Sitecore Experience Commerce Installation Guide for On-Premise Solutions

Sitecore Experience Commerce 9.1

May 17, 2019

Table of Contents

1. Getting started

This guide describes how to install Sitecore Experience Commerce™ (XC) solution in a single-server configuration (local environment).

> **NOTE**
> This document assumes that you have already installed the Sitecore Experience Platform software (using the XP Single configuration), as described in the *Sitecore Experience Platform Installation Guide*, available from dev.sitecore.net.

This chapter contains the following sections:

- Sitecore Experience Commerce solution
- System requirements
- Packages for on-premise installation

1.1. Sitecore Experience Commerce solution

Sitecore Experience Commerce (XC) is an e-commerce solution, built on the Sitecore Experience Platform (Sitecore XP).

The Sitecore XC solution provides a core framework for rapidly delivering commerce functionality through the following components:

- **Commerce Engine**
  An extensible commerce core framework, hosting commerce services such as Cart, Order, Pricing, Promotions, Catalogs, and Inventory. The Commerce Engine includes a pluggable framework for extending the engine to modify or add to existing functionality.
- **Commerce Business Tools**
  A set of rich business tools for merchandisers and customer service representatives. The business tools are built on the Angular framework, and can also be extended using the same pluggable framework.
- **Sitecore Experience Accelerator (SXA) Storefront**
  A sample storefront website that is integrated with the Commerce Engine. You can use the SXA Storefront as a starting point to building a customized storefront.

For more information about Sitecore XP, refer to the Sitecore Documentation site.

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

With Sitecore 9.0 and later, the on-premises installation of the platform occurs in the context of the Sitecore Installation Framework (SIF). The framework deploys Web Deploy Packages (WDP) by passing parameters to configuration files through a Microsoft® PowerShell module.

1.2. System requirements

This section describes the system requirements for a Sitecore XC 9.1 hosting environment.

- Hardware requirements
- Hosting environment requirements

> **NOTE**
> You need a Braintree sandbox account to enable web payment functionality through the Commerce Engine. Follow the instructions on the Braintree website to set up an account, and note the MerchantID, Public Key and Private Key information. You need to specify the values for those parameters in the Sitecore XC deployment script.

1.2.1. Hardware requirements

The minimum hardware configuration requirements for running a single Sitecore XC installation are:

- 4 core processor
- 16 GB of RAM

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

1.2.2. Hosting environment requirements

The following table lists the software requirements for Sitecore XC 9.1 hosting environment.

Operating system	Windows Server 2016 Windows 10 Professional (32/64-bit)
.NET Framework	.NET Framework 4.7.1 Developer Pack (including available updates) ASP.NET MVC 5.2 or later ASP.NET Core 2.1 .NET Core Windows Server Hosting 2.1.3 or later versions of 2.1 OData 7.2.0
Database	Microsoft SQL Server 2016 SP2 Microsoft SQL Server 2017
Web server	IIS 10.0 or IIS 8.5 (the version supported by your operating system) Microsoft Web Deploy 3.6 URL Rewriter
Development	Visual Studio 2017 + latest updates .NET Core 2.1 Visual Studio 2017 Tooling MSBuild Microsoft Visual Studio Web targets (available from Nuget) Git client

Deployment	PowerShell 5.1 or later
Runtime	Node.js
	OData Connected Service (available from Visual Studio Marketplace)
Search indexing	Solr 7.2.1
Sitecore software	Sitecore Experience Platform 9.1.1 (available on dev.sitecore.net)
	Sitecore Experience Accelerator 1.8.1 (available on dev.sitecore.net)
	Sitecore PowerShell Extensions 4.7 or 5.0 (available from the Sitecore Marketplace)
Web browsers	Microsoft Edge
	Internet Explorer 11
	Mozilla Firefox
	Google Chrome
	Apple Safari

1.3. Packages for on-premise installation

The Sitecore XC release package does not include any Sitecore XP software. You must install all Sitecore XP pre-requisite software first.

The following table lists the software packages provided with the Sitecore XC release package for an on-premises installation. Each package has a unique version number and a .zip extension.

Package	Description
SIF.Sitecore.Commerce	Contains Commerce-specific extensions to the Sitecore Installation Framework (SIF), including the master deployment script for Commerce packages, and Commerce configuration.
Sitecore.Commerce.Engine	Contains the binary for the Commerce Engine, a lightweight, micro-service based framework for the development of commerce solutions.
Sitecore.Commerce.Engine.SDK	Contains the development kit for compiling the Commerce Engine. Compiling the SDK programmatically fetches Commerce Engine plugins from Sitecore's public NuGet feed. This NuGet feed hosts plugins that are released and supported with the Sitecore XC release.
Sitecore.Biz.FX	Contains an integrated suite of Sitecore XC Business Tools, built on the Angular application platform version 4.
Sitecore.Biz.FX.SDK	Contains the development kit for compiling the Sitecore XC business tools.
Commerce Connect	
Sitecore Commerce Connect Core	Contains a middleware integration layer between the Sitecore XC back-end and the Storefront front-end, and with the Sitecore Experience Platform.
Sitecore.Commerce.Engine.Connect	Contains a thin integration layer for integrating the Commerce Engine with Sitecore Commerce Connect Core. The package file format is a Team Development for Sitecore (TDS) .update package.
Sitecore Commerce ExperienceAnalytics	Contains Commerce Experience Analytics installer and core files required for integration with Commerce.
Sitecore Commerce ExperienceProfile	Contains Commerce Experience Profile installer and core files required to integrate Experience Profile with Commerce.
Sitecore Commerce Marketing Automation Core	Contains Commerce Marketing Automation installation and core files.
Sitecore Commerce Marketing Automation for AutomationEngine	Contains Commerce Marketing automation files required to setup the Marketing Automation Engine service.
SXA Storefront	
Sitecore Commerce Experience Accelerator	Contains Commerce-specific extensions to the Sitecore Experience Accelerator (SXA) templated UX layouts (for example, UI renderings used to display a catalog on the Storefront).
Sitecore Commerce Experience Accelerator Storefront	Contains the sample and starter storefront as an integrated part of the Sitecore XC solution, built using SXA UX layouts.
Sitecore Commerce Experience Accelerator Storefront Themes	Contains the themes used for the SXA Storefront site.
Sitecore Commerce Experience Accelerator Habitat Catalog	Contains the Habitat sample catalog for the Storefront site.

Package	Description
Sitecore.Commerce.Habitat.Images	Contains images for the Habitat sample catalog.
Adventure Works Images	Contains images for the Adventure Works sample catalog.

2. Preparing for a Sitecore XC installation

This chapter outlines the tasks you must complete before installing the Sitecore XC solution on a single machine in a non-production environment.

This chapter contains the following sections:

- Verify your Sitecore XP setup
- Download the Sitecore XC release package

2.1. Verify your Sitecore XP setup

The instructions in this document assume that you have already installed the Sitecore XP software on your system, as described in the *Sitecore Experience Platform Installation Guide*.

These instructions also assume that you have .NET Core SDK 2.1.3 installed in your development environment and have the following installed in your deployment environment:

- .NET Core Windows Server Hosting 2.1.3
- Solr 7.2.1 deployment (with HTTPs enabled)
- Sitecore Experience Platform 9.1.1 (XP Single deployment)

2.1.1. Search provider setup

Solr is the recommended search provider for a single-server installation. However, Sitecore also supports Azure Search as a search provider.

If you are using Solr as a search provider, make sure that Solr is running properly before you install the Sitecore XC solution by rebuilding the search indexes and the link databases in the Sitecore client (**Control Panel > Indexing Manager**). See the *Sitecore Experience Platform Installation Guide* for detailed instructions.

In addition, make a note of the values you used for the Solr installation ($SolrUrl, $SolrRoot, $SolrService) during the Sitecore XP installation. You must specify these values in the Sitecore XC deployment script.

If you are using Azure Search as a search provider, you must create an Azure Search Service and make a note of the values for the service name, the service admin key, and the service query key. You must specify these values in the Sitecore XC deployment script.

2.1.2. SQL server setup

The Sitecore Installation Framework (SIF) installs the required databases during Sitecore XP deployment, but you must set up the correct user roles for authentication. Ensure that you add a recognized user (for example, collectionuser) to the xDB shard databases (as described in the *Sitecore Experience Platform Installation Guide*).

Sitecore XP uses SQL authentication during installation (for example, $SqlAdminUser = "sa"). Sitecore XC uses Windows Authentication during installation, which means that the user identity running the Sitecore XC deployment script is typically the current Windows logon user. This user account must have proper database permissions (for example, sysadmin at the SQL instance level) during the installation.

In addition, make a note of the value you used for the SQL installation ($SqlServer) during the Sitecore XP installation. You must specify this value in the Sitecore XC deployment script.

2.1.3. IIS setup

If the Sitecore XP IIS site name is different from its application pool name (by default, they are the same), you must change the values in the Sitecore XC deployment scripts for the `StopAppPool` and `StartAppPool` parameters in the `SXAStorefront.CreateBinding.json` file to reflect the correct application pool name.

2.2. Download the Sitecore XC release package and prerequisites

You can download the Sitecore XC release package from the Sitecore Experience Commerce Download page.

To download the Sitecore XC 9.1 release package:

1. Open a browser and go to the Sitecore Experience Commerce Download page.
2. Download the Sitecore Experience Commerce release package (`Packages for On-Premises 2019.04-x.x.xxx`).
3. Extract the contents of the zip file to the location from which you are deploying Sitecore XC (for example, `c:\deploy`).
   The extraction process results in multiple .zip files.
4. Go to the folder where you extracted the release package files and unzip the following packages:
   - `SIF.Sitecore.Commerce.x.x.xxxx.zip`
   - `Sitecore.Commerce.Engine.x.x.xxxx.zip`
   - `Sitecore.Commerce.Engine.SDK.x.x.xxxx.zip`

     NOTE
     Note the location of the `Sitecore.Commerce.Engine.DB.dacpac` file in the Sitecore Commerce Engine SDK package. You must specify this path in the Sitecore XC deployment script.
   - `Sitecore.BizFX.x.x.xxx.zip`

   Make a note of the destination folders (for example, `c:\deploy\Sitecore.BizFX.x.x.xxx\`). You must specify the path to their locations in the Sitecore XC deployment script. Leave all other files in the release package zipped.
5. Download the following software to a folder in your local deployment folder:
   - Sitecore PowerShell Extensions 4.7 or 5.0 (available from the Sitecore Marketplace)
   - Sitecore Experience Accelerator 1.8 -1 (available from the Sitecore Experience Accelerator Download page)
   - MSBuild Microsoft Visual Studio Web targets (available from Nuget)

     NOTE
     When you unpack the Web targets NuGet package, copy the `\tools\VSToolsPath\Web\Microsoft.Web.XmlTransform.dll` file into a folder and note the path.

3. Installing Sitecore XC

The Sitecore XC release package includes PowerShell scripts for installing Sitecore XC. The `Deploy-Sitecore-Commerce.ps1` script (located in the *SIF Sitecore Commerce* folder) installs all of the Sitecore XC modules and the SXA Storefront site.

This chapter contains the following sections:

3.1. Customize the deployment script

The Sitecore XC deployment script executes a single command to deploy Sitecore XC, and includes a number of parameters that you must change. Sitecore recommends that you modify the existing deployment script and save it with a new name (to preserve a record of the factory default script).

After you specify values and paths to reflect your own deployment environment you can run the script to install the solution. If there are any parameters for which no value is specified, the script prompts you to enter a value during installation.

To customize the deployment script:

1. Navigate to the `SIF Sitecore Commerce` folder that you unzipped earlier (for example, `c:\deploy\SIF.Sitecore.Commercec.x.x.xxx\`).
2. Open the `Deploy-Sitecore-Commerce.ps1` script file in a text editor.
3. Save a copy of the file with a new name (for example, `MyDeploy-Sitecore-Commerce.ps1`).
4. Specify values for the following parameters, according to your own environment. In particular, make sure that the path values correspond to the locations where you unzipped or copied files needed for the installation.

> **IMPORTANT**
>
> Consider the following when customizing the Sitecore XC deployment script:
>
> - Shared parameters used by the XC deployment script and the XP deployment script do not necessarily use the same default values. To ensure that there are no mismatch, you must change some of the XC default values used by the XC deployment script to align with the Sitecore XP parameter values.
> - The Sitecore XP installation script defines a prefix parameter (`$Prefix = "xp0"`). This parameter is then used as a variable to compose the value of other parameters, for example, `$SitecoreSiteName = "$Prefix.sc"`. The Sitecore XC deployment script does not define such `$Prefix` parameter out-of-the-box. Sitecore recommends that you add the parameter `$Prefix = "xp0"` to your Sitecore XC deployment script to avoid mismatch with the default Sitecore XP installation configuration.

Parameter	Description
`Path`	The path to the `Master_SingleServer.json` file provided in the `SIF.Sitecore.Commerce` package.
`BaseConfigurationFolder`	The path to the `SIF Sitecore Commerce` folder.
`SiteName`	Name of your Sitecore site. **NOTE** The default value set by the XC deployment script does not align with the Sitecore XP default value. Make sure to specify the value used in your Sitecore XP deployment. By default, Sitecore XP uses `"$Prefix.sc"` (for example, `"xp0.sc"`).

Parameter	Description
`SiteHostHeaderName`	Host header for your Sitecore site.
`InstallDir`	Location of the Sitecore site. **NOTE** The default value set by the XC deployment script does not align with the Sitecore XP default value. Make sure to specify the value used in your Sitecore XP deployment. By default, Sitecore XP uses `"c:\inetpub\wwwroot\$SiteName"`.
`XConnectInstallDir`	Location of the xConnect site. **NOTE** The default value set by the XC deployment script does not align with the Sitecore XP default value. Make sure to specify the value used in your Sitecore XP deployment. By default, Sitecore XP uses `"c:\inetpub\wwwroot\$Prefix.xconnect"`.
`CommerceInstallRoot`	The path to the `inetpub` folder where Commerce is installed.
`CommerceServicesDbServer`	Location of the database server where you want to deploy the Commerce databases.
`CommerceServicesDbName`	Name of the database for the Commerce Services shared database.
`CommerceServicesGlobalDbName`	Name of the database for the Commerce Services global database.
`SitecoreDbServer`	Location of the database server hosting the Sitecore databases (same value used for `$SqlServer` during Sitecore XP installation).
`SqlDbPrefix`	Parameter variable (string) used as part of the Name of the Sitecore core database (SitecoreCoreDbName).
`SitecoreCoreDbName`	Name of the Sitecore core database. Replace the `$SqlDbPrefix` string in this parameter value (variable with the Sitecore DB prefix value you used for the Sitecore XP installation (for example, “xp0”).
`SitecoreUsername`	Name of the Sitecore user account. **NOTE** If you change the default value of the Sitecore username, you must perform additional post-installation steps.
`SitecoreUserPassword`	Password for the Sitecore XP user account, as define in your Sitecore XP deployment. **NOTE** If you change the default value of the Sitecore user password, you must perform additional post-installation steps.
`CommerceSearchProvider`	The type of Search provider you are using for your deployment (Solr or Azure).
`SolrUrl`	URL for the Solr service (same value used for the Sitecore XP installation), if the Search provider is Solr.

Parameter	Description
`SolrRoot`	Path to Solr installation (same value used for the Sitecore XP installation), if the Search provider is Solr.
`SolrService`	Name of the Solr service (same value used for the Sitecore XP installation), if the Search provider is Solr.
`SolrSchemas`	Path to the Commerce Solr schemas (provided as part of the `SIF.Sitecore.Commerce` package), if the Search provider is Solr.
`CommerceServicesPostfix`	Postfix appended to Commerce services folders names and site names. The postfix allows you to host more than one Commerce installment on one server.
`CommerceServicesHostPostfix`	Postfix used as the root domain name (two-levels) to append as the hostname for Commerce services. By default, all Commerce services are configured as sub-domains of the domain identified by the postfix. **NOTE** The domain name must contain at least two levels. For example, `sc9.qa`.
`SearchIndexPrefix`	Prefix for the Search index. Using the `SiteName` value for the prefix is recommended.
`EnvironmentsPrefix`	Prefix for environments from the same artifact store.
`Environments`	Lists of Authoring environments from all artifact stores.
`AzureSearchServiceName`	Name of the Azure Search service, if the Search provider is Azure.
`AzureSearchAdminKey`	The value of the admin API-key to access the Azure Search service, if the Search provider is Azure.
`AzureSearchQueryKey`	The value of the query API-key to access the Azure Search service, if the Search provider is Azure.
`CommerceEngineDacPac`	Path to the `Sitecore.Commerce.Engine.DB.dacpac` file in the Sitecore Commerce Engine SDK folder. **NOTE** The XC deployment script uses a wildcard as part of the default path value. To ensure that the file path can be resolved, remove the wildcard and specify the actual path, for example, `"..\sdk\Sitecore.Commerce.Engine.DB.dacpac"`.
`CommerceOpsServicesPort`	Port for the Commerce Ops Service. Default value is 5015.
`CommerceShopsServicesPort`	Port for the Commerce Shops Service. Default value is 5005.
`CommerceAuthoringServicesPort`	Port for the Commerce Authoring Service. Default value is 5000.
`CommerceMinionsServicesPort`	Port for the Commerce Minions Service. Default value is 5010.
`SitecoreBizFxPort`	Port for the Commerce Business Tools. Default value is 4200.

Parameter	Description
SitecoreCommerceEnginePath	Path to the Sitecore.Commerce.Engine.zip file **NOTE** The XC deployment script uses a wildcard as part of the default path. To ensure that the file path can be resolved, remove the wildcard and specify the actual path, for example, "..\Sitecore.Commerce.Engine.3.0.163.zip".
SitecoreBizFxServicesContentPath	Path to the folder containing the contents extracted from the Sitecore.Commerce.BizFx.zip package. **NOTE** The XC deployment script uses a wildcard as part of the default path. To ensure that the file path can be resolved, remove the wildcard and specify the actual path.
CommerceEngineCertificateName	The name of the Commerce Engine certificate generated during the installation process.
SiteUtilitiesSrc	Path to the SiteUtilityPages folder (part of the SIF Sitecore Commerce package).
HabitatImagesModuleFullPath	Path to the Sitecore.Commerce.Habitat.Images.zip file.
AdvImagesModuleFullPath	Path to the Adventure Works Images.zip file
CommerceConnectModuleFullPath	Path to Sitecore Commerce Connect Core.zip file.
CommercexProfilesModuleFullPath	Path to the Sitecore XC xProfiles Reporting module.
CommercexAnalyticsModuleFullPath	Path to the Sitecore XC xAnalytics Reporting module.
CommerceMAModuleFullPath	Path to the Sitecore XC Marketing Automation Core module.
CommerceMAForAutomationEngineModuleFullPath	Path to the Sitecore XC Marketing Automation for AutomationEngine module.
CEConnectPackageFullPath	Path to the Sitecore.Commerce.Engine.Connect.update file.
PowerShellExtensionsModuleFullPath	Path to the location where you downloaded Sitecore PowerShell Extensions package.
SXAModuleFullPath	Path to the location where you downloaded Sitecore Experience Accelerator package.
SXACommerceModuleFullPath	Path to the Sitecore XC Accelerator zip file. **NOTE** You must change the default value to the following: "..\Sitecore Commerce Experience Accelerator 2.*.zip".
SXAStorefrontModuleFullPath	Path to the Sitecore XC Accelerator Storefront zip file. **NOTE** You must change the default value to the following: "..\Sitecore Commerce Experience Accelerator Storefront 2.*.zip".

Parameter	Description
`SXAStorefrontThemeModuleFullPath`	Path to the Sitecore XC Accelerator Storefront Themes zip file.
`SXAStorefrontCatalogModuleFullPath`	Path to the Sitecore XC Accelerator Storefront Habitat zip file.
`MergeToolFullPath`	Path to the location where you downloaded the `Microsoft.Web.XmlTransform.dll` file. **NOTE** This DLL file is included as part of the Nuget package you downloaded previously.
`UserName`	User name for a local account to be set up for the various application pools that are created as part of the deployment.
`UserPassword`	User password for authentication.
`BraintreeAccount`	Credentials for the Braintree account set up to enable web payment functionality through the Commerce Engine.
`SitecoreIdentityServerApplicationName`	Name of the Sitecore Identity Server.
`SitecoreIdentityServerHostName`	Host name of the Sitecore Identity Server.
`SitecoreBizFxServerName`	Name of the Sitecore BizFx Server.
`IdentityServerSiteName`	Name of the identity server site. (This parameter is used as value for other parameters, for example, `SitecoreIdentityServerApplicationName`, `SitecoreIdentityServerHostName`). **NOTE** This value should match the value defined on the Sitecore XP deployment. By default, the Sitecore XP installation script uses `$IdentityServerSiteName = "$prefix.identityserver"`.

5. If you want to install the Sitecore XC solution in a Commerce-Engine only deployment (that is, without installing the SXA Storefront), you can specify that the script omit the SXA modules in the `Install-SitecoreConfiguration` command, after the parameters block:

```
Install-SitecoreConfiguration @params –Skip
InstallSXASolrCores, InstallPowershellExtensions,
InstallSXAFrameworkModule, InstallSXAStorefrontModule,
CreateDefaultTenantAndSite, PopulateSXAIndexesSchema, RebuildSXAIndexes
```

> **NOTE**
> Add the `–Skip` provision to the installation command that is appropriate for your search provider (that is, Solr or Azure).

6. When you have made your changes to the custom deployment script, save the file.

3.2. Add a missing block to the deployment script file

A known issue with the Sitecore XC deployment script prevents the creation of a default storefront tenant and site. To ensure that running the deployment script creates a default storefront tenant and site, you must add the required block to the script file, and repackaged the script file prior to running the script.

To update the deployment script:

1. Go to the folder where you extracted the release package files and unzip the following package: `Sitecore Commerce Experience Accelerator Storefront 2.0.181.zip`.
2. Open the *Sitecore Commerce Experience Accelerator Storefront 2.0.181* folder, and, within it, unzip the `package.zip`.
3. In the *package* folder, open the `\items\master\sitecore\system\Modules\PowerShell\Script Library\CXA - Internal\Web API\CreateDefaultStorefrontTenantAndSite\{6FEC77C8-00DC-4B7B-9597-82588616A1F2}\en\1\xml` file, locate the section `Function CreateCXATenant`, and, just before that section, insert the following block:

```
Function Write-Progress
  {
    [CmdletBinding()]
    param(
        [Parameter(Mandatory = $false)]
        $Activity,
        [Parameter(Mandatory = $false)]
        $CurrentOperation,
        [Parameter(Mandatory = $false)]
        $Status,
        [Parameter(Mandatory = $false)]
        $PercentComplete,
        [Parameter(Mandatory = $false)]
        [switch]$Completed
        ) process
      {
        # do nothing
      }
  }
```

4. Save the changes.
5. Repackage (zip) the *package* folder to obtain `package.zip`.
6. Repackage (zip) the *Sitecore Commerce Experience Accelerator Storefront* to obtain a `Sitecore Commerce Experience Accelerator Storefront.zip` package.

3.3. Run the deployment script

To run your customized Sitecore XC deployment script:

> **IMPORTANT**
> Before you run the deployment script, you must perform this procedure to ensure that the Sitecore XC deployment script creates a default storefront tenant and site.

1. Launch PowerShell as an administrator.
2. Run the customized deployment script by executing the following command (using the file name you used for your deployment script), for example:

   ```
   c:\deploy\<path-to-SIF-folder>\MyDeploy-Sitecore-Commerce.ps1
   ```

   > **NOTE**
   > The deployment script creates new site bindings for Sitecore application in IIS Manager and removes the existing Sitecore application bindings.

4. Post-installation steps

Once you have successfully installed the Sitecore XC software, you must complete the following tasks to complete your deployment:

- Change the default Sitecore username and password

> **NOTE**
> This step is only required if you changed the default Sitecore username and password when you customized the XC deployment script.

- Update the Sitecore Identity service CORS endpoints
- Update the Business Tools URL
- Configure user accounts
- Update the Storefront theme (optional)

4.1. Change the default Sitecore username and password

NOTE

This step is only required if you changed the default Sitecore username and password when you customized the XC deployment script.

If you changed the default Sitecore username and password parameters when customizing the Sitecore XC installation script, the bootstrap process fails to access the database to commit the new configuration. You must make the username and password change in two other locations.

To ensure change the user name and password:

1. Open the `wwwroot\bootstrap\Global.json` file and update the configuration with the new Sitecore username and password. For example:

```
 {
        "$type":
"Sitecore.Commerce.Plugin.Management.SitecoreConnectionPolicy,
Sitecore.Commerce.Plugin.Management",
        "Host": "sxa.storefront.com",
        "SitecoreDatabase": "master",
        "UserName": "NewSitecoreUserName",
        "Domain": "sitecore",
        "Password": "NewUserPassword",
        "ConnectionLeaseTimeout": 60000,
        "AuthRequestUrl": "sitecore/api/ssc/auth/login",
        "Protocol": "https"
      }
```

2. Save your changes.

3. Open the `wwwroot\data\Environments\PlugIn.Content.PolicySet-1.0.0.json` and modify the Sitecore username and password in this location also. For example:

```
 {
  "$type": "Sitecore.Commerce.Core.PolicySet, Sitecore.Commerce.Core",
  "Id": "Entity-PolicySet-ContentPolicySet",
  "Version": 1,
  "IsPersisted": false,
  "Name": "ContentPolicySet",
  "Policies": {
    "$type":
"System.Collections.Generic.List`1[[Sitecore.Commerce.Core.Policy,
Sitecore.Commerce.Core]], mscorlib",
    "$values": [
      {
        "$type":
"Sitecore.Commerce.Plugin.Management.SitecoreConnectionPolicy,
Sitecore.Commerce.Plugin.Management",
        "Host": "sxa.storefront.com",
        "SitecoreDatabase": "master",
        "UserName": "NewSitecoreUserName",
        "Domain": "sitecore",
        "Password": "NewUserPassword",
        "ConnectionLeaseTimeout": 60000,
        "AuthRequestUrl": "sitecore/api/ssc/auth/login",
        "Protocol": "https"
      },
      {
        "$type":
"Sitecore.Commerce.Plugin.Content.DefaultContentPathsPolicy,
Sitecore.Commerce.Plugin.Content",
        "ContentPaths": {
          "$type": "System.Collections.Generic.List`1[[System.String,
mscorlib]], mscorlib",
          "$values": [
            "/sitecore/Commerce/Commerce Control Panel"
          ]
        }
      }
    ]
  }
}
```

4. Save your changes.
5. Propagate your configuration changes to the global database by performing a Commerce Engine bootstrap operation. You have two options to perform the bootstrap:
    - You can use the Environment Bootstrap API provided with the Postman samples included in the Sitecore XC Commerce Engine SDK.
    - Alternatively, you can run the Sitecore Commerce deployment script again.

> **NOTE**
> You must perform this procedure for each instance of the Commerce Engine you deploy.

4.2. Update the Sitecore Identity service CORS endpoints

After you run the Sitecore XC installation script, you must update the Sitecore Identity service Cross-Origin Resource Sharing (CORS) configuration.

To update the Sitecore Identity service CORS configuration:

1. Open the `C:\inetpub\wwwroot\XP0.identityserver\Config\production\Sitecore.IdentityServer.Host.xml` file.
2. Under the `<AllowedCorsOriginsGroup1>` section, add the following configuration block:

   ```
   <AllowedCorsOriginsGroup2>https://sxa.storefront.com|http://
   sxa.storefront.com</AllowedCorsOriginsGroup2>
   ```

3. Save the file.
4. Restart the Sitecore Identity service.

4.3. Update the Storefront theme (optional)

By default, the Sitecore XC deployment script installs the SXA Storefront site with the *Storefront Mock-up* theme.

You can assign a new Storefront-branded theme, as described in the Assign a theme topic on the Sitecore Documentation site.

You must re-publish the Storefront site for your changes to take effect.

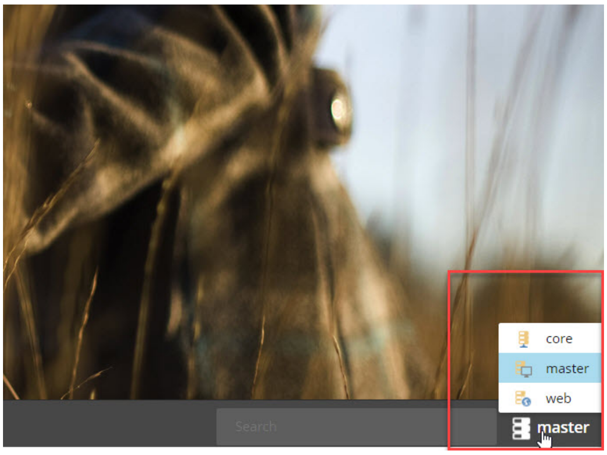
4.4. Update the Business Tools URL

By default, the **Business Tools** button on the Sitecore Launchpad points to a localhost URL.

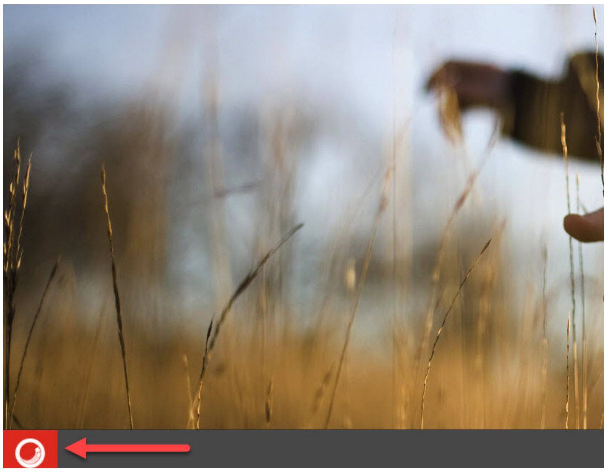
After you have deployed your Sitecore XC solution, you must update the Sitecore item in the Core database that represents the Business Tools button on the Sitecore Launchpad, so that it points to the hostname URL based on the host postfix specified during installation, for example, https://bizfx.sc9.qa.

To update the **Business Tools** button URL:

1. Open the Sitecore Launchpad and click on **Desktop**.
2. On the Desktop window, click the **master** button in the bottom right corner of the window. The system displays a list of database.



3. To switch to the Core database, click **core**.
4. In the bottom left corner of the **Desktop** window, click the Sitecore logo to open the Launchpad menu.



5. In the Launchpad menu, click **Content Editor**.
6. In the left pane of the Content Editor, navigate to the following Sitecore item: `sitecore\client\Applications\Launchpad\PageSettings\Buttons\Commerce\BusinessTools`

7. In the Business Tools window on the right, in the **Behavior** section, enter the hostname URL for the Business Tools site.
   For example, if you used the default value for the `CommerceServicesHostPostfix` parameter during installation, the value is `https://bizfx.sc9.qa`.

8. Save your changes.

4.5. Configure user accounts

After you have deployed your Sitecore XC solution, you must create user accounts and assign the appropriate roles.

> **NOTE**
>
> Every Sitecore XC user who requires access to the Business Tools must have the *Commerce Business User* role assigned, at a minimum.

You create users and assign roles using the **User Manager** tool on the **Sitecore Launchpad**.

Refer to the User roles and permissions topic for information on the the pre-defined roles and associated permissions for the Sitecore XC Business Tools.

5. Troubleshooting

This section lists common issues that can arise during the initial deployment of Sitecore Experience Commerce and proposes possible solutions.

- Default storefront site fails to load or loads without catalog navigation bar
- Install-SitecoreConfiguration: The service cannot accept control messages at this time
- Remove completed tasks from the script to resume deployment

5.1. Default storefront site fails to load or loads without catalog navigation bar

If, after completing the deployment procedure, the default storefront site fails to load or loads without displaying the catalog navigation bar, you can perform the following procedures as possible solutions:

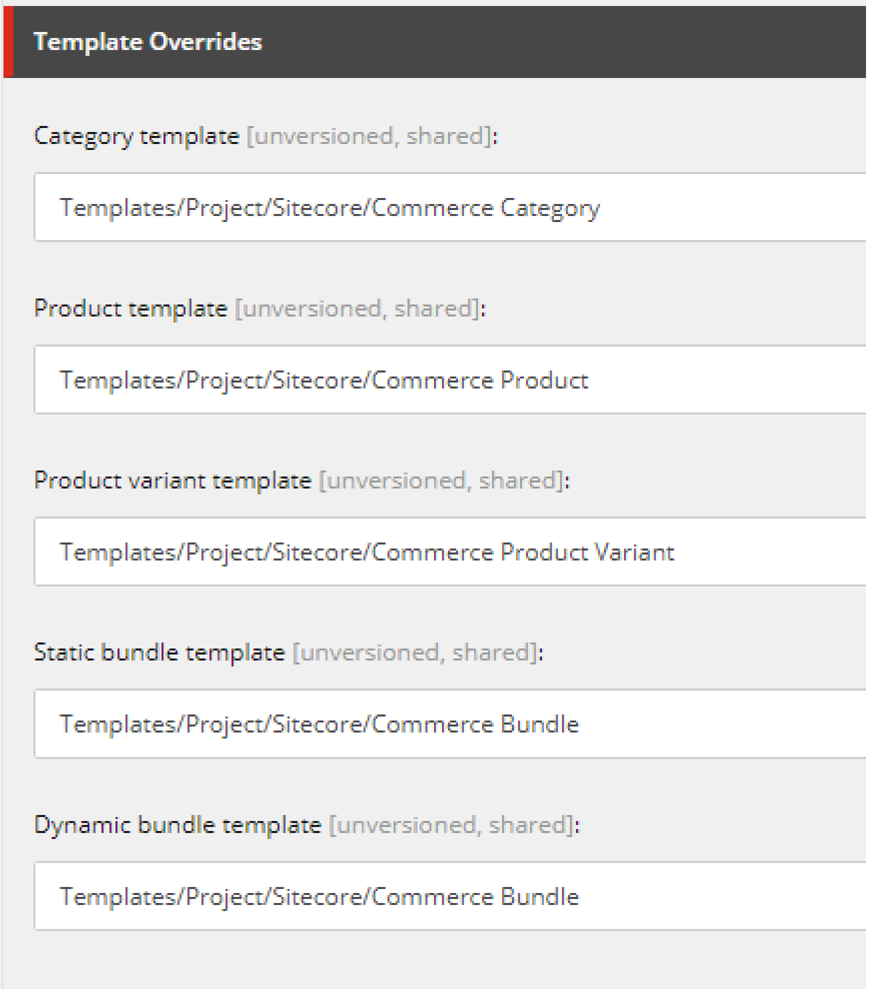
- Verify the template overrides values
- Verify catalog configuration
- Verify the site grouping configuration
- Refresh the Commerce cache and data templates
- Republish the site
- Rebuild the Solr index

5.1.1. Verify the template overrides values

If the template overrides values of the Habitat Master catalog do not match the default storefront values, the storefront site fails to load properly.

To verify the template overrides for the default storefront catalog:

1. In the Content Editor, navigate to `/sitecore/content/Sitecore/Storefront/Home/Catalogs/Habitat_Master`.
2. Go to the **Template Overrides** section, and verify that the value template override. The values for the Habitat_Master should be as follows:



Template Overrides

Category template [unversioned, shared]:
Templates/Project/Sitecore/Commerce Category

Product template [unversioned, shared]:
Templates/Project/Sitecore/Commerce Product

Product variant template [unversioned, shared]:
Templates/Project/Sitecore/Commerce Product Variant

Static bundle template [unversioned, shared]:
Templates/Project/Sitecore/Commerce Bundle

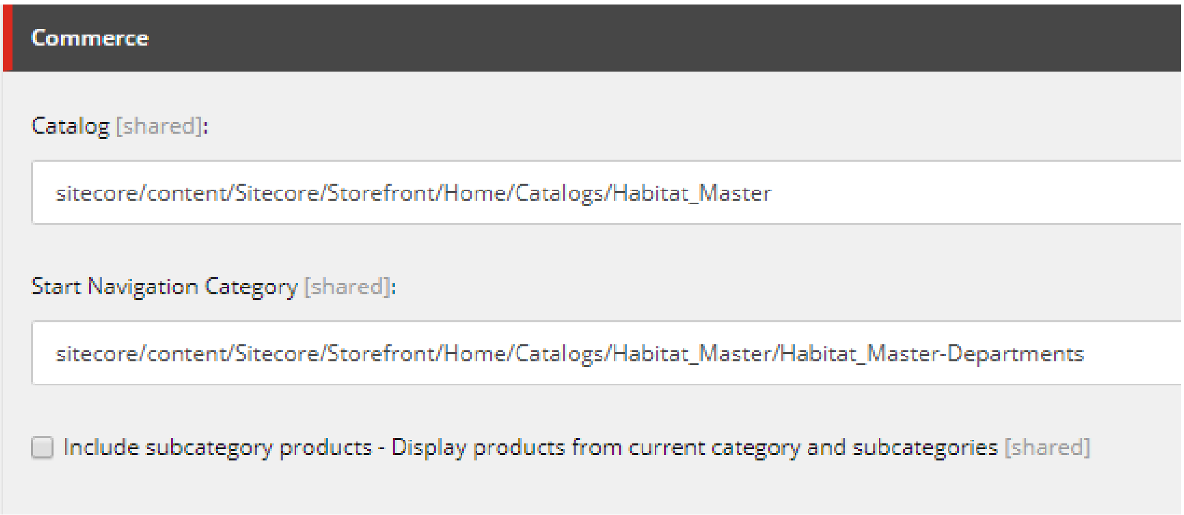
Dynamic bundle template [unversioned, shared]:
Templates/Project/Sitecore/Commerce Bundle

5.1.2. Verify the catalog configuration

If the default Habitat catalog configuration contains errors, the storefront site or catalog information may fail to load properly.

To verify the default storefront catalog configuration:

1. In the Content Editor, navigate to `/sitecore/content/Sitecore/Storefront/Settings/Commerce/Catalog Configuration`.
2. Go to the **Commerce** section, and verify that values for the **Catalog** and **Start Navigation Category** fields are set as follows:



Commerce

Catalog [shared]:

sitecore/content/Sitecore/Storefront/Home/Catalogs/Habitat_Master

Start Navigation Category [shared]:

sitecore/content/Sitecore/Storefront/Home/Catalogs/Habitat_Master/Habitat_Master-Departments

☐ Include subcategory products - Display products from current category and subcategories [shared]

5.1.3. Verify the site grouping configuration

If the host name value specified in the storefront site grouping configuration does not match the default storefront hostname, the default storefront site cannot load.

To verify the site grouping configuration:

1. In the Content Editor, navigate to `/sitecore/content/Sitecore/Storefront/Settings/Site Grouping/Storefront`.
2. Go to the **Basic** section, and verify that the field **Host Name (use* as wild card and | to list more values)** is set to *sxa.storefront.com*.

5.1.4. Refresh the Commerce cache and data templates

A staled Commerce cache or obsolete data templates may prevent the default storefront from loading or displaying properly.

To refresh the Commerce cache:

1. In the Content Editor, click the **Commerce** tab.
2. On the ribbon, click **Refresh Commerce Cache**.

To refresh data templates:

1. In the Content Editor, click the **Commerce** tab.
2. On the ribbon, click **Delete Data Templates**.
3. On the ribbon, click **Update Data Templates**.

5.1.5. Re-publish the site

If you made any changes to the storefront configuration or settings, you must republish the default storefront site.

5.1.6. Rebuild the Solr index

After you publish the storefront, you should rebuild the Solr XP index.

5.2. Install-SitecoreConfiguration: The service cannot accept control messages at this time

If the Sitecore XC deployment script fails with an error message similar to the one below, perform the following procedure.

```
Install-SitecoreConfiguration: The service cannot accept control messages at
this time. (Exception from HRESULT: 0x80070425) At C:\Program Files
\WindowsPowerShell\Modules\SitecoreInstallFramework\2.1.1\Public\Install-
SitecoreConfiguration.ps1:641 char:25 + & $entry.Task.Command @paramSet | Out-
Default + ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ + CategoryInfo : NotSpecified: (:) [
```

To resolve this issue:

1. In IIS manager, re-start the app pool and the application.
2. Run the Commerce deployment script again.

5.3. Remove completed tasks from the script to resume deployment

As a general rule, when a script task fails during execution, you can remove the completed tasks from the `\Configuration\Commerce\Master_SingleServer.json` file, and attempt to resume the deployment.

To remove completed tasks from the script:

1. In the folder where you deployed the `SIF.Sitecore.Commerce` package, open the `Configuration\Commerce\Master_SingleServer.json`.
2. In the `Master_SingleServer.json`, locate the `"Tasks"` section, remove completed tasks, and save your changes.
3. Run the `Deploy-Sitecore-Commerce.ps1` script again. The script performs the remaining deployment tasks.