Sitecore XC 10.2 Installation Guide for On-Premises Solutions

Sitecore Experience Commerce 10.2

November 10, 2021

Table of Contents

1. Getting started

This guide describes how to install a Sitecore Experience Commerce™ (XC) solution on-prem, in a single-server configuration, in a local environment or on a hosted virtual machine.

> **NOTE**
> This document assumes that you have already installed the Sitecore Experience Platform software, using the XP Single Instance (XP0) configuration, as described in the Sitecore XP Quick Installation Guide for a Developer Workstation.

This chapter contains the following sections:

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 Sitecore Support.

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 10.2 hosting environment.

The Sitecore Experience Commerce compatibility matrix is available in this Sitecore Knowledge Base article

- 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**
> These are 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 10.2 hosting environment.

> **IMPORTANT**
> Address all system requirements before you proceed with the Sitecore XC deployment or upgrade.

Operating system	• Windows Server 2016 • Windows Server 2019 • Windows 10 Professional (64-bit)
.NET Framework	The latest available version of the ASP.NET Core Runtime 3.1. Hosting Bundle

Database	• Microsoft SQL Server 2019
	• Microsoft SQL Server 2017
	• Microsoft Azure SQL
	• Redis data store
	  • Redis (Linux): 4.0.9 (or later)
	  • Redis (Azure): 4.0.14 (or later)
	  • Redis (Windows): 3.0.504 (or later)
	
	NOTE
	Sitecore recommends using Redis for Linux for production installations. The version for Windows should only be used for development or non-production single machine deployments.

Web server	• IIS 10.0 • Microsoft Web Deploy 3.6 • URL Rewriter
Development	• Visual Studio 2019 • The latest available version of .NET Core SDK 3.1 • 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 take note the path. • OData Connected Service (available from Visual Studio Marketplace) • Node.js (for BizTools customization)
Deployment	PowerShell 5.1 or later
Search indexing	• Solr 8.8.2 • SolrCloud
Sitecore software	• Sitecore Experience Platform 10.2 (available on dev.sitecore.net) • Sitecore Experience Accelerator (SXA) 10.2 (available on dev.sitecore.net) • Sitecore PowerShell Extensions 6.3 for Sitecore 10.2 (available with SXA downloads on dev.sitecore.net) • Sitecore Identity 6.0.0

1.3. Packages for on-premises installation

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

The following table lists the software packages provided with the Sitecore XC release package for an on-premises installation.

The SIF.Sitecore.Commerce package contains Sitecore Installation Framework scripts and Web Deployment Packages (WDP). Each package has a unique version number.

Package	Description
SIF.Sitecore.Commerce	Contains Commerce-specific extensions to the Sitecore Installation Framework (SIF), including the master deployment script for Commerce packages, Commerce configuration.
Sitecore.Commerce.Engine.OnPrem.Solr	Contains the binary (pre-compiled code) for the Commerce Engine, a lightweight, micro-service based framework for the development of commerce solutions. For deployment using Solr search
Sitecore.Commerce.Engine.SDK	Contains the development kit (source code) 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. Available as ZIP package.
Sitecore.BizFX.OnPrem	Contains an integrated suite of Sitecore XC Business Tools (pre-compiled code), built on the Angular application platform version 4.
Sitecore.Biz.FX.SDK	Contains the development kit (source code) for compiling the Sitecore XC business tools. Available as ZIP package.
SolrSchemas.Sitecore.Commerce	Contains Commerce-specific Solr schemas to support search for Commerce entities (CatalogItems, Customers, Orders, PriceCards, and Promotions).
Commerce Connect	
Sitecore Commerce Connect Core OnPrem	Contains a middleware integration layer between the Sitecore XC back-end and the Storefront front-end, and the Sitecore Experience Platform. It also contains the Commerce marketing automation campaign templates.
Sitecore Commerce Engine Connect OnPrem	Contains a thin integration layer for integrating the Commerce Engine with Sitecore Commerce Connect Core. Available as ZIP package.
Sitecore Commerce ExperienceAnalytics Core OnPrem	Contains Commerce Experience Analytics installer and core files required for integration with Commerce.
Sitecore Commerce ExperienceProfile Core OnPrem	Contains Commerce Experience Profile installer and core files required to integrate Experience Profile with Commerce.
Sitecore Commerce Marketing Automation Core OnPrem	Contains Commerce Marketing Automation installation and core files as well as the default Commerce Marketing Automation templates.

Package	Description
Sitecore Commerce Marketing Automation for AutomationEngine	Contains Commerce Marketing automation files required to setup the Marketing Automation Engine service. Available as ZIP package.
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) as well as the Commerce email campaign templates.
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.
Sitecore.Commerce.Habitat.Images.OnPrem	Contains images for the Habitat sample catalog.
Adventure Works Images.OnPrem	Contains images for the Adventure Works sample catalog.

NOTE

In addition to the above, the Sitecore XC release package includes other packages that are not used in the deployment of the sample Commerce solution on-prem.

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:

2.1. Verify your Sitecore XP and Sitecore SXA 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*.

If you are installing the SXA Storefront, this document assumes you have already installed the Sitecore Experience Accelerator module as described in the *Sitecore Experience Accelerator Installation Guide.*

These instructions also assume that you have .Net Core SDK 3.1.302 (or latest) installed in your development environment and have the following installed in your deployment environment:

- Windows Server hosting .Net Core 3.1.6
- Solr 8.8.2 deployment (with HTTPs enabled)
- Sitecore Experience Platform 10.2 (XP Single deployment)
- Sitecore Identity (6.0.0)

2.1.1. Search provider setup

Solr is the supported search provider for Sitecore XC on-prem installation.

The Sitecore XC deployment script assumes that your deployment uses Solr.

Sorl is installed with Sitecore XP. You must make sure that Solr is running properly before you install the Sitecore XC solution. Rebuild the search indexes and the link databases in the Sitecore client (**Control Panel > Indexing Manager**). See the *Quick Installation Guide for a Developer Workstation* for detailed instructions. .

In addition, make a note of the values you used for Solr (`$SolrUrl`, `$SolrRoot`, `$SolrService`) during the Sitecore XP installation. 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 shared 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.

2.2. Download the Sitecore XC release package and prerequisites

You can download the Sitecore XC release package (`Packages for On Premise WDP 2021.xx-x.x.xxx`) from the Sitecore Experience Commerce Downloads page.

To download the Sitecore XC 10.2 release package and prerequisites:

1. Download and install Redis.

   NOTE

   - Sitecore recommends using Redis for Linux for production installations. Redis for Windows should only be used for development or non-production, single machine deployments.
   - The deployment script assumes a developer scenario where Redis and XC are installed on the same host. If your development environment is using a remote Redis installation, you need to perform additional post-installation steps to establish connectivity with the remote Redis server.

2. Open a browser and go to the Sitecore Experience Commerce Downloads and access the page for Sitecore Experience Commerce 10.2. release downloads page.
3. In the section Download Options for On Premise Deployments, download the Sitecore Experience Commerce Packages for On Premise WDP (`Packages for On Premises WDP 2021.xx-x.x.xxx`).
4. 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.
5. Extract the content of the `SIF.Sitecore.Commerce.zip` file.
6. Download the following software package to a folder in your local deployment folder:
   - 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.

2.3. Create a secret client certificate

The Commerce Engine Connect clients authenticate with the Sitecore Identity server using a client secret. You must create an authentication secret as a prerequisite to running the Sitecore XC deployment script. The client authentication secret is a required parameter value that you must provide when you customize the deployment script.

There are many ways to create client authentication secrets, but following is an example using PowerShell:

```
$bytes = New-Object Byte[] 32
$rand = [System.Security.Cryptography.RandomNumberGenerator]::Create()
$rand.GetBytes($bytes)
$rand.Dispose()
$newClientSecret = [System.Convert]::ToBase64String($bytes)
```

NOTE

Take note of the generated secret value so that you have it at hand when you customize the the deployment script.

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 required parameter values and paths to reflect your own deployment environment, you can run the script to install the solution.

By default, Sitecore XC parameters values are aligned with the Sitecore Experience Platform (XP0) default parameters values. If your Sitecore XP deployment uses custom value, make sure to customize the Sitecore XC deployment script accordingly.

For your convenience, the script file provides a brief description of each parameter.

> **NOTE**
> The deployment script requires that you provide a client secret to authenticate with Commerce Engine Connect. Deployment fails if you omit to provide a value for parameter `$CommerceEngineConnectClientSecret = ""`. Make sure to generate a client secret prior to customizing the script.

To customize the deployment script:

1. Navigate to the `SIF Sitecore Commerce` folder that you unzipped earlier (for example, `c:\deploy\SIF.Sitecore.Commerce.x.x.xxx\`).
2. In a text editor, open the `Deploy-Sitecore-Commerce.ps1` script file.
3. Save a copy of the file with a new name (for example, `MyDeploy-Sitecore-Commerce.ps1`).
4. Specify values for 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. The script provides a description of each parameter.

   > **NOTE**
   > Use a double-backslash character in parameter values that specify named server instance (`"<servername>\\<instanceName>"`). For example, `CommerceServicesDbServer = "SQLServerName\\SQLInstanceName`.

5. By default, the script deploys the SXA Storefront. If you want to install the Sitecore XC solution in a Commerce-Engine only deployment (that is, without installing the SXA Storefront), set the value of the parameter `$skipInstallDefaultStorefront` to `true`. The script predefines the list of tasks to skip.

   > **NOTE**
   > When you skip the installation of the SXA Storefront, the script still deploys all SF-related packages. To avoid deploying unnecessary storefront packages, set the `$SkipDeployStorefrontPackages` parameter to `true`.

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

3.2. 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.Preconfigure.json` file to reflect the correct application pool name.

3.3. Run the deployment script

To run your customized Sitecore XC deployment script:

1. In IIS Manager, restart the Sitecore Identity service.

   > **IMPORTANT**
   > After you have restarted the service, do not browse to the Sitecore Identity service. Run the deployment script (by performing the following steps) before accessing the Sitecore Identity service.

2. Launch PowerShell as an administrator, and navigate to: `C:\deploy\<path-to-SIF-folder>`.
3. Run the customized deployment script by executing the following command (using the file name you used for your deployment script), for example: `.\MyDeploy-Sitecore-Commerce.ps1`

   > **NOTE**
   > The installation script performs tasks that update the Sitecore databases. If you attempt to re-run the Sitecore XC deployment script after a Sitecore database was altered by a previous installation attempt, the script fails with an error message. To avoid this error, you can modify the script to skip or remove the tasks previously completed, and run the script again. Otherwise, you must uninstall and re-install Sitecore Experience Platform and Sitecore Commerce XC.

4. Post-installation steps

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

- Update the Storefront theme (optional)
- Configure user accounts
- Configure connectivity with a remote Redis server (optional)

4.1. Update the Storefront theme (optional)

By default, the Sitecore XC deployment script installs the SXA Storefront site with the storefront-branded theme.

Optionally, you can assign a different theme (for example, a custom theme), as described in the Create a new tenant and site topic on the Sitecore Documentation site.

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

4.2. 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 pre-defined roles and associated permissions for the Sitecore XC Business Tools.

4.3. Configure connectivity with a remote Redis server (optional)

The default deployment script assumes that Redis and Sitecore Experience Commerce (XC) are deployed on the same host. If your Sitecore XC developer environment uses a remote instance of Redis, you must perform additional steps to establish connectivity with Redis.

> **NOTE**
> You must perform the following procedures on each instance of the Commerce Engine in your deployment.

1. Change caching options in the Commerce engine configuration
2. Add a remote Redis server to Commerce Engine Connect configuration
3. Restart the Commerce Engine in IIS

4.3.1. Change caching options in the Commerce engine configuration

The Commerce Engine `config.json` file defines default caching options that you must change when using a remote Redis instance.

To configure the Commerce Engine to use a remote Redis instance:

- Open the `c:\inetpub\wwwroot\<COMMERCE_ENGINE_INSTANCE_NAME>\wwwroot\config.json` file, locate the `"Caching"` section and, in the Redis `Options` subsection, change the default `"Configuration": "localhost"` to specify the machine name hosting your Redis server. For example:

```
"Caching": {
    "Memory": {
                "Enabled":  false,
                "CacheStoreName":  "Commerce-Memory-Store"
               },
    "Redis":  {
                "Enabled":  true,
                "CacheStoreName":  "Commerce-Redis-Store",
                "IntervalBetweenConnectionAttemptsInSeconds":  60
                "RedisConnectionPoolSize":  1,
                "RedisCompressionEnabled":  true,
                "Options":  {
                              "Configuration":  "<MACHINENAME>:<PORT>",
                              "InstanceName":  "Redis"
                            }
                }
          },
```

> **NOTE**
> You do not need to specify the Redis port number when using the default port for Redis (6380). If your Redis deployment does not use the default, you must specify the port number in the configuration (for example `"Configuration": "localhost:6398"`).

4.3.2. Add a remote Redis server to Commerce Engine Connect configuration

To establish connectivity with a remote Redis server:

1. In the `c:\inetpub\wwwroot\<site>\App_Config\Include\Y.Commerce.Engine` folder, open the `Sitecore.Commerce.Engine.Connect.config` file.
2. In the cache settings section of the files, change the configuration to specify the machine name and port where the Redis server is running, and the named instance.
   For example:

> **NOTE**
> Configuration options used in this procedure are provided as example only. Use configuration options required in your deployment.

```
<redis type="Sitecore.Commerce.Engine.Connect.RedisCacheSettings,
Sitecore.Commerce.Engine.Connect.Caching">
    <enabled>true</enabled>
    <cacheStoreName>Commerce-Connector-Redis</cacheStoreName>
    <options type="Microsoft.Extensions.Caching.Redis.RedisCacheOptions,
Microsoft.Extensions.Caching.Redis">
        <configuration>MACHINENAME:PORT,
defaultDatabase=1,allowAdmin=true,syncTimeout=3600000</configuration>
        <instanceName>MACHINENAME</instanceName>
    </options>
</redis>
```

> **NOTE**
> You do not need to specify the Redis port number when using the default port for Redis (6380). If your Redis deployment does not use the default, you must specify the port number in the configuration (for example `localhost:6398`).

4.3.3. Restart the Commerce Engine in IIS

After you have completed the configuration changes, restart the Commerce Engine and Sitecore instances in IIS for the new configuration to take effect.

5. Uninstall Sitecore XC

You can uninstall Sitecore XC when you want to start with a fresh deployment. You use the Sitecore Installation Framework (SIF) to uninstall the Sitecore XC on-premises topology.

> **WARNING**
> When you uninstall Sitecore XC, your Sitecore Experience Platform (XP) deployment becomes nonfunctional. You must also uninstall Sitecore XP.

To uninstall Sitecore Experience Commerce:

1. In the `Deploy-Sitecore-Commerce.ps1` script (or the new script you created), change the `Install-SitecoreConfiguration` command to `Uninstall-SitecoreConfiguration`. For example,
   Change:

   ```
   Install-SitecoreConfiguration @deployCommerceParams -Verbose *>&1 | Tee-Object
   "$XCSIFInstallRoot\XC-Install.log
   ```

   To:

   ```
   Uninstall-SitecoreConfiguration @deployCommerceParams -Verbose *>&1 | Tee-Object
   "$XCSIFInstallRoot\XC-Uninstall.log
   ```

   > **NOTE**
   > If the SQL database server in your Commerce deployment uses a named instance, you must remove the extra backslash character ("\") from the sql named instance in the script. For example, if you deployed using `ss-mysqlinstance-01-cn\\sql2017`, you must update the SQL database server name in the script to `ss-mysqlinstance-01-cn\sql2017` before running the uninstall command.

2. Execute the script to remove the topology.
3. Uninstall Sitecore XP. For instructions on how to uninstall your Sitecore XP single topology deployment, refer to the Sitecore XP *Quick Installation Guide for a Single Developer Workstation*, available on the Sitecore Download site.

6. Troubleshooting

This section contains a list of issues that can arise during the initial deployment of Sitecore Experience Commerce, and proposes possible solutions.

6.1. Installation script fails to create a tenant and storefront site

If the installation process fails to create a tenant and default storefront site, you can manually create a tenant and storefront site by following steps described in this topic.

6.2. 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 XP index

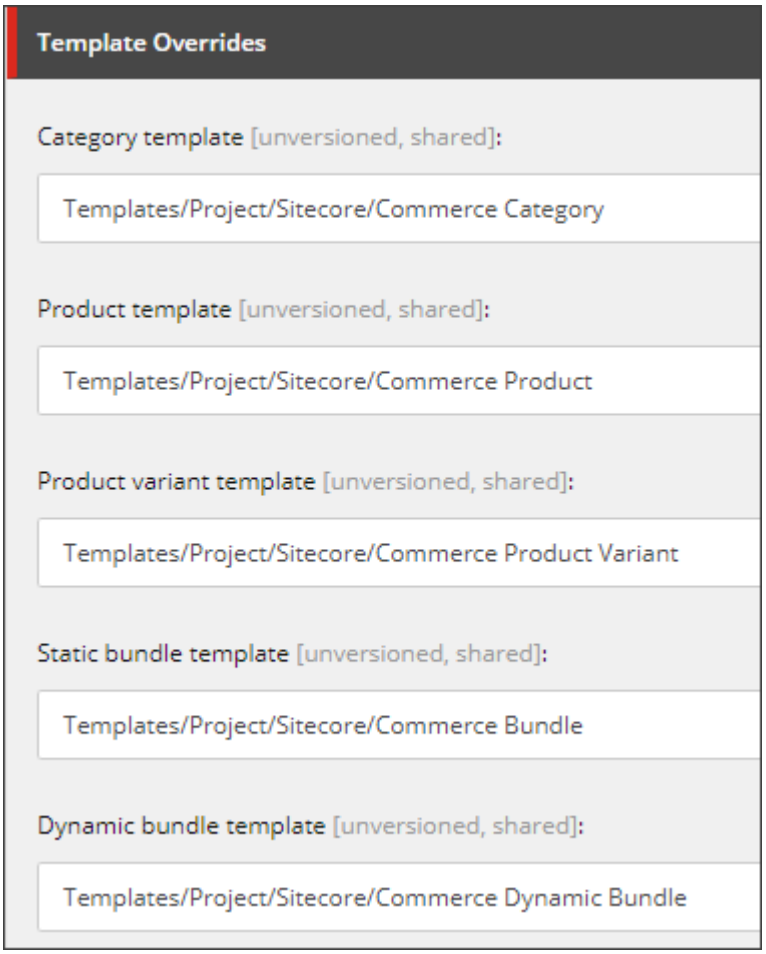
For additional troubleshooting solutions, see Troubleshooting SXA Storefront.

6.2.1. 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:

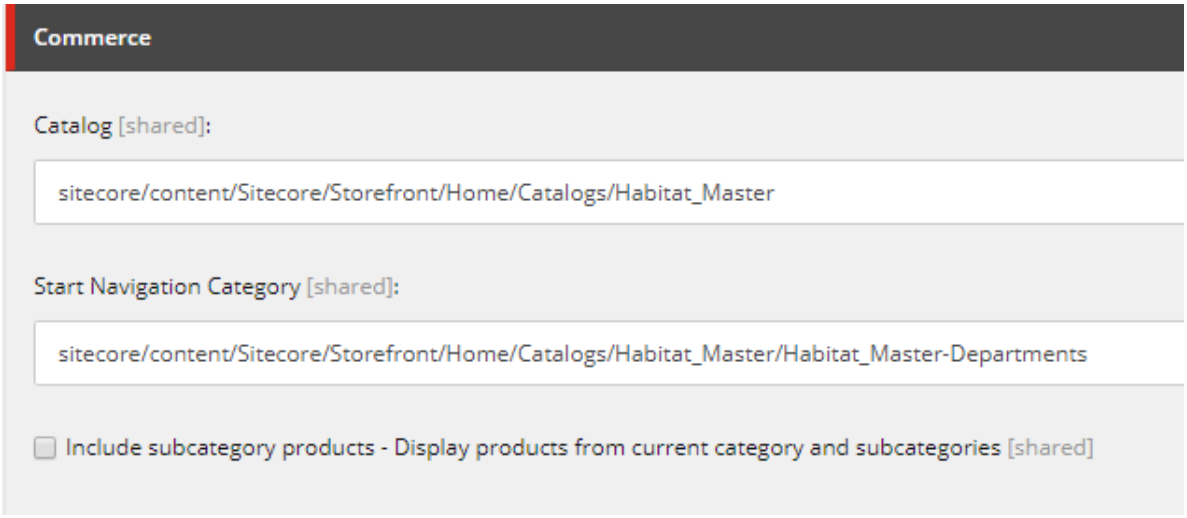


6.2.2. Default storefront site fails to load or loads without catalog navigation bar

If the default Habitat catalog configuration contains errors, the storefront site or catalog information may fail to load properly. The root cause of this error might be the catalog configuration.

To verify the 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:



6.2.3. 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*.

6.2.4. 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**.

6.2.5. Rebuild the Solr XP index

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

6.2.6. Re-publish the site

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

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

The Sitecore XC deployment script might fail, with an error message similar to the following:

```
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 deployment script again.

6.4. Remove a module or task from the deployment script

As a general rule, when a task in the `Deploy-Sitecore-Commerce.ps1` script fails during execution, you can remove preceding tasks that completed successfully from the script file before you attempt to run the deployment script again.

Before running the deployment script again, you can either remove the reference to a specific module (for example, if all tasks defined in that module's JSON file successfully executed), or you can remove specific completed tasks in a module's JSON file.

6.4.1. Remove a reference to a module

To remove a reference to a module from the `Master_SingleServer.json` file:

1. In the folder where you deployed the `SIF.Sitecore.Commerce` package, open the `Configuration/Commerce/Master_SingleServer.json` file and, in the file, locate the `"Includes"` section. The following shows an example of the `"Includes"` section and some of the modules it contains:

```
},
  "Includes": {
    "IdentityServer": {
      "Source": ".\\Configuration\\Commerce\\IdentityServer\\IdentityServer.Config.json"
    },
    "Solr": {
      "Source": ".\\Configuration\\Commerce\\Solr\\Solr.Install.json"
    },
    "CommerceEngine": {
      "Source": ".\\Configuration\\Commerce\\CommerceEngine\\CommerceEngine.json"
    },
    "BizFx": {
      "Source": ".\\Configuration\\Commerce\\SitecoreBizFx\\SitecoreBizFx.Install.json"
    },
    "StoreFront-PreconfigureInstance": {
      "Source": ".\\Configuration\\Commerce\\SXAStorefront\
\SXAStorefront.Preconfigure.json"
    },
    "Module-DirDst-EnsurePath": {
      "Source": ".\\Configuration\\Commerce\\Common\\Common.EnsurePath.json"
    },
    "Module-PowershellExtensions": {
      "Source": ".\\Configuration\\SitecoreUtilities\\InstallModule.json"
    },
    "Module-SXAFramework": {
      "Source": ".\\Configuration\\SitecoreUtilities\\InstallModule.json"
    },
    "Publish-Extensions": {
      "Source": ".\\Configuration\\Commerce\\Common\\Common.PublishToWeb.json"
    },
    "Module-HabitatImages": {
      "Source": ".\\Configuration\\Commerce\\Common\\Common.InstallWdpModule.json"
  ...
}
```

2. Delete the reference to the module you want to remove from the deployment process, and save the changes.
   For example, to exclude the module that deploys the `Sitecore.Commerce.Habitat.Images.OnPrem.scwdp.zip` package, delete the following block from the `Master_SingleServer.json` file: `"Module-HabitatImages": { "Source": ".\\Configuration\\Commerce\\Common\ \Common.InstallWdpModule.json"`.

3. Run the `Deploy-Sitecore-Commerce.ps1` script again. The script deploys the remaining modules.

6.4.2. Remove a completed task from a module

To remove a task from module's JSON file:

1. In the folder where you deployed the `SIF.Sitecore.Commerce` package, open the `Configuration\Commerce/<MODULE><FILENAME.json>`, and locate the `"Tasks"` section. The following shows an example of the `"tasks"` section in the `Configuration/Commerce/CommerceEngine/CommerceEngine.Initialize.json` file.

```
},
 "Tasks": {
    "DisableCsrfValidation":
    "RestartWebAppPoolOps":
    "GetIdServerToken":
    "BootStrapCommerceServices": {
    "EnsureSyncDefaultContentPaths": {
      "Type": "EnsureSyncDefaultContentPaths",
      "Params": {
        "UrlEnsureSyncDefaultContentPaths":
"[variable('UrlEnsureSyncDefaultContentPaths')]",
        "UrlCheckCommandStatus": "[variable('UrlCheckCommandStatus')]",
        "Environments": "[parameter('Environments')]"
      }
    },
}
```

2. Remove completed tasks, and save your changes.
3. Run the `Deploy-Sitecore-Commerce.ps1` script again. The script performs the remaining deployment tasks.

6.4.3. Skip specific tasks

Sitecore Installation Framework (SIF) allows you skip specific tasks by passing the `-Skip` parameter to the `Install-SitecoreConfiguration` function.

For example, you can use `Install-SitecoreConfiguration @params -Skip Module-PowershellExtensions_CheckPaths,Module-PowershellExtensions_InstallModule` to skip the following tasks during deployment: `Module-PowershellExtensions_CheckPaths` and `Module-PowershellExtensions_InstallModule`.

6.5. Orders information in Experience Analytics reports and XP is not up-to-date

To make sure that the most up-to-date orders information is visible in Sitecore Experience Analytics reports and in Sitecore Experience Platform, you must deploy all marketing definitions and taxonomies.

6.6. HTTP Error 500.30 - ANCM In-Process Start Failure

If your Commerce Engine fails to start and reports error “HTTP Error 500.30 - ANCM In-Process Start Failure”, make sure that, in the `Deploy-Sitecore-Commerce.ps1` file, you use a double-backslash (\\) character in parameter values that specify a server name and a named instance (`"<serverName>\\<instanceName>`).

For example, `CommerceServicesDbServer = "SQLServerName\\SQLInstanceName"`