Sitecore Experience Platform Installation Guide

Sitecore Experience Platform 9.1.1

Installation guide for administrators and developers

sitecore®
Own the experience™

Table of Contents

Chapter 1

Introduction

This guide describes how to install Sitecore Experience Platform 9.1 Update 1.

The document contains the following chapters:

- **Chapter 1 – Introduction**
  An introduction to the installation process for Sitecore XP 9.1.1.
- **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 – Appendix**
  Additional reference information to help you install Sitecore and answers to common problems you may encounter while installing Sitecore.

1.1 Getting Started

Sitecore is divided into two distinct product areas:

- Sitecore Experience Management (XM) – the content management, personalization features, and the security token service.
- Sitecore Experience Platform (XP) – the content management, personalization, the security token service, 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, EXM, FXM, and Experience Analytics.
- Experience content management – with applications such as the Experience Editor and the Experience Explorer.
- Sitecore Identity server – an OpenID Connect compliant security token service (STS).

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

For more information about Sitecore XP or XM, see the Sitecore Documentation.

Note
This document only describes how to install Sitecore XP and XM in an on-premise environment. For information about how to deploy Sitecore XP and XM in Azure, see the Getting started with Sitecore Azure Toolkit documentation.

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.com.

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 9.1.1.

This chapter contains the following sections:

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

2.1 Sitecore Hosting Environment Requirements

Sitecore XP 9.1.1 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

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

For the Sitecore Identity server, you must install:

- .NET Core 2.1.3 Windows Hosting Module or later

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 9.1.1 is only compatible with the client and server operating systems that support .NET Framework 4.7.1 or later.

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

- Windows Server 2016
- Windows 10 (32/64-bit)

Important
You must enable the Transport Layer Security (TLS) protocol version 1.2 on all of your Sitecore XP CM and Dedicated Dispatch servers (DDS).

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

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

2.1.3 .NET Requirements

Sitecore XP 9.1.1 requires .NET Framework 4.7.1.

Sitecore Identity server requires .NET Core 2.1.3 Windows Hosting Module or later.

You must apply any available updates to the .NET Framework on 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 9.1.1 requires Microsoft Visual Studio 2015 or later.

2.1.6 Database Requirements

Sitecore 9.1.1 supports the following database servers:

- Microsoft SQL Server 2017, 2016 SP2

  Supports the XM databases and is the required for the Experience Database (xDB).

- Microsoft SQL Server 2014 SP2

  Only supports XM databases and does not support the Experience Database (xDB).

- MongoDB Server 3.6.6

  Required if you are going to use MongoDB for the *Collection* database or as a Session State Provider.

  Note
  Sitecore XP 9.1.1 does not support the MMAPv1 storage engine because it does not support retryable writes. For more information about retryable writes, see the section *MongoDB High* Availability.

Important
Sitecore XP 9.1.1 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

SQL Server Contained Database Authentication

To enable the *Contained Database Authentication* server configuration option on SQL Server, run the following script:

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

Database Collation

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.

2.1.7 Search Indexing Requirements

Sitecore XP 9.1.1 supports Solr, Lucene, and Azure Search as search providers.

- Solr 7.2.1 – `http://archive.apache.org/dist/lucene/solr/7.2.1`

  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.

For information about installing or managing them in a Sitecore context, see the Sitecore documentation.

If you want to use a search provider that works in both analytics and content search, we recommend that you use either Solr or Azure Search.

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 9.1.1 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 9.1.1 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 9.1.1 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
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 Microsoft's documentation.

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 *Modify* access rights to the `\temp` folder for the ASP.NET and/or IUSR accounts.
%WINDIR%\Globalization\	Modify	Required for registering custom languages by the .NET Framework correctly.
%PROGRAMDATA%\Microsoft\Crypto	Modify	Required for storing cryptographic keys used for encrypting/decrypting data.

These variables have the following default values:

Variable	Default value
%WINDIR%	`C:\Windows`
%PROGRAMDATA%	`C:\ProgramData for IIS 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 9.1.1 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 Install Framework

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

- Microsoft PowerShell® version 5.1 or later

2.4 The Prerequisites

You can install the perquisites automatically or manually.

2.4.1 Automated Installation of Prerequisites

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

This file does not install:

- Microsoft SQL Server
- MongoDB
- Solr
- Microsoft Machine Learning Server

For more information about how to use this file, see the section *Install the Prerequisites*.

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

2.4.2 Manual Installation of the Prerequisites

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

- Web Platform Installer 5.0
- IIS including the Web administration PowerShell Module
- SQL PowerShell Module
- Microsoft sqlcmd Utility

  The sqlcmd Utility is part of SQL Server. If you do not have SQL Server installed on the computer that is assigned a Sitecore server role, you must download and install it.

The Sitecore server roles require:

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

Requirement	Feature	Details
Microsoft SQL Server Data-Tier Application Framework (DacFx) version 2017 (x86 and x64)	Supports the installation of `.dac` files in Web Deploy Packages.	Download and install DacFx x86. Download and install DacFX x64. This must be installed on servers that have been assigned a Sitecore server role and where you are going to install DAC packages: In the XM Scaled topology • Content Management In the XP Scaled topology • Content Management • Processing • Collection • Reference Data • Sitecore Cortex™ Processing Engine • Sitecore Cortex™ Reporting service To ensure that DacFx works correctly, you must install its system requirements including SQLSysCLRTypes.msi. If you are running an x64 environment, you must install both the x64 and x86 versions of DacFx and SQLSysCLRTypes. **Note** If DACFx fails to install, you can see the following error message when you use the framework: *The SQL provider cannot run with dacpac option because of a missing dependency. Please make sure that DACFx is installed.* For information about how to resolve this error, see this Sitecore Knowledge Base article.

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**.

2.4.3 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:

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

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

2.4.4 Install Solr

Sitecore Experience Platform 9.1.1 supports Solr, Lucene, and Azure Search as search providers. If you want to use a search provider that works in both analytics and content search, we recommend that you use either Solr or Azure Search.

- Lucene is only supported for content search and you should not use Lucene on a production server.

  Sitecore xConnect does not support Lucene.
- Solr is supported for both content search and analytics search.
- Azure Search is supported for both content search and analytics search – but only on Microsoft Azure and not in on-premise installations.

  In Paas solutions, you can use Azure Search, Solr, or SolrCloud in Azure.

For more information about the supported search providers, see Using Solr, Lucene, or 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
- xConnect Indexer service – if you install it on a separate server.

  For more information about the xConnect Indexer service role, see this article.

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:

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

3.1 Choose 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 (XP0)
- 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 an XM Single topology, you must install the CM instance from the XM Scaled topology and then in the `web.config` file, specify the following setting:

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

3.1.1 Choose Your On-premise Topology

The following table describes the three topologies that are available: XP Single (XP0), XM Scaled (XM1), and XP Scaled (XP1).

Note
Azure Cloud supports additional 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 (XP0)	The Sitecore Experience Platform, running as three single instances: Sitecore, xConnect, and Sitecore Identity server. The Experience Database (xDB) is partially included in the Sitecore and xConnect instances. Use this topology for local development and testing. **Note** For security and scalability reasons, in production environments, it is best practice to use the XM Scaled (XM1) or XP Scaled (XP1) configuration.
XM Scaled (XM1)	The Sitecore Experience Management configuration (CMS-only mode) running the Content Delivery (CD), Content Management (CM) server roles and the Sitecore Identity server. Use this topology if you are not planning to use the Analytics and Marketing features of the Sitecore Experience Platform. **Note** When you select this topology, xDB and xConnect are not available.

Deployment topology	Description
XP Scaled (XP1)	The Sitecore Experience Platform configuration running the following separated server roles: • Content Delivery, • Content Management, • Content Management + DDS (optional), • Sitecore Identity, • Processing, Reporting, • xConnect Collection, • xConnect Collection Search, • xDB Reference Data, • xDB Automation Operations, • xDB Automation Reporting, • Sitecore Cortex™ Processing Engine, • Sitecore Cortex™ Reporting service. Use this topology if you are planning a fully featured Sitecore Experience Platform installation.

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 about the Sitecore scaling scenarios.

Download the Web Deploy Packages and SIF Configuration Files

After you select your deployment topology, you must download the corresponding zip file from the Sitecore Downloads page. Each zip file contains the web deploy packages and resources for that topology.

XP Single (XP0)

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

- `sitecore 9.1.1 rev. 002459 (OnPrem)_single.scwdp.zip`
- `sitecore 9.1.1 rev. 002459 (OnPrem)_xp0xconnect.scwdp.zip`
- `Sitecore.IdentityServer.2.0.1-r00166.scwdp.zip`

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

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

To install all the XP Single packages on a single server, use the following files:

- `XP0-SingleDeveloper.json`

- `XP0-SingleDeveloper.ps1`

XM Scaled (XM1)

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

- `sitecore 9.1.1 rev. 002459 (OnPrem)_cd.scwdp.zip`
- `sitecore 9.1.1 rev. 002459 (OnPrem)_cm.scwdp.zip`
- `Sitecore.IdentityServer.2.0.1-r00166.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`
- `IdentityServer.json`
- `createcert.json`

To install all the XM Scaled packages on a single server, use the following files:

- `XM1-SingleDeveloper.json`
- `XM1-SingleDeveloper.ps1`

XP Scaled (XP1)

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

- `sitecore 9.1.1 rev. 002459 (OnPrem)_cd.scwdp.zip`
- `sitecore 9.1.1 rev. 002459 (OnPrem)_cm.scwdp.zip`
- `sitecore 9.1.1 rev. 002459 (OnPrem)_prc.scwdp.zip`
- `sitecore 9.1.1 rev. 002459 (OnPrem)_rep.scwdp.zip`
- `sitecore 9.1.1 rev. 002459 (OnPrem)_xp1collection.scwdp.zip`
- `sitecore 9.1.1 rev. 002459 (OnPrem)_xp1collectionsearch.scwdp.zip`
- `sitecore 9.1.1 rev. 002459 (OnPrem)_xp1marketingautomation.scwdp.zip`
- `sitecore 9.1.1 rev. 002459 (OnPrem)_xp1marketingautomationreporting.scwdp.zip`
- `sitecore 9.1.1 rev. 002459 (OnPrem)_xp1referencedata.scwdp.zip`
- `sitecore 9.1.1 rev. 002459 (OnPrem)_dds.scwdp.zip`
- `sitecore 9.1.1 rev. 002459 (OnPrem)_xp1cortexprocessing.scwdp.zip`
- `sitecore 9.1.1 rev. 002459 (OnPrem)_xp1cortexreporting.scwdp.zip`
- `Sitecore.IdentityServer.2.0.1-r00166.scwdp.zip`

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

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

- `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-xp1-collection.json`
- `xconnect-xp1-collectionsearch.json`
- `xconnect-xp1-marketingautomation.json`
- `xconnect-xp1-marketingautomationreporting.json`
- `xconnect-xp1-referencedata.json`
- `xconnect-xp1-CortexProcessing.json`
- `xconnect-xp1-CortexReporting.json`

To install all the XP Scaled packages on a single server, use the following files:

- `XP1-SingleDeveloper.json`
- `XP1-SingleDeveloper.ps1`

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:

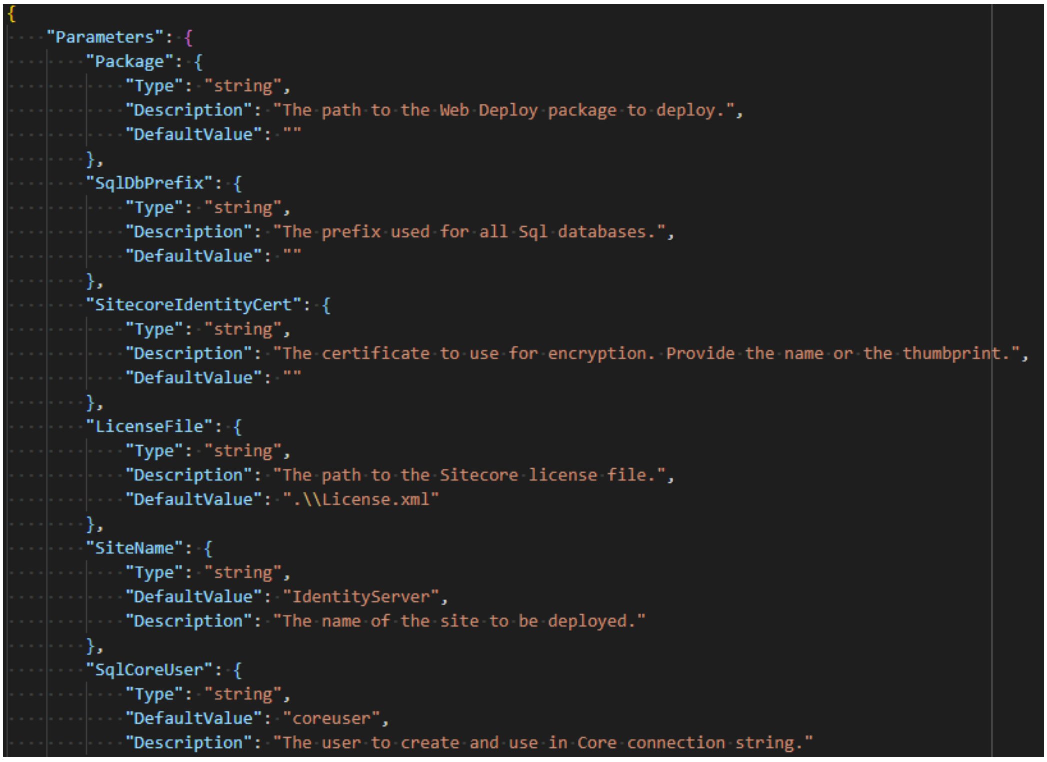
- Creating an IIS Application Pool
- Creating an IIS website
- Installing Web Deploy Packages
- Configuring File Permissions

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

The configuration files contain a description for each parameter. The following screenshot shows a snippet of a `Parameters` block from a configuration file.



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

If you have installed Machine Learning Server, you must:

1. Create a folder to store the certificates for Machine Learning Server, for example, `c:\temp\certificates`.
2. Open the Control Panel and search for *Manage computer certificates*.
3. To export the xConnect client certificate for the hosting *Sitecore Cortex™ Processing Blob Storage* web endpoint, in the **certlm** dialog box, expand *Personal*, *Certificates.*
4. Right-click the certificate for your installation, for example, *sc91.xconnect_client* and then click **All Tasks**, **Export…**
5. The **Certificate Export Wizard** opens and guides you through the process.
6. In the wizard, select the following options:
    - *Yes, export the private key*
    - *Export all extended properties*
    - Password-protect the file – for example, *secret*.
7. Save the file with any name in the folder that you created in step 1 – for example `c:\temp\certificates\client.pfx`.
8. If the *Sitecore Cortex™ Processing Table Storage* web endpoint is located on a different instance, you must perform steps 1-6 to export the certificate with the private key for the xConnect instance that hosts the *Sitecore Cortex™ Processing Table Storage* web endpoint.
9. To export the certificate for the certification authority that issued the certificates for the *Sitecore Cortex™ Processing Blob Storage* web and *Sitecore Cortex™ Processing Table Storage* web endpoints, in the **certlm** dialog box, expand *Trusted Root Certification Authorities*, *Certificates*.
10. Right-click the certificate for your installation, for example: *DO_NOT_TRUST_SitecoreRootCert* and then click **All Tasks**, **Export…**
11. The **Certificate Export Wizard** opens and guides you through the process.

12. In the wizard, select the following options:
    - *No, do not export the private key*
    - *Base-64 encoded X.509 (.cer)*
13. Save the file with any name in the folder that you created in step 1 – for example `c:\temp\certificates\auth.cer`.
14. Provide the following parameters that SIF needs to install xConnect instance (for XP0) and *Sitecore Cortex™ Processing Engine* instance (for XP1):
    - `MachineLearningServerBlobEndpointCertificatePath` – the path to the exported certificate.
    - `MachineLearningServerBlobEndpointCertificatePassword` – the password to the certificate.
    - `MachineLearningServerTableEndpointCertificatePath` – the path to the exported certificate.
    - `MachineLearningServerTableEndpointCertificatePassword` – the password to the certificate.
    - `MachineLearningServerEndpointCertificationAuthorityCertificatePath` – the path to certificate for the certification authority.

If the certification authorities for the *Blob Storage and Table Storage* web endpoints are different, after you have installed an xConnect instance for XP Single or Processing Engine instance for XP1, you must:

1. Export another certificate for the certification authority. One that was not exported earlier.
2. In the `ConnectionStrings.config` file, add a new connection string.

   For example:

   In the `C:\path\to\xconnect\App_data\jobs\continuous\ProcessingEngine\App_Config\ConnectionStrings.config` file, add a connection string with the path to the certificate for the certification authority that you just exported:

   ```
   <add name="processing.webapi.table.certificate.authority"
   connectionString="C:\temp\certificates\tablerootcert.cer" />
   ```
3. In the `sc.Processing.Engine.Scripting.Mrs.xml` file, change the value of either the `BlobCaConnectionStringName` setting or the `TableDataCaConnectionStringName` setting to the connection string name that you just added.

   This value depends on which certificate you used as a parameter for the SIF deployment value.

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 XP 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.

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

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:

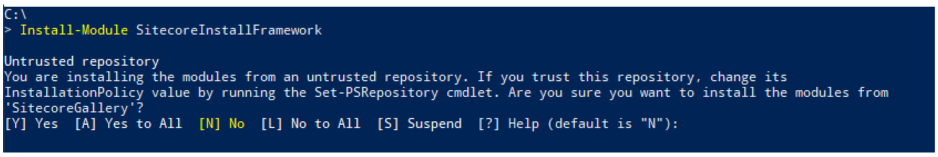
1. In Windows, launch PowerShell as an administrator.
2. To register the repository, in a PowerShell command line, run the following cmdlet:

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

3. To install the PowerShell module, run the following cmdlet:

```
Install-Module SitecoreInstallFramework
```

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



```
C:\
> Install-Module SitecoreInstallFramework

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"):
```

Update the Sitecore Installation Framework Module

When a newer version of the SIF module is available, you can update to the latest version by running a PowerShell cmdlet.

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

```
Update-Module SitecoreInstallFramework
```

3.2.2 Manual Installation

SIF also available as a `.zip` package.

You can download the Sitecore Install 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 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 *(global path)*	`C:\Program Files\WindowsPowerShell\Modules`
Specific user	`C:\Users\<user>\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 SIF `.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 the 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
```

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.

3.3 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:

```
$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. To start the installation, run the following cmdlet:

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

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

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

Note
For more details about the `Invoke-Command` cmdlet, see the PowerShell documentation.

3.4 Running Multiple Versions of SIF

If you want to install a 9.0.x version of Sitecore XP on the same computer as a 9.1.1 installation, you must also have SIF 1.2.x installed. PowerShell uses the latest available version of a module in a session by default.

To install a specific version of SIF, run the following command:

```
Install-Module -Name SitecoreInstallFramework -RequiredVersion x.x.x
```

Enter the appropriate value in the `RequiredVersion` parameter.

The following table lists the versions of SIF that are compatible with Sitecore XP 9.X:

Sitecore XP Version	SIF Version
9.0.x	1.2.1
9.1.0	2.0.0
9.1.1	2.1.0

3.4.1 Running a specific version of SIF

To run a specific version of SIF, launch a new Powershell session and run the following cmdlet:

```
Import-Module -Name SitecoreInstallFramework -Force -RequiredVersion x.x.x
```

You will use the specified version for the remainder of the session.

The next time you start a PowerShell session it automatically uses the latest available version.

Chapter 4

Local Environment Setup

This chapter describes how to install Sitecore XP 9.1.1 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
- Uninstall Sitecore XP

4.1 Install the Prerequisites

If you have manually installed all the prerequisites as described in chapters 2 and 3, skip this section.

To automatically install the prerequisites for any of the topologies on a single server:

1. Download the following file from the Sitecore Downloads Page – http://dev.sitecore.net
   - `Prerequisites.json`

   This file is stored in the `XP0 Configuration files 9.1.1 rev. 002459.zip` file.
2. To install the prerequisites, run the following cmdlet:

```
Install-SitecoreConfiguration -Path .\Prerequisites.json
```

When the installation is complete, you are informed if a server reboot is required.

For more information about the prerequisites and the `Prerequisites.json` file, see the section *The Prerequisites*.

4.1.1 Download the SIF Configuration Files

Download the relevant SIF configuration files and the Web Deploy Packages (WDPs) that contain each of the topologies from the Sitecore Downloads page – http://dev.sitecore.net.

For more information about the SIF configuration files, see the section *Choose Your On-premise Topology*.

4.2 Install Sitecore XP

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

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

The sample PowerShell scripts that Sitecore provides are designed to install the standard Sitecore XP topologies. These scripts demonstrate how to execute SIF commands with the appropriate configuration files and the required parameters.

To install one of the standard Sitecore XP topologies on a single workstation, you only need to execute one SIF command and set the path to the `XXX-SingleDeveloper.json` file for that topology and use the default parameters.

4.2.1 Sitecore XP Single Topology

The predefined XP Single topology configures:

- A Sitecore stand-alone website that handles content management, content delivery, reporting, and processing.
- The xConnect and xDB web services.
- The Sitecore Identity server that is a stand-alone website that acts as an OpenID Connect compliant security token service (STS).
- The search indexes on the Solr search engine.
- A Windows service that runs the Marketing Automation engine.
- A Windows service that runs the xConnect indexer.
- A Windows service that runs the Sitecore Cortex™ Processing Engine.
- The Sitecore Cortex™ Reporting service.
- 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.
- A self-signed server certificate for running HTTPS on the Sitecore Identity server.

The `XP0-SingleDeveloper.json` executes the following configuration files in the appropriate order:

- `createcert.json`

  Creates the self-signed certificate for the Sitecore Identity server. You must add the appropriate parameters.

- `IdentityServer.json`

  Sets-up the stand-alone .Net Core website for the Sitecore Identity server and contains the connection strings to the *Core* database that is installed by `sitecore-XP0.json`.

  This configuration uses the following WDP package:

  - `Sitecore.IdentityServer.2.0.1-r00166.scwdp.zip`

- `createcert.json`

  Creates the self-signed client certificate which is passed to the `xconnect-xp0.json` and `sitecore-XP0.json` SIF configuration files. You must add the appropriate parameters.

- `xconnect-solr.json`

  Creates the Solr indexes that are used by xConnect.

- `xconnect-xp0.json`

  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 9.1.1 rev. 002459 (OnPrem)_xp0xconnect.scwdp.zip`

- `sitecore-solr.json`

  Creates the Solr indexes that are used by Sitecore.

- `sitecore-XP0.json`

  Sets-up a stand-alone Sitecore website on IIS and the content databases on SQL Server.

  This configuration uses the following WDP package:

  - `Sitecore 9.1.1 rev. 002459 (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. The `XPO-SingleDeveloper.json` file contains the default parameters that are required to install a developer workstation.

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 `createcert.json` SIF configuration file.

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

4.2.2 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:

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

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

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

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

4.2.3 Sitecore Identity Server Configuration

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

The Sitecore Identity server configuration requires the following additional parameters:

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

  If a user forgets their password, they are redirected to the appropriate Sitecore server to fill in the form for password recovery.

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

For more information see Sitecore Identity documentation.

4.2.4 Edit, and Run the Installation Script for the XP Single Topology

To simplify your installation, you can use a PowerShell script to install the XP Single (XP0) 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:
   - `sitecore 9.1.1 rev. 002459 (OnPrem)_single.scwdp.zip`
   - `sitecore 9.1.1 rev. 002459 (OnPrem)_xp0xconnect.scwdp.zip`
   - `Sitecore.IdentityServer.2.0.1-r00166.scwdp.zip`
   - `IdentityServer.json`
   - `createcert.json`
   - `sitecore-solr.json`
   - `xconnect-solr.json`
   - `sitecore-XP0.json`
   - `xconnect-xp0.json`
   - `XP0-SingleDeveloper.json`
   - `XP0-SingleDeveloper.ps1`
3. Save your Sitecore license file in the `c:\resourcefiles` folder as `license.xml`.

4. In the `c:\resourcefiles` folder, edit the `XP0-SingleDeveloper.ps1` script and update each line with the settings that are appropriate for your environment.

   One of the settings that you must edit is `SitecoreAdminPassword`.

   If you do not specify the Sitecore administrator password in the script and leave the example value *"SIF-Default"* unchanged, a random password is generated for you. This password is written to the `XP0-SingleDeveloper.log` file – search for *Sitecore Admin Password* – and displayed when the `sitecore-xp0.json` file has been processed.

   Important
   The default Sitecore administrator password is not valid when you use this script to install Sitecore XP.

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

   ```
   .\XP0-SingleDeveloper.ps1
   ```

   All the passwords for DB users and other secrets such as the Identity Server client secret and the TelerikEncryptionKey are not displayed on the screen; they are automatically generated and inserted into the appropriate configuration files.

   If you add a DDS server, you must pass the generated passwords for the CM server to the DDS SIF installation parameters. The generated passwords can be found in the installation logs and in the `App_Config/connectionStrings.config` file on the CM server.

   If the `SqlCollectionUser` parameter is not set to a custom value during the installation, the prefix that is specified in the installation script is added to it.

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*.

We recommend that you keep these scripts. You can use them to uninstall this topology.

For more information about uninstalling Sitecore XP, see the section *Uninstall Sitecore XP*.

4.2.5 Edit, and Run the Installation Script for the XM Scaled Topology

To simplify your installation, you can use a PowerShell script to install the XM Scaled (XM1) 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:
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_cd.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_cm.scwdp.zip`
   - `Sitecore.IdentityServer.2.0.1-r00166.scwdp.zip`
   - `IdentityServer.json`
   - `createcert.json`
   - `sitecore-solr.json`
   - `sitecore-XM1-cd.json`
   - `sitecore-XM1-cm.json`
   - `XM1-SingleDeveloper.json`
   - `XM1-SingleDeveloper.ps1`
3. Save your Sitecore license file in the `c:\resourcefiles` folder as `license.xml`.

4. In the `c:\resourcefiles` folder, edit the `XM1-SingleDeveloper.ps1` file script and update each line with the appropriate settings for your environment.

   One of the settings that you must edit is `SitecoreAdminPassword`.

   If you do not specify the Sitecore administrator password in the script and leave the example value *"SIF-Default"* unchanged, a random password is generated for you. This password is written to the `XM1-SingleDeveloper.log` file – search for *Sitecore Admin Password* – and displayed when the `sitecore-XM1-cm.json` file has been processed.

   Important
   The default Sitecore administrator password is not valid when you use this script to install Sitecore XP.

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

   ```
   .\XM1-SingleDeveloper.ps1
   ```

   All the passwords for DB users and other secrets such as the Identity Server client secret and the TelerikEncryptionKey are not displayed on the screen; they are automatically generated and inserted into the appropriate configuration files.

   If the `SqlCollectionUser` parameter is not set to a custom value during the installation, the prefix that is specified in the installation script is added to it.

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*.

We recommend that you keep these scripts. You can use them to uninstall this topology.

For more information about uninstalling Sitecore XP, see the section *Uninstall Sitecore XP*.

4.2.6 Edit and Run the Installation Script for the XP Scaled Topology

To simplify your installation, you can use a PowerShell script to install the XP Scaled (XP1) 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:
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_cd.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_cm.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_prc.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_rep.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_xp1collection.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_xp1collectionsearch.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_xp1marketingautmation.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_xp1marketingautomationreporting.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_xp1cortexprocessing.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_xp1referencedata.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_xp1cortexreporting.scwdp.zip`
   - `Sitecore.IdentityServer 2.0.1 r00166 (OnPrem)_identityserver.scwdp.zip`

   - `IdentityServer.json`
   - `sitecore-solr.json`
   - `Sitecore-XP1-cd.json`
   - `Sitecore-XP1-cm.json`
   - `Sitecore-XP1-prc.json`
   - `Sitecore-XP1-rep.json`
   - `createcert.json`
   - `xconnect-solr.json`
   - `xconnect-xp1-collection.json`
   - `xconnect-xp1-collectionsearch.json`
   - `xconnect-xp1-MarketingAutomation.json`
   - `xconnect-xp1-MarketingAutomationReporting.json`
   - `xconnect-xp1-CortexProcessing.json`
   - `xconnect-xp1-ReferenceData.json`
   - `xconnect-xp1-CortexReporting.json`
   - `XP1-SingleDeveloper.json`
   - `XP1-SingleDeveloper.ps1`

3. Save your Sitecore license file in the `c:\resourcefiles` folder as `license.xml`.
4. In the `c:\resourcefiles` folder, edit the `XP1-SingleDeveloper.ps1` script and update each line with the appropriate settings for your environment.

   One of the settings that you must edit is `SitecoreAdminPassword`.

   If you do not specify the Sitecore administrator password in the script and leave the example value *"SIF-Default"* unchanged, a random password is generated for you. This password is written to the `XP1-SingleDeveloper.log` file – search for *Sitecore Admin Password* – and displayed when the `sitecore-XP1-cm.json` file has been processed.

   > **Important**
   > The default Sitecore administrator password is not valid when you use this script to install Sitecore XP.

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

   ```
   .\XP1-SingleDeveloper.ps1
   ```

   All the passwords for DB users and other secrets such as the Identity Server client secret and the TelerikEncryptionKey are not displayed on the screen; they are automatically generated and inserted into the appropriate configuration files.

   If the `SqlCollectionUser` parameter is not set to a custom value during the installation, the prefix that is specified in the installation script is added to it.

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*.

We recommend that you keep these scripts. You can use them to uninstall this topology.

For more information about uninstalling Sitecore XP, see the section *Uninstall Sitecore XP*.

4.3 Uninstall Sitecore XP

4.3.1 Uninstall a SIF Configuration File

You can use SIF to uninstall a SIF configuration file on a local server.

To uninstall a SIF configuration file on a local instance:

1. Launch PowerShell as an administrator.
2. Run the `Uninstall-SitecoreConfiguration` cmdlet, and specify the path to your SIF configuration file.

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

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

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

   For example:

   ```
   Uninstall-SitecoreConfiguration -Path <configurationpath>\sitecore-XP0.json -SqlDbPrefix SC.
   ```

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

   For example, running the `Verbose` cmdlet increases the amount of information that is logged, and the `-Skip <taskname>` cmdlet skips one or more tasks.

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

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

4.3.2 Uninstall the XP Single, XP Scaled, or XM Scaled Topologies

To uninstall any of the topologies, change the `Install-SitecoreConfiguration` cmdlet to `Uninstall-SitecoreConfiguration`.

For example:

Change

```
Install-SitecoreConfiguration @XP1Parameters *>&1 | Tee-Object XP1-SingleDeveloper.log
```

to

```
Uninstall-SitecoreConfiguration @XP1Parameters *>&1 | Tee-Object XP1-SingleDeveloper-
Uninstall.log
```

Run the script and the topology is removed.

Chapter 5

Production Environment Setup

This chapter describes how to install Sitecore XP 9.1.1 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:

- Enable Telemetry Reporting
- Secure the Sitecore installation
- Install a Scaled Sitecore XP
- Distributed Installation Scripts

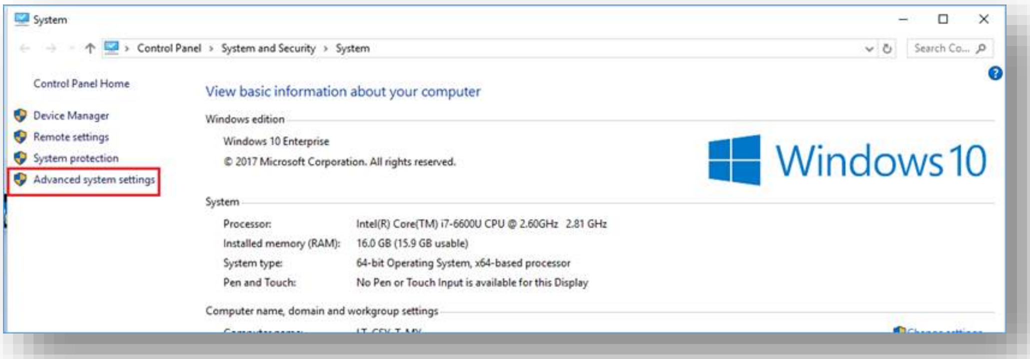
5.1 Enable Telemetry Reporting

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

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

To define a system variable:

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



View basic information about your computer

Windows edition

Windows 10 Enterprise

© 2017 Microsoft Corporation. All rights reserved.

System

Processor: Intel(R) Core(TM) i7-6600U CPU @ 2.60GHz 2.81 GHz

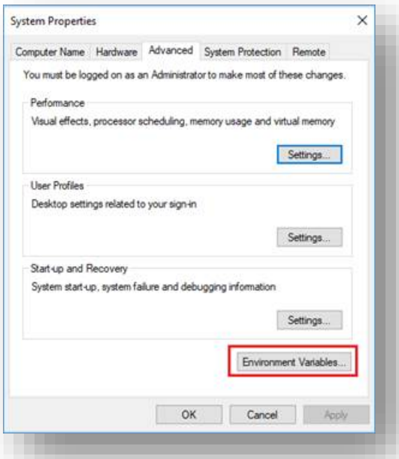
Installed memory (RAM): 16.0 GB (15.9 GB usable)

System type: 64-bit Operating System, x64-based processor

Pen and Touch: No Pen or Touch Input is available for this Display

Computer name, domain and workgroup settings

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



You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage and virtual memory

User Profiles

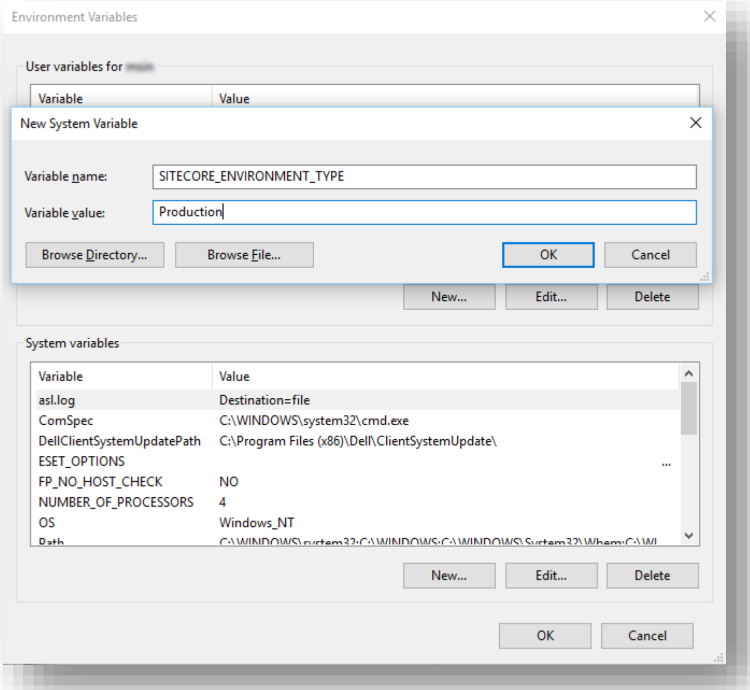
Desktop settings related to your sign-in

Start-up and Recovery

System start-up, system failure and debugging information

3. In the **Environment Variables** dialog, in the **System variables** section, click **New**.

4. In the **New System Variable** dialog, in the **Variable name** field, enter SITECORE_ENVIRONMENT_TYPE and in the **Variable value** field, enter *Production*.



Environment Variables

User variables for [redacted]

New System Variable

Variable name: SITECORE_ENVIRONMENT_TYPE

Variable value: Production

System variables

Variable	Value
asl.log	Destination=file
ComSpec	C:\WINDOWS\system32\cmd.exe
DellClientSystemUpdatePath	C:\Program Files (x86)\Dell\ClientSystemUpdate\
ESET_OPTIONS	...
FP_NO_HOST_CHECK	NO
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WI

5. Restart the computer.

5.2 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 the xConnect web services, the Sitecore Identity server, Microsoft Machine Learning Server, 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.

5.2.1 Set Up Server Certificate SSL Authentication on IIS

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

The following table lists the full set of server authentication certificates for each Sitecore XP topology:

XM Scaled (XM1)	XP Single (XP0)	XP Scaled (XP1)
Content Management Sitecore Identity server	xConnect Sitecore Identity server	Content Management Reporting Processing Sitecore Cortex™ Processing Engine Sitecore Cortex™ Reporting Service Sitecore Identity server 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 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	Sitecore Identity server Processing Reporting xConnect Collection xConnect Collection Search xDB Reference Data xDB Automation Operations xDB Automation Reporting Sitecore Cortex™ Reporting Service
Content Delivery	Content Management xConnect Collection xDB Reference Data xDB Automation Operations
Reporting	*None required*
Processing	xConnect Collection
xConnect Collection	*None required*
xConnect Collection Search	*None required*
xDB Reference Data	*None required*
xDB Automation Operations	xConnect Collection xDB Reference Data
xDB Automation Reporting	*None required*
EXM Dedicated Dispatch Server	Sitecore Identity server Processing Reporting xConnect Collection xConnect Collection Search xDB Reference Data xDB Automation Operations xDB Automation Reporting

Role Name	Server Certificate
Sitecore Cortex™ Reporting service	*None required*
Sitecore Cortex™ Processing Engine	xConnect Collection xConnect Collection Search

5.2.2 Set Up the Client Certificates

You must obtain and install the client certificates for a production environment before running SIF. If you do not provide the certificates, SIF creates self-signed certificates when they are required.

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.

The following table lists the full set of client authentication certificates for each Sitecore XP topology:

XM Scaled (XM1)	XP Single (XP0)	XP Scaled (XP1)
None	xConnect	xConnect Collection xConnect Collection Search xDB Reference Data xDB Automation Operations Sitecore Cortex™ Processing Engine Sitecore Cortex™ Reporting service

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*, *xDB Automation Reporting*, *Sitecore Cortex Processing Engine*, and *Sitecore Cortex Reporting Service* 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.reporting.client.certificate`

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

XP Scaled (XP1)

Role Name	Client Certificates	Associated connection strings containing client thumbprint

Content Management	xConnect Collection Search
	xDB Reference Data
	xDB Automation Operations
	xDB Automation Reporting
	Sitecore Cortex Reporting Service
		`xconnect.collection.certificate`
		`xdb.referencedata.client.certificate`
		`xdb.marketingautomation.operations.client.certificate`
		`xdb.marketingautomation.reporting.client.certificate`
		`sitecore.reporting.client.certificate`

Content Delivery	xConnect Collection
	xDB Reference Data
	xDB Automation Operations
		`xconnect.collection.certificate`
		`xdb.referencedata.client.certificate`
		`xdb.marketingautomation.operations.client.certificate`

Processing	xConnect Collection	`xconnect.collection.certificate`

Sitecore Cortex Processing Engine	xConnect Collection
	xConnect Search
		`xconnect.collection.certificate`
		`xconnect.configuration.certificate`
		`xconnect.search.certificate`

Sitecore Marketing Automation Engine	xConnect Collection	`xconnect.collection.certificate`
xConnect Collection*	*None required - because this role does not make calls to other roles.*	-

Role Name	Client Certificates	Associated connection strings containing client thumbprint
xConnect Collection Search*	*None required - because this role does not make calls to other roles.*	-
xDB Reference Data*	*None required - because this role does not make calls to other roles.*	-
xDB Automation Operations	xConnect Collection	`xconnect.collection.certificate`
xDB Automation Reporting*	*None required - because this role does not make calls to other roles.*	-

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

By default, these users are:

- The *ApplicationPoolIdentity* for the web sites.
- The *Local Service* account for Windows services.

5.2.3 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)
None required	xConnect Sitecore	Content Management xConnect Collection Search

5.3 Install a Scaled Sitecore XP

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

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 XP:

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\IdentityServer\IdentityServer.json"

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"

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

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

For information about setting up EXM, see the EXM documentation.

5.3.1 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 the deployment:

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

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

Specify the Names or Thumbprints of the Certificates

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

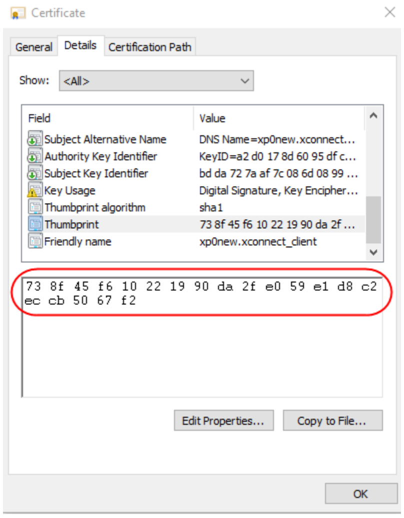
- For the *client* authentication certificate:

```
-XConnectCert "xConnect_client"
```

or

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

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



Show: <All>

Field	Value
Subject Alternative Name	DNS Name=xp0new.xconnect...
Authority Key Identifier	KeyID=a2 d0 17 8d 60 95 df c...
Subject Key Identifier	bd da 72 7a af 7c 08 6d 08 99 ...
Key Usage	Digital Signature, Key Encipher...
Thumbprint algorithm	sha1
Thumbprint	73 8f 45 f6 10 22 19 90 da 2f ...
Friendly name	xp0new.xconnect_client

```
73 8f 45 f6 10 22 19 90 da 2f e0 59 e1 d8 c2
ec cb 50 67 f2
```

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

```
-SSLCert "CM_test"
```

or

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

5.4 Distributed Installation Scripts

To simplify your scaled installation on multiple servers, you can use a PowerShell script to install the XM or XP Scaled topologies.

5.4.1 Distributed Installation Script Prerequisites

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

1. Enable PowerShell Remoting.

   For more information about PowerShell Remoting, see the section *Running SIF Remotely*.

2. Install the Prerequisites.

   For more information about the prerequisites, see the section *Install the Prerequisites*.

3. Create the `c:\resourcefiles` folder on every machine that will be assigned a Sitecore server role or will run Solr.

5.4.2 Run the Distributed Installation Script for the XM Scaled Topology

To edit and run the XM topology installation script:

1. Create a folder called `c:\resourcefiles`.
2. Download and save the following WDP packages and SIF resource files in this folder:
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_cd.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_cm.scwdp.zip`
   - `Sitecore.IdentityServer.2.0.1-r00166.scwdp.zip`
   - `IdentityServer.json`
   - `createcert.json`
   - `importcert.json`
   - `sitecore-solr.json`
   - `sitecore-XM1-cd.json`
   - `sitecore-XM1-cm.json`
   - `Role-Remote.json`
   - `XM1-Distributed.json`
   - `XM1-Distributed.ps1`

   Note
   There is an example of the `XM1-Distributed.ps1` file in the section *Install and configure Microsoft* Machine Learning Server

Sitecore 9.1.1 introduces the new Sitecore Cortex™ Processing Engine. You can use the Processing Engine with or without Microsoft Machine Learning Server.

Note
There are no workers in the Processing Engine that use Machine Learning Server integration by default.

To use Machine Learning Server integration:

1. Install Machine Learning Server
2. Navigate to `C:\Program Files\Microsoft\ML Server\R_SERVER\bin\x64\Rgui.exe` and run RGui as an administrator.

3. In RGui, run the following commands:

```
```
lib <- tail(.libPaths(), n=1)
repo <- 'https://cran.microsoft.com/snapshot/2017-07-04';
install.packages('openssl', lib, repos=repo)
install.packages('curl', lib, repos=repo)
install.packages('httr', lib, repos=repo)
```
```

4. To configure Machine Learning Server to operationalize analytics on a single machine, set up the web and compute nodes.

   For more information about configuring MLS, see Microsoft's documentation.

5. Configure the HTTPS protocol for the Web node.

   For more information about how to configure HTTPS, see Microsoft's documentation.

6. Make a note of the web node (port 12800 by default), username, and password.

   For example: `https://admin:Secret123!@localhost:12800/`

   If you want to use special characters in the *processing.engine.mrs* connection string, the characters must be encoded. For example, *Secret123#* becomes *Secret123%23*:

```
<add name="processing.engine.mrs"
connectionString="http://admin:Secret123%23@localhost:12800/"/>
```

   Distributed Installation Script Examples.

7. Save your Sitecore license file in the `c:\resourcefiles` folder as `license.xml`.

8. In the `c:\resourcefiles` folder, edit the `XM1-Distributed.ps1` file script and update each line with the appropriate settings for your environment.

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

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

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

5.4.3 Run the Distributed Installation Script for the XP Scaled Topology

To edit and run the Distributed XP topology installation script:

1. Create a folder called `c:\resourcefiles`.

2. Download and save the following WDP packages and SIF resource files in this folder:

   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_cd.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_cm.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_prc.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_rep.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_xp1collection.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_xp1collectionsearch.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_xp1marketingautomation.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_xp1marketingautomationreporting.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_xp1cortexprocessing.scwdp.zip`
   - `Sitecore 9.1.1 rev. 002459 (OnPrem)_xp1referencedata.scwdp.zip`

- `Sitecore 9.1.1 rev. 002459 (OnPrem)_xp1cortexreporting.scwdp.zip`
- `Sitecore.IdentityServer 2.0.1 r00166(OnPrem)_identityserver.scwdp.zip`
- `IdentityServer.json`
- `Sitecore-solr.json`
- `Sitecore-XP1-cd.json`
- `Sitecore-XP1-cm.json`
- `Sitecore-XP1-prc.json`
- `Sitecore-XP1-rep.json`
- `createcert.json`
- `importcert.json`
- `xconnect-solr.json`
- `xconnect-xp1-collection.json`
- `xconnect-xp1-collectionsearch.json`
- `xconnect-xp1-MarketingAutomation.json`
- `xconnect-xp1-MarketingAutomationReporting.json`
- `xconnect-xp1-CortexProcessing.json`
- `xconnect-xp1-ReferenceData.json`
- `xconnect-xp1-CortexReporting.json`
- `Role-Remote.json`
- `XP1-Distributed.json`
- `XP1-Distributed.ps1`

Note
There is an example of the `XP1-Distributed.ps1` file in the section *Install and configure Microsoft* Machine Learning Server

Sitecore 9.1.1 introduces the new Sitecore Cortex™ Processing Engine. You can use the Processing Engine with or without Microsoft Machine Learning Server.

Note
There are no workers in the Processing Engine that use Machine Learning Server integration by default.

To use Machine Learning Server integration:

3. Install Machine Learning Server
4. Navigate to `C:\Program Files\Microsoft\ML Server\R_SERVER\bin\x64\Rgui.exe` and run RGui as an administrator.
5. In RGui, run the following commands:

~~~
```
lib <- tail(.libPaths(), n=1)
repo <- 'https://cran.microsoft.com/snapshot/2017-07-04';
install.packages('openssl', lib, repos=repo)
install.packages('curl', lib, repos=repo)
install.packages('httr', lib, repos=repo)
```
~~~

6. To configure Machine Learning Server to operationalize analytics on a single machine, set up the web and compute nodes.

   For more information about configuring MLS, see Microsoft's documentation.

7. Configure the HTTPS protocol for the Web node.

   For more information about how to configure HTTPS, see Microsoft's documentation.

8. Make a note of the web node (port 12800 by default), username, and password.

   For example: `https://admin:Secret123!@localhost:12800/`

   If you want to use special characters in the *processing.engine.mrs* connection string, the characters must be encoded. For example, *Secret123#* becomes *Secret123%23*:

   ```
   <add name="processing.engine.mrs"
   connectionString="http://admin:Secret123%23@localhost:12800/"/>
   ```
   Distributed Installation Script Examples.

9. Save your Sitecore license file in the `c:\resourcefiles` folder as `license.xml`.

10. In the `c:\resourcefiles` folder, edit the `XP1-Distributed.ps1` script and update each line with the appropriate settings for your environment.

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

    ```
    .\XP1-Distributed.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*.

5.4.4 Uninstalling a Distributed Deployment

Centralized uninstallation of distributed deployments is not supported yet. If you need to re-install a scaled deployment you *must* remove everything that was installed during previous installation from each individual computer. For more information about uninstalling Sitecore XP, see the section *Uninstall Sitecore XP*.

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 9.1.1.

This chapter contains the following sections:

- Configure MongoDB Provider for xConnect
- Configure High Availability 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.

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 Configure the xConnect MongoDB Data Provider

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

To enable the MongoDB provider, you must modify the configuration files for all the server roles that you use in your topology:

1. Enable the `sc.Xdb.Collection.Data.MongoDb.xml.disabled` configuration file by removing `.disabled` extension.
2. Disable `sc.Xdb.Collection.Data.Sql.xml` configuration file by adding the `.disabled` file extension.
3. Update the *collection* connection string to point to the MongoDB instance.

   You must also update the *collection* connection string in the indexer job that exists under the following roles:

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

   An example of a connection string for a configured replica set with least privilege users:

   ```
   mongodb://sa:12345@10.45.111.102:57017,10.45.111.102:57018,10.45.111.102:57019/collect
   ion?replicaSet=testReplicaSet&amp;retryWrites=true
   ```

   An example of a connection string for a configured sharded cluster with least privilege users:

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

4. Rebuild the xDB search indexes in Solr.

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

6.1.2 MongoDB 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 recommended 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 retry Writes 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 MongoDB 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 a 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.

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 Configure High Availability for xConnect

Important
If you are deploying a PaaS solution, skip this section.

If you are not going to configure a High Availability feature, skip this section.

High Availability is an xConnect feature that is based on Always On availability groups and configurable retryers.

6.2.1 Configure Always On Availability Groups

A Windows Server Failover Cluster (WSFC) is required when you deploy Always On availability groups.

You must also configure Synchronous-commit mode with Automatic Failover.

For more information about how to configure availability groups, see Microsoft's documentation.

To provide client connectivity to the database for a given availability group, you must create an availability group listener.

The availability group listener allows a client to connect to an availability replica without knowing the name of the physical instance of SQL Server that it is connecting to. You *must* specify the DNS name of the listener in the connection string instead of the server name. This helps your solution to automatically switch to the primary replica during a failover.

Here is an example of a connection string for an availability group listener:

```
Data Source=tcp:SMMListener,5025;Initial Catalog=ShardMapManagerDb;User Id=sa;Password=12345;
```

6.2.2 Configure the Collection Database

Important
If you are going to use MongoDB for the *Collection* database, skip this section.

The *Collection* database is designed to manage high read/write activity, and therefore supports sharding. The catalog of all the shards is located in the *Shard Map Manager* database. You must manually update the *[ShardsGlobal]* table in the catalog so that it can use listeners instead of server names.

6.3 Rebuild the Search Indexes and the Link Databases

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

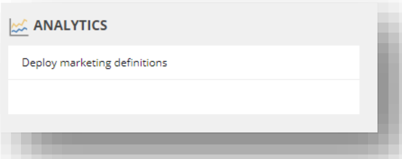
6.4 Deploy Marketing Definitions

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

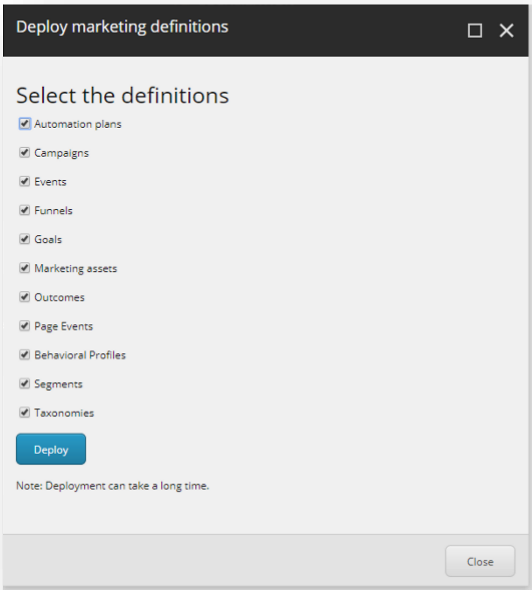
Note
If you do not plan to use the Sitecore Experience Database (xDB), you do not have to perform these steps.

To deploy the 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**.


Deploy marketing definitions

Select the definitions

☑ Automation plans
☑ Campaigns
☑ Events
☑ Funnels
☑ Goals
☑ Marketing assets
☑ Outcomes
☑ Page Events
☑ Behavioral Profiles
☑ Segments
☑ Taxonomies

Deploy

Note: Deployment can take a long time.

Close

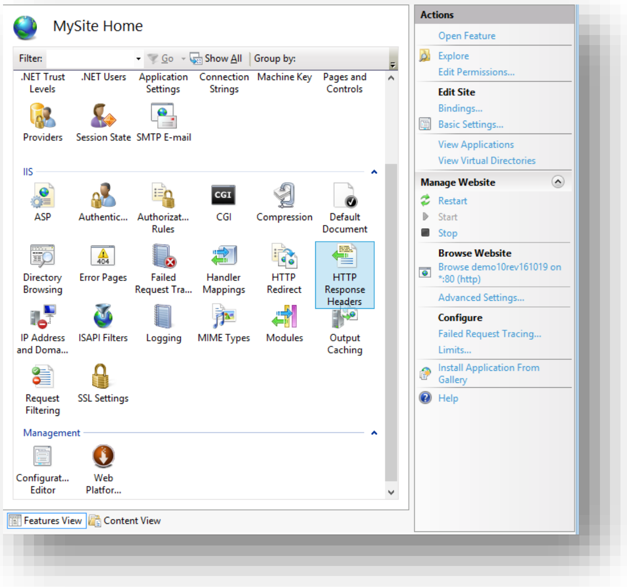
6.5 Content Expiration

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.

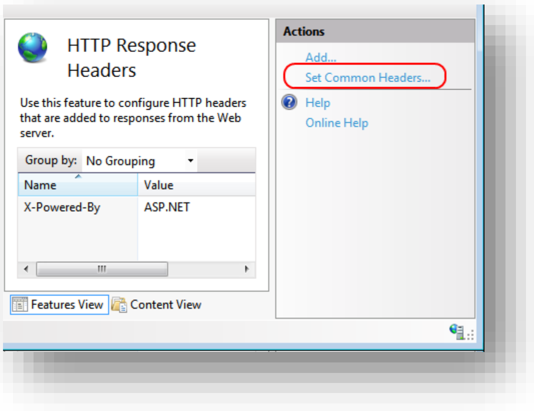
By setting Expire Web content to something other than *immediately*, you can reduce second-access load times by 50 –70%. This setting does not affect dynamically-generated content.

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…**

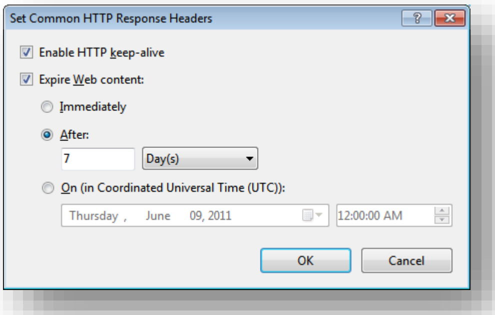


Use this feature to configure HTTP headers that are added to responses from the Web server.

Name	Value
X-Powered-By	ASP.NET

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.



Set Common HTTP Response Headers

☑ Enable HTTP keep-alive

☑ Expire Web content:

○ Immediately

● After:

7 Day(s)

○ On (in Coordinated Universal Time (UTC)):

Thursday , June 09, 2011 12:00:00 AM

7. Click **OK**.

6.6 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 allowed 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:

   - 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:

   - 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.

6.7 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
```

6.8 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.

6.9 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.

6.10 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

Appendix

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

This chapter contains the following sections:

- Common Issues
- Access Rights
- Certificates
- Install and configure Microsoft Machine Learning Server
- Distributed Installation Script Examples

7.1 Install and configure Microsoft Machine Learning Server

Sitecore 9.1.1 introduces the new Sitecore Cortex™ Processing Engine. You can use the Processing Engine with or without Microsoft Machine Learning Server.

Note
There are no workers in the Processing Engine that use Machine Learning Server integration by default.

To use Machine Learning Server integration:

4. Install Machine Learning Server
5. Navigate to `C:\Program Files\Microsoft\ML Server\R_SERVER\bin\x64\Rgui.exe` and run RGui as an administrator.
6. In RGui, run the following commands:

~~~~
```
lib <- tail(.libPaths(), n=1)
repo <- 'https://cran.microsoft.com/snapshot/2017-07-04';
install.packages('openssl', lib, repos=repo)
install.packages('curl', lib, repos=repo)
install.packages('httr', lib, repos=repo)
```
~~~~

7. To configure Machine Learning Server to operationalize analytics on a single machine, set up the web and compute nodes.

   For more information about configuring MLS, see Microsoft's documentation.
8. Configure the HTTPS protocol for the Web node.

   For more information about how to configure HTTPS, see Microsoft’s documentation.
9. Make a note of the web node (port 12800 by default), username, and password.

   For example: `https://admin:Secret123!@localhost:12800/`

   If you want to use special characters in the *processing.engine.mrs* connection string, the characters must be encoded. For example, *Secret123#* becomes *Secret123%23*:

```
<add name="processing.engine.mrs"
connectionString="http://admin:Secret123%23@localhost:12800/"/>
```

- Distributed Installation Script Examples

7.2 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.
- Ensure your thumbprint is in uppercase letters.

  For example: `3D703B5198D6D3CEE1D0C1B1BC9ECB6D34989BA4`.

  You can find the thumbprint in the following locations:

  - `Sitecore\App_Config\ConnectionStrings.config`
  - `XConnect\App_Config\ConnectionStrings.config`
  - `XConnect\App_Data\jobs\continuous\AutomationEngine\App_Config\ConnectionStrings.config`
- Ensure the self-signed certificate you created in IIS has the same name as your xConnect instance.
- Ensure the client authentication certificate (under the local machine's *Personal* store) has *read* permissions for the *IIS_IUSR* group and the *NETWORK SERVICE* group.

My Solr index is empty

- 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 9.1.1, it is possible that your xConnect installation may fail. This occurs because Skype and Sitecore xConnect both use port 443, which interferes with the installation.

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

Failed Installations

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

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

For more information about uninstalling, see the section *Uninstall a SIF Configuration File*.

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

To remove an incorrect certificate:

1. To open the **Certificate Management** console, in the Windows command prompt, enter `certlm.msc` and press **Enter**.

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

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

My Solr Schema is not Populated

To populate the Solr schema:

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

The Indexing Manager shows no results after rebuilding the indexes

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

To solve this problem, populate the Solr schema again.

7.3 Access Rights

7.3.1 Use Windows Authentication with SQL Server

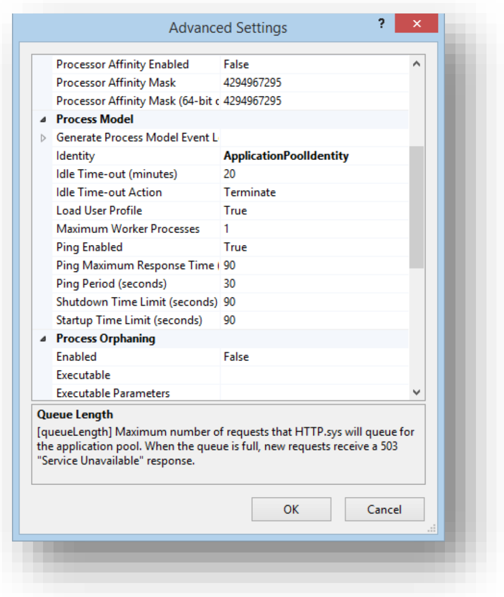
You can configure Sitecore to use Windows Authentication for a SQL connection and remove the user name and password from the `connectionStrings.config` file.

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.



Processor Affinity Enabled	False
Processor Affinity Mask	4294967295
Processor Affinity Mask (64-bit c	4294967295
◢ **Process Model**	
▷ Generate Process Model Event L	
Identity	**ApplicationPoolIdentity**
Idle Time-out (minutes)	20
Idle Time-out Action	Terminate
Load User Profile	True
Maximum Worker Processes	1
Ping Enabled	True
Ping Maximum Response Time (	90
Ping Period (seconds)	30
Shutdown Time Limit (seconds)	90
Startup Time Limit (seconds)	90
◢ **Process Orphaning**	
Enabled	False
Executable	
Executable Parameters	

Queue Length
[queueLength] Maximum number of requests that HTTP.sys will queue for the application pool. When the queue is full, new requests receive a 503 "Service Unavailable" response.

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.

7.3.2 Use 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*.

7.4 Certificates

Client certificates

When you have installed Sitecore XP 9.1.1, 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:

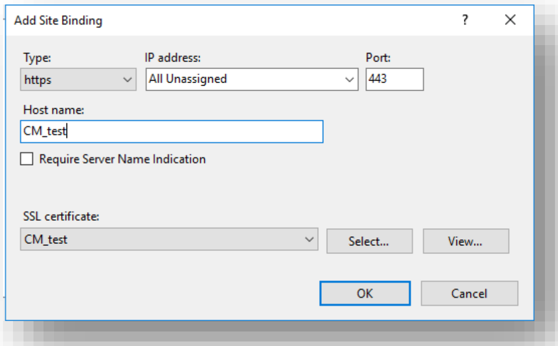
```
<add name="xconnect.collection.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;Fi
ndValue=738F45F610221990DA2FE059E1D8C2ECCB5067F2" />
```

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

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

Server certificates

When you have installed Sitecore XP 9.1.1, you have an IIS site named `CM_test` with HTTPS binding and an associated SSL certificate called `CM_test`.



Add Site Binding

Type: https

IP address: All Unassigned

Port: 443

Host name: CM_test

☐ Require Server Name Indication

SSL certificate: CM_test

7.4.1 Configure a New Client Certificate

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

To configure Sitecore to use a new client certificate:

1. Install the new client certificate on every computer on which you have installed the xConnect client and ensure that the authority that issued the certificate is in the *Trusted Authorities* list.

   For more information about the appropriate role and the certificate that you must install, see the section *Set Up the 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**.

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 the Client* Certificates.

11. Update the thumbprint values in all of the certificate connection strings on the following instances:

XP Single (XP0)	XP Scaled (XP1)
Sitecore	Content Delivery
xConnect	Content Management
	Processing
	Marketing Automation Engine
	xDB Automation Operations

    For example, on each XP Single (XP0) Sitecore instance, update the thumbprint value in these connection strings:

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

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

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

<add name="sitecore.reporting.client.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;Fi
ndValue=859E88DC0692BA1583145223C455F186937C0D62" />

<add name="xdb.marketingautomation.operations.client.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;Fi
ndValue=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 (XP0)	XP Scaled (XP1)
xConnect	xConnect Collection
	XConnect Search
	xDB Reference Data
	xDB Automation Reporting
	xDB Automation Operations
	Sitecore Cortex Processing Engine
	Sitecore Cortex Reporting Service

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

7.4.2 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.

7.5 Install and configure Microsoft Machine Learning Server

Sitecore 9.1.1 introduces the new Sitecore Cortex™ Processing Engine. You can use the Processing Engine with or without Microsoft Machine Learning Server.

Note
There are no workers in the Processing Engine that use Machine Learning Server integration by default.

To use Machine Learning Server integration:

4. Install Machine Learning Server
5. Navigate to `C:\Program Files\Microsoft\ML Server\R_SERVER\bin\x64\Rgui.exe` and run RGui as an administrator.
6. In RGui, run the following commands:

~~~
```
lib <- tail(.libPaths(), n=1)
repo <- 'https://cran.microsoft.com/snapshot/2017-07-04';
install.packages('openssl', lib, repos=repo)
install.packages('curl', lib, repos=repo)
install.packages('httr', lib, repos=repo)
```
~~~

7. To configure Machine Learning Server to operationalize analytics on a single machine, set up the web and compute nodes.

   For more information about configuring MLS, see Microsoft's documentation.
8. Configure the HTTPS protocol for the Web node.

   For more information about how to configure HTTPS, see Microsoft's documentation.
9. Make a note of the web node (port 12800 by default), username, and password.

   For example: `https://admin:Secret123!@localhost:12800/`

   If you want to use special characters in the *processing.engine.mrs* connection string, the characters must be encoded. For example, *Secret123#* becomes *Secret123%23*:

```
<add name="processing.engine.mrs"
connectionString="http://admin:Secret123%23@localhost:12800/"/>
```

7.6 Distributed Installation Script Examples

7.6.1 Distributed XM Scaled Topology

You can use the following script to deploy the Distributed XM Scaled topology:

```
# The Prefix that will be used on SOLR, Website and Database instances.
$Prefix = "sc910xm1"
# The root folder with the license file and WDP files.
$SCInstallRoot = "C:\Temp\XM1"
# The name for the Sitecore Content Management server.
$SitecoreContentManagementSitename = "$Prefix.cm"
# The name for the Sitecore Content Delivery Server.
$SitecoreContentDeliverySitename = "$Prefix.cd"
# Identity Server site name
$IdentityServerSiteName = "$prefix.identityserver"
# The Path to the license file
$LicenseFile = "$SCInstallRoot\license.xml"
# The URL of the Solr Server
$SolrUrl = "https://solr_host:8983/solr"
# The Folder that Solr has been installed in.
$SolrRoot = "C:\Solr-7.2.1"
# The Name of the Solr Service.
$SolrService = "Solr-7.2.1"
# The DNS name or IP of the SQL Instance.
$SqlServer = ""
# A SQL user with sysadmin privileges.
$SqlAdminUser = "sa"
# The password for $SQLAdminUser.
$SqlAdminPassword = "Password12345"
# The path to the Sitecore Content Management Package to Deploy
$SiteCoreContentManagementPackage = (Get-ChildItem "$SCInstallRoot\Sitecore XM 9.1.1 rev.
002459 (OnPrem)_cm.scwdp.zip").FullName
# The path to the Sitecore Content Delivery Package to Deploy
$SitecoreContentDeliveryPackage = (Get-ChildItem "$SCInstallRoot\Sitecore XM 9.1.1 rev.
002459 (OnPrem)_cd.scwdp.zip").FullName
# The path to the Identity Server Package to Deploy.
$IdentityServerPackage = (Get-ChildItem "$SCInstallRoot\Sitecore.IdentityServer 2.0.1 rev.
00166 (OnPrem)_identityserver.scwdp.zip").FullName
# The Identity Server password recovery URL, this should be the URL of the CM Instance
$PasswordRecoveryUrl = "http://$CMComputerName"
# The URL of the Identity Authority
$SitecoreIdentityAuthority = "https://$IdentityServerComputerName"
#The random string key used for establishing connection with IdentityService. This will be
regenerated if left on the default.
$ClientSecret = "SIF-Default"
#adding allowed CORS Origins, urls of all clients should be added here
$allowedCorsOrigins = "https://$CMComputerName"
# The username for computers to deploy Sitecore roles to.
```

```
$ComputerUserName = ""
# The password for computers to deploy Sitecore roles to.
$ComputerPassword = ""
# The DNS name or IP of the Solr computer.
$SolrComputerName = ""
# The DNS name of the Identity Server computer.
$IdentityServerComputerName = ""
# The DNS name of the Content Management computer.
$CMComputerName = ""
# The DNS name of the Content Delivery computer.
$CDComputerName = ""

# Install XM1 via combined partials file.
$Parameters = @{
    Path = "$SCInstallRoot\XM1-Distributed.json"
    ComputerUserName = $ComputerUserName
    ComputerPassword = $ComputerPassword
    SitecoreSolrComputerName = $SolrComputerName
    IdentityServerComputerName = $IdentityServerComputerName
    SitecoreCMComputerName = $CMComputerName
    SitecoreCDComputerName = $CDComputerName
    SqlServer = $SqlServer
    SqlAdminUser = $SqlAdminUser
    SqlAdminPassword = $SqlAdminPassword
    SolrUrl = $SolrUrl
    SolrRoot = $SolrRoot
    SolrService = $SolrService
    Prefix = $Prefix
    IdentityServerSiteName = $IdentityServerSiteName
    LicenseFile = $LicenseFile
    SiteCoreContentManagementPackage = $SiteCoreContentManagementPackage
    SitecoreContentDeliveryPackage = $SitecoreContentDeliveryPackage
    IdentityServerPackage = $IdentityServerPackage
    SitecoreContentManagementSitename = $SitecoreContentManagementSitename
    SitecoreContentDeliverySitename = $SitecoreContentDeliverySitename
    PasswordRecoveryUrl = $PasswordRecoveryUrl
    SitecoreIdentityAuthority = $SitecoreIdentityAuthority
    ClientSecret = $ClientSecret
    AllowedCorsOrigins = $AllowedCorsOrigins
}

Push-Location $SCInstallRoot

Install-SitecoreConfiguration @Parameters *>&1 | Tee-Object XM1-Distributed.log

Pop-Location
```

7.6.2 Distributed XP Scaled Topology

You can use the following script to deploy the distributed XP scaled topology:

```
# General Settings
# Prefix for SOLR collection and SQL database instances
$prefix = "sc910xp1"
# Folder with the install files in it.
$SCInstallRoot = "C:\Temp\XP1"
# The location of the license file.
$LicenseFile = "$SCInstallRoot\license.xml"
# The URL of the SOLR Service, must be HTTPS
$SolrUrl = "https://solr_host:8983/solr"
# The Folder that Solr has been installed in.
$SolrRoot = "C:\Solr-7.2.1"
# The Name of the Solr Service.
$SolrService = "Solr-7.2.1"
# The DNS name or IP of the SQL Instance.
$SqlServer = ""
# A SQL user with sysadmin privileges.
$SqlAdminUser = "sa"
# The password for $SqlAdminUser.
$SqlAdminPassword = "Password12345"
# The name for the Content Delivery site.
$ContentDeliverySiteName = "$prefix.cd"
# The name for the Content Management site.
$ContentManagementSiteName = "$prefix.cm"
# The name for the Reporting site.
$ReportingSiteName = "$prefix.rep"
# The name for the Processing site.
$ProcessingSiteName = "$prefix.prc"
# The name for the Reference Data site.
$ReferenceDateSiteName = "$prefix.refdata"
# Identity Server site name.
$IdentityServerSiteName = "$prefix.identityserver"
# The name for the Marketing Automation site.
$MarketingAutomationSiteName = "$prefix.ma"
# The name for the Marketing Automation Reporting site.
$MarketingAutomationReportingSiteName = "$prefix.mareporting"
# The name for the XConnect site url.
$ClientCertificateName = "$prefix.xconnect_client"
# The name for the XConnect collection site.
$CollectionSitename = "$prefix.collection"
# The name for the search site.
$CollectionSearchSitename = "$prefix.search"
# The name fore the XConnect processing engine service.
$CortexProcessingSitename = "$prefix.processingEngine"
# The name for the XConnect reporting servcice.
$CortexReportingSitename = "$prefix.reporting"
```

```
# The random string key used for establishing connection with IdentityService
$ClientSecret = "SIF-Default"

#Computer Names
# The username for computers to deploy Sitecore roles to.
$ComputerUserName = ""
# The password for computers to deploy Sitecore roles to.
$ComputerPassword = ""
# The DNS name or IP of the Solr computer.
$SolrComputerName = ""
# The DNS name of the Identity Server computer.
$IdentityServerComputerName = ""
# The DNS name of the Content Management computer.
$CMComputerName = ""
# The DNS name of the Content Delivery computer.
$CDComputerName = ""
# The DNS name of the Processing computer.
$PrcComputerName = ""
# The DNS name of the Reporting computer.
$RepComputerName = ""
# The DNS name of the Collection computer.
$CollectionComputerName = ""
# The DNS name of the Collection Search computer.
$CollectionSearchComputerName = ""
# The DNS name of the Reference Data computer.
$ReferenceDataComputerName = ""
# The DNS name of the Marketing Automation computer.
$MarketingAutomationComputerName = ""
# The DNS name of the Marketing Automation Reporting computer.
$MarketingAutomationReportingComputerName = ""
# The DNS name of the Cortex Processing computer.
$CortexProcessingComputerName = ""
# The DNS name of the Cortex Reporting computer.
$CortexReportingComputerName = ""

# Service URLs
# The URL for the XConnect Search Service.
$XConnectCollectionSearchService = "https://$CollectionSearchComputerName"
# The URL of the XConnect Collection service.
$XConnectCollectionService = "https://$CollectionComputerName"
# The URL of the XConnect Reference Data service
$XConnectReferenceDataService = "https://$ReferenceDataComputerName"
# The URL for the processing service.
$ProcessingService = "https://$PrcComputerName"
# The URL for the reporting service.
$ReportingService = "https://$RepComputerName"
# The URL of the Marketing Automaton Service
$MarketingAutomationOperationsService = "https://$MarketingAutomationComputerName"
```

```
# The URL of the Marteting Automation Reporting Service
$MarketingAutomationReportingService = "https://$MarketingAutomationReportingComputerName"
# The URL of the Cortex Reporting Service
$CortexReportingService = "https://$CortexReportingComputerName"
# The URL for the Identity Server password recovery, this should be the URL of the CM
Instance.
$PasswordRecoveryUrl  = "https://$CMComputerName"
# The URL of the Identity Authority
$SitecoreIdentityAuthority = "https://$IdentityServerComputerName"
# The URL of the Machine Learning server
$MachineLearningServerUrl = ""
# Allowed CORS Origins, urls of all clients should be added here
$AllowedCorsOrigins = "https://$CMComputerName"

# Packages
# The path to the Sitecore Content Delivery package.
$CDPackage = (Get-ChildItem "$SCInstallRoot\Sitecore 9.1.1 rev. 002459
(OnPrem)_cd.scwdp.zip").FullName
# The path to the Sitecore Content Management package.
$CMPackage = (Get-ChildItem "$SCInstallRoot\Sitecore 9.1.1 rev. 002459
(OnPrem)_cm.scwdp.zip").FullName
# The path to the Sitecore Content Reporting package.
$RepPackage = (Get-ChildItem "$SCInstallRoot\Sitecore 9.1.1 rev. 002459
(OnPrem)_rep.scwdp.zip").FullName
# The path to the Sitecore Content Processing package.
$PrcPackage = (Get-ChildItem "$SCInstallRoot\Sitecore 9.1.1 rev. 002459
(OnPrem)_prc.scwdp.zip").FullName
# The path to the XConnect Collection package.
$CollectionPackage = (Get-ChildItem "$SCInstallRoot\Sitecore 9.1.1 rev. 002459
(OnPrem)_xp1collection.scwdp.zip").FullName
# The path to the XConnect Collection Search package.
$CollectionSearchPackage = (Get-ChildItem "$SCInstallRoot\Sitecore 9.1.1 rev. 002459
(OnPrem)_xp1collectionsearch.scwdp.zip").FullName
# The path to the XConnect Marketing Automation package.
$MarketingAutomationPackage = (Get-ChildItem "$SCInstallRoot\Sitecore 9.1.1 rev. 002459
(OnPrem)_xp1marketingautomation.scwdp.zip").FullName
# The path to the XConnect Marketing Automation Reporting package.
$MarketingAutomationReportingPackage = (Get-ChildItem "$SCInstallRoot\Sitecore 9.1.1 rev.
002459 (OnPrem)_xp1marketingautomationreporting.scwdp.zip").FullName
# The path to the XConnect Reference Data package.
$ReferenceDataPackage = (Get-ChildItem "$SCInstallRoot\Sitecore 9.1.1 rev. 002459
(OnPrem)_xp1referencedata.scwdp.zip").FullName
# The path to the XConnect Reporting package.
$CortexReportingPackage = (Get-ChildItem "$SCInstallRoot\Sitecore 9.1.1 rev. 002459
(OnPrem)_xp1cortexreporting.scwdp.zip").FullName
# The path to the XConnect Processing engine package.
$CortexProcessingPackage = (Get-ChildItem "$SCInstallRoot\Sitecore 9.1.1 rev. 002459
(OnPrem)_xp1cortexprocessing.scwdp.zip").FullName
```

```
# The path to the Identity Server package.
$IdentityServerPackage = (Get-ChildItem "$SCInstallRoot\Sitecore.IdentityServer 2.0.1 rev.
00166 (OnPrem)_identityserver.scwdp.zip").FullName

# Install  via combined partials file.
$Parameters = @{
    Path = "$SCInstallRoot\XP1-Distributed.json"
    ComputerUserName = $ComputerUserName
    ComputerPassword = $ComputerPassword
    SolrComputerName = $SolrComputerName
    IdentityServerComputerName = $IdentityServerComputerName
    CMComputerName = $CMComputerName
    CDComputerName = $CDComputerName
    PrcComputerName = $PrcComputerName
    RepComputerName = $RepComputerName
    CollectionComputerName = $CollectionComputerName
    CollectionSearchComputerName = $CollectionSearchComputerName
    ReferenceDataComputerName = $ReferenceDataComputerName
    MarketingAutomationComputerName = $MarketingAutomationComputerName
    MarketingAutomationReportingComputerName = $MarketingAutomationReportingComputerName
    CortexProcessingComputerName = $CortexProcessingComputerName
    CortexReportingComputerName = $CortexReportingComputerName
    CertificateName = $ClientCertificateName
    LicenseFile = $LicenseFile
    SolrUrl = $SolrUrl
    SolrRoot = $SolrRoot
    SolrService = $SolrService
    Prefix = $Prefix
    SqlServer = $SqlServer
    SqlAdminUser = $SqlAdminUser
    SqlAdminPassword = $SqlAdminPassword
    IdentityServerSiteName = $IdentityServerSiteName
    CollectionSearchSitename = $CollectionSearchSitename
    MarketingAutomationSitename = $MarketingAutomationSiteName
    MarketingAutomationReportingSitename = $MarketingAutomationReportingSiteName
    ReferenceDataSitename = $ReferenceDateSiteName
    CortexProcessingSitename = $CortexProcessingSitename
    CortexReportingSitename = $CortexReportingSitename
    CollectionSitename = $CollectionSitename
    CDSitename = $ContentDeliverySiteName
    CMSitename = $ContentManagementSiteName
    RepSitename = $ReportingSiteName
    PrcSitename = $ProcessingSiteName
    XConnectCollectionService = $XConnectCollectionService
    XConnectReferenceDataService = $XConnectReferenceDataService
    XConnectCollectionSearchService = $XConnectCollectionSearchService
    MarketingAutomationOperationsService = $MarketingAutomationOperationsService
    MarketingAutomationReportingService = $MarketingAutomationReportingService
    MachineLearningServerUrl = $MachineLearningServerUrl
    SitecoreIdentityAuthority = $SitecoreIdentityAuthority
    ProcessingService = $ProcessingService
    ReportingService = $ReportingService
    CortexReportingService = $CortexReportingService
    CollectionPackage = $CollectionPackage
    CollectionSearchPackage = $CollectionSearchPackage
    CortexProcessingPackage = $CortexProcessingPackage
    MarketingAutomationPackage = $MarketingAutomationPackage
    MarketingAutomationReportingPackage = $MarketingAutomationReportingPackage
    ReferenceDataPackage = $ReferenceDataPackage
    CortexReportingPackage = $CortexReportingPackage
    CDPackage = $CDPackage
    CMPackage = $CMPackage
    RepPackage = $RepPackage
    PrcPackage = $PrcPackage
    IdentityServerPackage = $IdentityServerPackage
    PasswordRecoveryUrl = $PasswordRecoveryUrl
    ClientSecret = $ClientSecret
    AllowedCorsOrigins = $AllowedCorsOrigins
}

Push-Location $SCInstallRoot

Install-SitecoreConfiguration @Parameters *>&1 | Tee-Object XP1-Distributed.log

Pop-Location
```