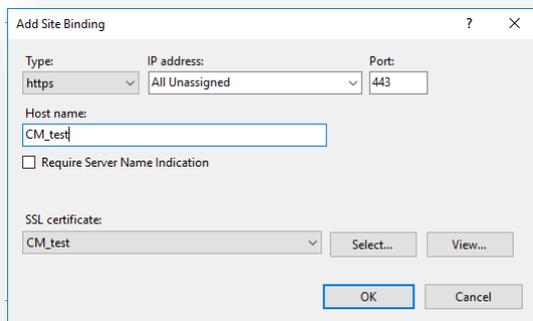
- `ConnectionStrings.config` for Sitecore and xConnect roles:

```
<add name="xconnect.collection.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=Fi
ndByThumbprint;FindValue=738F45F610221990DA2FE059E1D8C2ECCB5067F2" />
```
- `AppSettings.config` file for the xConnect roles:

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

Server certificates

After you have completed installation, you will have an IIS site named `CM_test` with HTTPS binding and an associated SSL certificate called `CM_test`.



8.1.2 Administrator Permissions

To run the Sitecore Installation Framework, you must run PowerShell as an Administrator.

Important

If you try to use the Sitecore Installation Framework in a non-admin window, one of the following error messages appears and you cannot install Sitecore.

```
Import-Module : The required module 'SitecoreFundamentals' is not loaded. Load the module or remove the module from 'RequiredModules' in the file
```

```
import-module : The script 'SitecoreFundamentals.psml' cannot be run because it contains a "#requires" statement for running as Administrator. The current Windows PowerShell session is not running as Administrator. Start Windows PowerShell by using the Run as Administrator option, and then try running the script again.
```

8.1.3 Azure Cloud Deployment Topologies

In Azure Cloud deployments, the [XP0 and XP1 On-Premises packages](#) are referred to as XP Single and XP Scaled respectively. This naming difference will be resolved in a later release.

The following table explains the mapping of Azure Cloud packages to On-Premise packages:

Packages	Description
Packages for XP Single	Packages for XP Single / Single (XP0) Instance configuration.
Packages for XP Scaled	Individual packages for each of the dedicated XP Scaled (XP1) roles.
Packages for XM Scaled	Individual packages for each of the dedicated XM Scaled (XM1) roles.