Sitecore Experience Commerce Container Upgrade Guide

Instructions for upgrading a containerized environment from XC 10.0 to XC 10.1

March 2, 2021

Table of Contents

1. Introduction

This guide describes how to upgrade a Sitecore Experience Commerce™ (XC) 10.0 containerized solution to Sitecore Experience Commerce (XC) 10.1.

> **WARNING**
>
> Prior to performing the Sitecore XC upgrade, you must ensure to download and review the release notes and the listed breaking changes.

1.1. Prepare for the XC container upgrade

To prepare for upgrading a containerized Commerce solution, download the following packages from the Sitecore Downloads site:

- The Sitecore XC Packages for On Premises WDP
- The Sitecore Commerce Database Upgrade Scripts for Containers package
- The Sitecore Commerce Containers SDK package
- The *Upgrade Guide for XC 10.0 to 10.1 for On Premises*

In addition to the above, you must also download the Sitecore XP Container Upgrade Guide for instruction on how to upgrade the SQL server databases of an existing Sitecore XP 10.0.X to Sitecore XP 10.1.0.

2. Prerequisites

These upgrade instructions assume you are upgrading a containerized Sitecore XC 10.0 environment to Sitecore XC 10.1.

> **WARNING**
> Before you begin the upgrade process, you must backup your Commerce Engine.

These instructions assume that your current Sitecore XC deployment (before the upgrade) uses the following software:

- Sitecore Experience Platform (XP) 10.0
- Sitecore Experience Accelerator (SXA) 10.0
- Sitecore Experience Commerce (XC) 10.0

2.1. Software Requirements

The following are software requirements for installing Sitecore Experience Commerce 10.1 with containers on your developer workstation:

- Operating system:

    - Windows 10 - The most recent 2 semi-annual feature releases.
      or

    - Windows Server - The current LTS version and most recent 2 semi-annual feature releases.

- Microsoft PowerShell 5.1

- Docker Desktop for Windows

- Sitecore Commerce Container SDK

> **NOTE**
> You must extract the Docker Compose configuration folder for the desired topology.

See this article for additional details about Sitecore XC software compatibility for this release.

3. Overview of the Sitecore XC container upgrade process

The following summarizes the steps you perform to upgrade a Sitecore XC 10.0 containerized solution to Sitecore XC 10.1 containers.

- Upgrade Sitecore XP databases.

- Upgrade Sitecore SQL databases.

- Upgrade Commerce SQL databases.

- Build and deploy your custom Commerce solution using XC 10.1 containers.

- Perform post-upgrade steps.

4. Upgrade Sitecore Experience Platform

Sitecore Experience Platform provides an upgrade container that you use to upgrade Sitecore SQL databases. The Sitecore Upgrade container upgrades Sitecore XP and xConnect databases that are stored on-prem in containers, or in an Azure SQL environment.

To upgrade Sitecore XP, follow the instructions provided in the *Sitecore Experience Platform Upgrade Container Deployment Guide* (available on the Sitecore Experience Platform 10.1 download page).

You must complete the following as part of the Sitecore XP upgrade before you upgrade your Sitecore Experience Commerce containerized solution:

- Upgrade the Sitecore SQL databases

- Clean up your environment

- Upgrade Sitecore Experience Accelerator (SXA) in containers.

> **IMPORTANT**
> Sitecore Experience Accelerator is a prerequisite for the SXA Storefront. Make sure to update the SXA databases as part of the Sitecore Experience XP upgrade.

5. Upgrade Sitecore SQL databases for Commerce

Prior to building your containerized Commerce solution for Sitecore Experience Commerce 10.1, you must update the existing Sitecore master databases with the latest changes for Commerce. The `Sitecore Commerce Database Upgrade Scripts for Containers` package available on Sitecore Downloads website includes the required scripts.

To upgrade the Sitecore SQL databases for Commerce:

1. Unzip the `Sitecore Commerce Database Upgrade Scripts for Containers.zip` package.

2. Open a connection to SQL Server Manager, and run the following scripts against the `Sitecore.Master` database in the order provided:

    -   `Sitecore Commerce Engine Connect_master.sql`

    -   `Sitecore Commerce Experience Accelerator_master.sql`

    -   `Sitecore Commerce Experience Accelerator Storefront_master.sql`

    -   `Sitecore Commerce Experience Accelerator Storefront Themes_master.sql`

6. Upgrade the Commerce databases

This release of Sitecore Experience Commerce contains changes to Commerce SQL databases schema and stored procedures. The `Sitecore Commerce Database Upgrade Scripts for Containers` package available on Sitecore Downloads website includes the scripts required to update the Commerce databases.

> **NOTE**
> The upgrade scripts use default global and shared environment database names. Before you run the scripts, you must modify them to match the names of databases in your deployment.

To upgrade Sitecore XC SQL database schema with the changes contained in this release:

1. Open the SQL Server Management application.

2. Navigate to the unzipped `Sitecore Commerce Database Upgrade Scripts for Containers` package, and locate the database schema update script (`SitecoreCommerce10.1.0_SchemaUpgrade.sql`)

3. Run the following script and execute against both the *SitecoreCommerce_Global* and *SitecoreCommerce_SharedEnvironments* databases:

   - `SitecoreCommerce10.1.0_SchemaUpgrade.sql`

4. If you plan on using the Entity Archiving feature introduced in this release, you must set up the `ArchiveDataBase`. To set up the ArchiveDatabase, open the `Sitecore.Commerce10.1.0_CreateArchiveDatabase.ps1` script file, and modify the following parameters to match your environment:

   - `$severname` – the connection to your SQL server.

   - `$archiveDatabaseName` – the name of the database for archiving.

   - `$dacpacPath` – the path to the `Sitecore.Commerce.Engine.SharedArchive.DB.dacpac` file.

     > **NOTE**
     > In the `Sitecore.Commerce.Engine.SDK.*` package, the `Sitecore.Commerce.Engine.SharedArchive.DB.dacpac` file might be called `Sitecore.Commerce.Engine.DB.dacpac`.

5. Modify the following authentication settings in the script to match the appropriate authentication credentials:

   - On line 26, update the `/TargetUser:"<Username>" / TargetPassword:"<Password>"` settings.

6. Save your changes to the script file.

7. Run the script.

NOTE

After the script is completed, you see the new
`SitecoreCommerce_ArchiveSharedEnvironments` database. The schema of
the new archive database matches the schema of existing commerce
databases.

7. Build and deploy your custom Commerce solution using XC 10.1 containers

To build your custom solution and deploy the individual Sitecore roles using Sitecore XC 10.1 containers:

1. Download and use the latest released code base from Sitecore XC Packages for On Premises WDP.

2. Consume all breaking changes listed in the release notes, and ensure to resolve all conflicts.

   > **IMPORTANT**
   > Your custom XC solution must fully align with the new Sitecore Commerce release. For a summary of all policy and configuration changes in Sitecore XC 10.1 that might affect your Commerce Engine customizations, refer to the *Sitecore Upgrade Guide for Sitecore XC 10.0 to XC 10.1*, available on the Sitecore Downloads site.

3. Use the new Sitecore Commerce Docker images included in the `Sitecore.Commerce.Container.SDK` package as a base in the Docker files for the corresponding roles, and rebuild your custom Commerce solution images.

4. Deploy the containerized environment using the new Sitecore Commerce roles images with your solution, and configure it to use the upgraded SQL database.

8. Post-upgrade steps

After you have successfully upgraded your Sitecore XC containerized solution, you must complete the following tasks to complete the upgrade:

- Bootstrap the Commerce Engine
- Synchronize Control Panel content items
- Populate Solr managed schema
- Rebuild Commerce indexes
- Update the data templates
- Republish and rebuild the Sitecore search indexes
- Verify the current version number

8.1. Bootstrap the Commerce Engine

You must bootstrap the Commerce Engine to propagate the changes you made to the environment configuration files into the global database.

The Sitecore XC SDK includes code samples for DevOps operations, so that you can access the Sitecore XC API directly. The following instructions assume that you are using Postman to exercise the Sitecore XC API and that you have imported the Adventure Works and Habitat environments into Postman.

To run the bootstrap operation:

1. Launch the Postman application.

2. In the **Collections** pane in Postman, navigate to the *Authentication* folder.

3. Open the `Sitecore` folder and execute the `GetToken` request.
   When Postman displays an access token in the **Body** pane, authentication is successful.

4. In the **Collections** pane, navigate to the *SitecoreCommerce_DevOps* folder.

5. Open the *1 Environment Bootstrap* folder, and execute the `Bootstrap Sitecore Commerce` request.

6. Launch Internet Information Services (IIS) Manager and perform an IIS reset.

8.2. Synchronize Control Panel content items

As part of the upgrade process, you must synchronize content items in the Control Panel to the Commerce Engine.

If you previously synchronized Sitecore Control Panel content items to the Commerce Engine, you must re-synchronize the content.

If this is the first time Sitecore Control Panel content items are synchronized to the Commerce Engine, use this procedure instead.

Re-synchronize Control Panel content items

To re-synchronized Control Panel content item paths:

1. Launch the Postman application.

2. Navigate to the *ContentAPISamples* folder in the **Collections** pane.

3. Open the *DoActions* folder and execute the `Sync Content Path` sample for each environment that has a different data store (that is, uses a distinct dataset and configuration, identified by a different `ArtifactStoreId`).
    You can use the `Check Long Running Command Status` sample to ensure that the synchronization operation completes, as the operation may take several minutes.

    > **NOTE**
    > You may need to obtain the `EntityView` value to use as input for the Sync Content Path sample if you receive an error related the entity version. See the Synchronize content items topic for more information.

4. Launch Internet Information Service (IIS) Manager and perform an IIS restart.

Synchronize Control Panel content items for the first time

If you have not synchronized Control Panel content items before, perform the following steps:

1. Launch the Postman application.

2. Navigate to the *SitecoreCommerce_DevOps* folder in the **Collections** pane.

3. Open the *3 Environment Initialize* folder, and run the `Ensure\Sync default content paths` sample request for each environment that has a different data store (that is, for each environment that use a different `ArtifactStoreId`).

    > **NOTE**
    > The operation might take several minutes to complete. You can run the `Check Long Running Command Status` sample in Postman to ensure that the synchronization operation completes.

4. Launch Internet Information Service (IIS) Manager and perform an IIS restart.

8.3. Rebuild Commerce indexes

To rebuild the Sitecore XC indexes:

> **NOTE**
> Before you rebuild the Sitecore indexes, make sure that previous indexes were deleted.

1. Launch the Postman application

2. In the **Collections** pane, navigate to the *SitecoreCommerce_DevOps* folder.

3. Expand the *Minions* folder and execute the following requests:

   > **NOTE**
   > When you execute the indexing minion requests, make sure that they point to the URL of the Commerce Engine instance that is running the Minions service.

   - `Run FullIndex Minion - Orders`

   - `Run FullIndex Minion - Customers`

   - `Run FullIndex Minion - PriceCards`

   - `Run FullIndex Minion - Promotions`

   - `Run FullIndex Minion - Catalog Items`

8.4. Update the data templates

To update the data templates:

1. Log in to Sitecore Launchpad (https://<host>/sitecore) and click **Content Editor**.

2. Click the **Commerce** tab.

3. Click **Delete Data Templates**.

4. Click **Update Data Templates**.

8.5. Populate Solr managed schema

> **NOTE**
> This procedure is required if your deployment uses new Solr core.

If your deployment uses new Solr core, you must populate the schema in Sitecore before you rebuild the indexes.

To populate the managed schema:

1. Open a browser, and navigate to: `https://cm.globalhost/sitecore`.

2. Login to Sitecore with the *admin* user and password.

3. In the Sitecore Control Panel, in the **Indexing** tab, click **Populate Solr Managed Schema**.

4. In the **Schema Populate** dialog box, select all the indexes and then click **Populate**.

    > **NOTE**
    > Wait for the process to complete, then close the dialog box.

5. In the Sitecore Control Panel, click **Indexing manager** and, in the **Indexing Manager** dialog box, select **sitecore_web_index** and other indexes that you want to rebuild and then click **Rebuild**.

8.6. Republish and rebuild the Sitecore search indexes

To republish and rebuild the search indexes:

1. Log in to Sitecore Launchpad and click **Content Editor**.

2. Click the **Publish** tab.

3. Click **Publish** and select **Publish site** from the drop-down menu.

4. In the **Publish Site** window, select **Republish – publish everything** and click **Publish**.

5. Return to the Sitecore Launchpad and click **Control Panel**.

6. Click **Indexing manager**.

7. Select the following indexes from the Local Indexes list and click **Rebuild**:

   - `sitecore_core_index`

   - `sitecore_master_index`

   - `sitecore_web_index`

8. In IIS Manager, start the upgraded Storefront site and verify that the upgrade completed successfully.

8.7. Verify the current version number

1. Log in to Sitecore Launchpad and click **Content Editor.**

2. In the left pane, navigate to */sitecore/system/Settings/Foundation/Commerce Experience Accelerator/Upgrade/Current*.

3. Verify that the value in the **Version** field is *6000*.