Sitecore Commerce powered by Microsoft Dynamics and Sitecore Commerce powered by Commerce Server

Sitecore Commerce Deployment Guide

A whitepaper discussing how to deploy Sitecore Commerce powered by Microsoft Dynamics and Sitecore Commerce powered by Commerce Server in a Sitecore 8 environment.

Table of Contents

Chapter 1

Introduction

1.1 Target Audience

The guide is targeted at anyone who needs to deploy Sitecore Commerce powered by Microsoft Dynamics (SCpbMD or Sitecore Commerce powered by Commerce Server (SCpbCS, in environments from a single box to a multi-server environment. Typically, these people would be QA, Developers, or IT professionals.

1.2 What is Sitecore Commerce?

Sitecore Commerce is an umbrella brand for Sitecore's commerce products: Sitecore Commerce Connect, Sitecore Commerce powered by Commerce Server (SCpbCS), and Sitecore Commerce powered by Microsoft Dynamics (SCpbMD).

Sitecore Commerce Connect is an API that provides a standard way for external commerce systems (ECS) to integrate into Sitecore, and for Sitecore to talk ECSs. The API does not provide any out of the box eCommerce functionality, but ships with a suite of ecommerce related goals, page events, and engagement automation plans, which are automatically triggered by the actions being performed via the API.

Sitecore Commerce powered by Commerce Server is a Commerce Server plugin for Sitecore written on top of Sitecore Commerce Connect. When the Commerce Connect plugin is used with a data provider that exposes the Commerce Server Catalog subsystem, you have complete access to Commerce Server through Sitecore.

Sitecore Commerce powered by Microsoft Dynamics (SCpbMD) is a Microsoft Dynamics AX plugin for Sitecore that is written on top of Sitecore Commerce Connect. The SCpbMD plugin allows you to expose all products from Dynamics AX in Sitecore, and use Dynamics AX to manage and checkout user's carts.

1.3 What is the relationship between SCpbMD & SCpbCS?

SCpbMD uses Commerce Server's Catalog subsystem as an edge cache for Dynamics AX products to improve the performance of exposing the products via Sitecore. Commerce Server Catalog data is exposed in Sitecore via a data provider in SCpbCS, because of this you will always need to go through a SCpbCS installation before setting up SCpbMD.

SCpbMD is dependent only on the catalog parts of SCpbCS and Commerce Server. If you are setting up SCpbMD, you can ignore the sections of this document that reference the Order, Inventory, Profile, or Marketing subsystems of Commerce Server. All other sections are relevant to a SCpbMD installation.

1.4 Prerequisites

Before starting this installation, you should have a thorough understanding of the architecture, subsytems, and unique requirements of Sitecore, SCpbCS, and SCpmMD. By being familiar with those products, you will be able to apply the guidance in this document to suit your specific business needs. All prerequisites for installation are located in the Deployment Requirements chapter.

Chapter 2 Deploying Sitecore Commerce

2.1 How to use the deployment guide

The following table summarizes the information that is contained in this guide…

Section	Description
Deploying Sitecore Commerce	Describes the concepts, terms, and tools that you must be familiar with when you deploy Sitecore Commerce.
Deployment Architecture	Describes the logical and physical types of deployments. Use this information to plan the type of Sitecore Commerce deployments that you will install.
Deployment Requirements	Describes how to plan your network and acquire the hardware and software that is required by the servers in a Sitecore Commerce deployment.
Deployment Tools	Describes how the Sitecore Commerce Installation, Configuration, and Packaging tools work. Provides best practices for using these tools. Provides information for using these tools for unattended installation and configuration.
Building Deployments	Provides the high-level sequence of tasks for building Sitecore Commerce Server tiers, environments, and deployments. Use these topics as checklists for building out a part of your network.
Securing the Deployment	Provides information and instructions for securing a Sitecore Commerce deployment.
Securing the Deployment	Provides information and instructions for securing a Sitecore Commerce deployment.
Appendices	Helpful lists and links to help with your deployment.

2.2 Terms & Concepts

This topic provides conceptual information about the objects and entities that involved in a Sitecore Commerce install:

SCpbMD	Sitecore Commerce powered by Microsoft Dynamics AX. This is the combined Sitecore and Microsoft Dynamics AX solution.
SCpbCS	Sitecore Commerce powered by Commerce Server. This is the combined Sitecore and Commerce Server solution.
CD Server	Content Delivery Sever is a Sitecore server role, which is responsible for serving up published content from the Web database.
CM Server	Content Management Sever is a Sitecore server role, which is responsible for serving unpublished content from the Master database.
Processing Server	A server role that is part of the xDB deployment. This is used to aggregate engagement data and optionally hold user session state.
Reporting Server	A server role, which is part of the xDB deployment, used to serve up aggregated engagement data.
Master database	A Sitecore database used to store unpublished content.
Web database	A Sitecore database used to store published data.
Core database	A database shared between master and web that is used to store various settings, applications, templates, and user account data.
Config file	Files with a ".config" extension found under the \Website\App_Config\ folder of a Sitecore instance. These files contain settings and pipelines for various Sitecore and custom modules.
Config patching	The ability or override or insert settings into another Sitecore config file without manually modifying the target. More details can be found at http://sdn.sitecore.net/upload/sitecore6/60/include_file_patching_facilities_sc6orlater-a4.pdf
Commerce Server Site	Different from an IIS or Sitecore Website. A Commerce Server Site is a collection of Commerce Server resources i.e. catalog, profile, orders, and marketing, which can be used by multiple Sitecore websites.
Commerce Server PuP Package	PuP (Package UnPackage) packages are a way to package up and move Commerce Server resources and data.
Commerce Server Resources	A Commerce Server resource is another name of a Commerce Server sub system e.g. profiles, catalog, orders, or marketing.
CSS	Commerce Server Staging.
Sitecore XP	A short name for the Sitecore Experience Platform.
BizTalk (aka BTS)	BizTalk Server is Microsoft's central platform for Enterprise Application Integration (EAI).

Delivery network	The network that contains the external facing websites and roles. For example, CD role
Internal network	The network that contains the authoring environment and roles. For example, CM and Reporting
EAI	Enterprise Application Integration is the use of software and computer systems' architectural principles to integrate a set of enterprise computer applications.
ETL	Extract Transform Load. A generic term used to describe the extraction of data from a system, the result of which is transformed and then imported into another system.

2.3 Deployment roadmap

The first stage in deploying Sitecore Commerce is to build your solution on a single computer. This stage includes setting up Sitecore, SCpbCS and/or SCpbMD, and the databases under the appropriate database accounts.

After the solution is up and running on a single computer, you can deploy the solution to a test environment and ultimately to a production environment. The test and production environments typically consist of multiple servers.

Sitecore Commerce supports the following primary deployment architectures:

- Development Environment
- Pro (1xCM 2xCD)
- Pro 2 (1xDEV 2xCD/CM)
- Pro+ (1xDev 1xCM 2xCD)
- Enterprise (1xDEV/QA/CM 1xDEV/QA/CD 2xCM 2xCD)
- xDB in Azure

The following information describes the purpose for each deployment scenario. These deployments are slight modifications of the standard Sitecore topologies that have been updated to include either Commerce Server or Microsoft Dynamics AX. For a description of each server role and what Sitecore Commerce software is installed on each server, see Where should you install Sitecore Commerce components?

Formatte

Deleted: componer

Development Environment

A single server with all Sitecore, Commerce Server, SCpbCS, SCpbMD, and SQL Server software installed. If you are working with Dynamics AX you will possibly need a second machine to install all of the Dynamics AX software, as Dynamics AX requires more resources than are available with most machines.

Pro

Pro configurations are intended for medium traffic sites with concerns about site availability. There are two redundant production content delivery servers, with developers and authors working in a single content management server located in the Internal network. However, because developers and authors are working in the same environment, chances are strong that developer activity could frequently bring down the site, stopping authors from creating content. This topology is for conceptual purposes only. Please contact your Sitecore Representative for actual production requirements and licensing details.

Pro 2

Pro 2 is similar to the Pro configuration, with the difference of the production servers are used for content delivery and management, while the server inside the Internal network is dedicated to developers. The issues with this configuration are that the development server will not be up to date with the latest content, so updates cannot be tested against the latest content. This situation could lead to production issues. This topology is for conceptual purposes only. Please contact your Sitecore Representative for actual production requirements and licensing details.

Pro +

Pro+ is the lowest risk deployment of the Pro series. Pro+ matches the Pro configuration, but adds a new server for developer integration. This configuration gives developers a chance to test integrations without the risk of disturbing content authors. After thorough testing, developers can deploy the updates to the content management server. The downtime for authors should be low, and the full content and code updates can be tested before being applied to production. This topology is for

production usage scenarios. Please contact your Sitecore Representative for any questions concerning licensing details.

Enterprise

For large sites with high traffic, a lot of xDB usage, and demand for redundancy in all areas, this is the role to start with. With this role, all Sitecore XP and Sitecore Commerce roles have been broken out on separate servers, and are in a position to be scaled horizontally or vertically with little effort. This topology is for production usage scenarios. Please contact your Sitecore Representative for any questions concerning licensing details.

xDB in Azure

One option for all topologies is to move the xDB part of the topology to the Sitecore hosted xDB solution in Azure. This change to all topologies will see the Reporting Service, Processing Service, MongoDB, Reporting database, and indexed xDB content, all removed from the local install. An advantage of using xDB in Azure is that additional training to familiarize people with MongoDB is unnecessary. Another advantage is the environment is built to support scaling out. For more information about xDB in Azure, go to: https://doc.sitecore.net/Products/Sitecore%20Experience%20Platform/xDB%20configuration/Cloud%20configuration%20options

2.4 How software is installed within a deployment

A Commerce Server deployment requires that you install, configure, and unpack various Sitecore Commerce components. You use the Sitecore, Commerce Server and Dynamics AX tools throughout the deployment.

2.4.1 What are the Server Roles in a Sitecore Commerce Deployment

Server Role	Description
Sitecore	
Content Management Server	A Sitecore instance used for authoring content before publishing it to production. For example, a master database.
Content Delivery Server	A Sitecore instance used for serving up published content. For example, a web database.
Search Server	A server used to host search queries. This is only used with Solr or other search solutions that are not hosted on the same server as Sitecore.
Processing Server	Performs reducing functions on experience data, which then stores the data in the reporting database.
Reporting Service	Handles queries against the Sitecore xDB aggregated data.
Commerce Server	
Business Management server	A set of web services used by BizTalk and the desktop business tools for managing Commerce Server sub system data. For example, catalog, profiles, orders, and marketing
Commerce Server Staging	A Windows service for moving Commerce Server subsystem data between environments.
SCpbMD	
Transaction Service	A service used to talk to Dynamics AX for cart, order, price, and profile data.
Routing Service	Used to map data from the Dynamics AX Channel database into the Commerce Server Catalog subsystem
Microsoft Dynamics AX	
Dynamics AX AOS Server	The Microsoft Dynamics AX Application Object Server (AOS) shares application objects and information to increase application performance.
General	

Server Role	Description
SQL Server	Used to store Sitecore and Commerce Server databases.
MongoDb	Used to store Sitecore xDB data.
BizTalk	Used for EAI to ETL data between Commerce Server and LOB systems.

2.4.2 Which Sitecore Commerce components should I install?

The following table describes the server roles used in the different environments supported by Sitecore Commerce deployments. For information about the environments and deployment types, see Deployment Roadmap.

Service	Description
Sitecore	
Sitecore XP	A complete installation of Sitecore is required on all CM, CD, Processing, and Reporting Servers.
Solr support package	The package enables Solr as the search provider instead of Lucene. If you are using Solr, this is required on all Sitecore servers.
Commerce Server	
Commerce Server Core	Includes the Commerce Server Catalog, Inventory, Marketing, Orders, and Profiles Systems, Commerce Server Manager, product documentation, Commerce Server client tools, and the SDK. Install the Commerce Server core components on the Business Management server, Content Management server, and Content Delivery server.
Commerce Server Staging Services	Includes the Commerce Server Staging (CSS) system components, including the deployment engine and CSS service. Use CSS to stage your business data from one environment to another. For example, from a test environment to the production environment. Install this component on the Business Management server and Web servers in the deployment.

Service	Description
Business Management applications	Supports business users, customer service representatives, developers, and administrators who make changes to the catalog, profile, orders, and marketing data. There are four Business Management applications: • Catalog Manager. Supports product catalog Web site management activities. Sitecore Commerce Merchandising Manager is another application to manage Catalog data. • Catalog and Inventory Schema Manager. Supports managing catalog definitions, categories, products, and properties. • Customer and Orders Manager. Supports order management Web site activities. • Marketing Manager. Supports discount, advertisement, and direct mail Web site management activities.
Commerce Server adapters for BizTalk Server	(Optional) Commerce Server adapters for BizTalk Server to help integrate Commerce Server with other applications, Web services, and trading partners. Install this component on computers that are running BizTalk Server.
Sitecore Commerce Connect	Sitecore's Commerce integration framework that enables all eCommerce engines to connect into Sitecore in the same way, and standardize all of the page events, goals, and EAPs used by each. Both SCpbCS and SCpbMD build on top of Commerce Connect, so this needs to be installed on all servers used by either product. For example, CM, CD, and Processing servers
SCpbCS	An extension to Sitecore Commerce Connect that exposes Commerce Server data to Sitecore. SCpbCS is made up of two packages, one package for the API level integration, and another for the Merchandising Manager tool. This Commerce Server Connect package is required on CM, CD, and processing servers.
SCpbMD	
All packages	An extension to Sitecore Commerce Connect that exposes Microsoft Dynamics AX to Sitecore. This package is required on CM, CD, and processing servers.
Transaction Service	The runtime communication layer for that allows the Sitecore site to manage user and order data in Microsoft Dynamics AX. This service can be used with the Sitecore CM or CD servers, or as a standalone IIS service.
Routing service	Used to transfer catalog data from the Dynamics AX Channel database into Commerce Server Catalog. This service can run inside of Sitecore as a job, or externally as a console app or Windows service.

Service	Description
Microsoft Dynamics AX AOS Server	This should be installed on its own server in the internal network.

The following table describes the all software components and the Sitecore Commerce tools that install them.

Component	Tool
Commerce Server global resource databases: • MSCS_Admin • MSCS_CatalogScratch	Commerce Server Configuration wizard
Commerce Server site-specific databases: • <site_name>_productcatalog • <site_name>_marketing • <site_name>_marketing_lists • <site_name>_profiles • <site_name>_transactionconfig • <site_name>_transactions	Site Packager
Commerce Server site and resources	Site Packager
Commerce Server Web services	Site Packager
IIS Default Web Site Widgets directory	Commerce Server Configuration wizard
COM Components	Commerce Server Configuration wizard

2.4.3 Where should you install Sitecore Commerce components?

The following table shows the Sitecore Commerce software components that you typically install to support each server role.

Server Role	Commerce Server Core	Commerce Server Staging	Business Management Applications	Commerce Server Adapters for BizTalk Server
Sitecore				
CM Server	Yes	Optional		
CD Server	Yes			
Search Server				
Processing Server				
Reporting Service				
Commerce Server				
Business Management server	Yes	Yes	Optional	
SCpbMD				

Sitecore Commerce powered by Commerce Server & Microsoft Dynamics

Server Role	Commerce Server Core	Commerce Server Staging	Business Management Applications	Commerce Server Adapters for BizTalk Server
Transaction Service				
Routing Service	Yes			
Microsoft Dynamics AX				
Dynamics AX AOS Server				
General				
SQL Server				
MongoDb				
BizTalk				Yes

Server Role	Sitecore XP	Sitecore Commerce Connect	SCpbCS	SCpbMD	Dynamics AX AOS
Sitecore					
CM Server	Yes	Yes	Yes	Yes	
CD Server	Yes	Yes	Yes	Yes	
Processing Server	Yes	Yes	Yes	Yes	
Reporting Service	Yes				
Commerce Server					
Business Management server					
SCpbMD					
Transaction Service				Yes	
Routing Service				Yes	
Microsoft Dynamics AX					
Dynamics AX AOS Server					Yes

Server Role	Sitecore XP	Sitecore Commerce Connect	SCpbCS	SCpbMD	Dynamics AX AOS
General					
SQL Server					
MongoDb					
BizTalk					

2.4.4 What databases are required in what environment?

A Sitecore Commerce solution can be deployed on internal and delivery networks. If you require a more secure deployment, the external databases can run on the internal network. However, this could cause performance issues. Be aware of the following:

- Sitecore XP
  - Create a Web database in the internal environment, to allow you to preview content. This web database also means that the Web search indexes can be created ahead of time and copied to the production environment. By doing this, there is no delay in waiting for the “new” web data to index in the production environment.
- Commerce Server
  - Do not use the same Commerce Server site for Content Management and Content Delivery. Instead, create two Commerce Server sites and use Commerce Server Staging (CSS) to move data between the two. Using the same site for Content Management and Content Delivery will cause any changes to catalog data on the Content Management site to be immediately published live to the Content Delivery site.

The databases that are required for a Sitecore Commerce Solution are listed in the following table:

	Internal	**Delivery**
Sitecore		
master	Yes	
core	Yes	Yes
web	Yes	Yes
Commerce Server		
All	Yes	Yes
Microsoft Dynamics AX		
AOS Databases	Yes	
Channel Database		Yes

2.4.5 How do you install Sitecore Commerce components?

Sitecore Commerce creates different databases at different points of the deployment. The following table summarizes how Sitecore Commerce is deployed across servers in a basic or enterprise deployment:

Server Role	Software	Tool
Sitecore		
CM Server	Sitecore XP	Sitecore zip or .exe plus some manual configuration
	Sitecore Commerce Connect	Sitecore package installer
	Commerce Server Connect	Sitecore update package installer
	Merchandising Manager	Sitecore update package installer
	Sitecore.Commerce.DynamicsIntegration.Connect	Sitecore update package installer
	Sitecore.Commerce.DynamicsIntegration.Routing.update	Sitecore update package installer
	Sitecore.Commerce.RoutingFramework	Sitecore update package installer
	Sitecore.Commerce.DynamicsIntegration.TransactionService	Sitecore update package installer
	Solr support package	Zip with .dlls to be copied to bin
CD Server	Sitecore XP	Sitecore zip or .exe plus some manual configuration
	Sitecore Commerce Connect	Sitecore package installer
	Commerce Server Connect	Sitecore update package installer
	Sitecore.Commerce.DynamicsIntegration.Connect	Sitecore update package installer
	Sitecore.Commerce.DynamicsIntegration.TransactionService.update	Sitecore update package installer
	Solr support package	Zip with .dlls to be copied to bin
Processing Server	Sitecore XP	Sitecore zip or .exe plus some manual configuration
	Sitecore Commerce Connect	Sitecore package installer
	Commerce Server Connect	Sitecore update package installer
	Solr support package	Zip with .dlls to be copied to bin
Reporting Service	Sitecore XP	Sitecore zip or .exe plus some manual configuration
	Solr support package	Zip with .dlls to be copied to bin

Server Role	Software	Tool
Commerce Server		
Business Management server	Commerce Server Core	Commerce Server Installer and Configuration wizard
	Commerce Server site and Web services	Commerce Server Site Packager
	Commerce Server Staging	Commerce Server Installer and Configuration wizard
Business Management Application clients	Business Management Application clients	Commerce Server Business Tools Installer
SCpbMD		
Transaction Service	TransactionService SDK	Zip file with solution to create custom service.
Routing Service	DynamicsRetail.Routing SDK	Zip file with solution to create custom service.
	Routing.Framework SDK	Zip file with solution to create custom service.
Microsoft Dynamics AX		
Dynamics AX AOS Server		Exe available from Microsoft
General		
SQL Server		Exe available from Microsoft
MongoDb		Exe available from MongoDB site
BizTalk	Commerce Server adapters for BizTalk Server	Commerce Server BizTalk Adapters Installer

Chapter 3 Deployment Architecture

This section provides information about the logical and physical designs for the recommended deployment architectures in Sitecore Commerce.

These architectures do not support a specific site, application, or scenario. They provide a reference that you can modify to meet a variety of business needs. Your actual deployment design will vary according to your resources and requirements.

Sitecore Commerce provides several ways for you to scale a deployment. You can isolate and extend each functional area onto separate clusters/servers.

3.1 What is the logical design of a deployment?

Understanding the logical design of a deployment is the first step in transforming your functional requirements and components into a physical design that you can deploy in your data center.

Conceptually, you can categorize the supported features in your deployment according to their functionality and relation to site data, while physically you can deploy these features in separate environments.

The recommended deployment scenario categorizes features into three functional design components: Development & QA, Content Management, and Content Delivery.

The following figure shows the three functional design components in Sitecore Commerce.

3.1.1 Development & QA

This tier is used for developers creating new functionality, QA testing the functionality, and the approved functionality being bundled in Sitecore packages and installed in the Content Management environment. In this environment, developers typically have all software installed on one machine, which makes the deployment and testing of solutions simpler and quicker. QA servers in this environment are typically of larger scale, so the solutions created are tested in a more realistic environment.

3.1.2 Content Management

In this tier, content is created for the production website, and can be viewed and updated without being reflected on the public site. This environment is also used by content creators for acceptance testing of functionality created by developers. Content is pushed from this environment to the Content Delivery environment by Sitecore Publishing and/or Commerce Server Staging. Any packages created by developers will be manually installed on the delivery network servers.

3.1.3 Content Delivery

In this tier, content is delivered to the public by publishing it from the Content Management environment. Content should never be edited directly in this environment. Multiple CD servers may be needed to handle load against this role, and some of the additional Siteore roles may need to be turned off and moved onto different servers to avoid overconsumption of resources, such as CPU, memory, or disk.

3.2 What is the physical design of a deployment?

This section provides information about the physical design of the recommended deployment scenarios. It presents the network topology of the deployment and describes the servers in the development, Content Management, and Content Delivery tiers.

In the recommended deployment scenario, the production environment consists of three tiers. Internal business users belong to a separate domain, the corporate network, and external site visitors access the deployment over the Internet.

You can create four separate network segments for the following tiers:

- Data tier in the run-time environment.
- Web tier in the run-time environment.
- Staging tier in the design-time environment.
- Internal development/test/business management tier in the design-time environment.

The internal development/test/business management tier has a separate domain. Set up a domain trust with the staging tier in order to stage content from the test tier to the staging tier.

The data tier is a collection of clustered computers that are running SQL Server, MongoDB, a business management server, and an Active Directory service domain controller. A firewall helps protect these servers from access from the Web tier, in the chance that a malicious user manages to compromise the Web tier. Another firewall protects the database servers against access from the internal development/test/business management tier to prevent internal users from accidentally overwriting run-time data and disrupting run-time operations.

The Web tier consists of two Web servers and an Active Directory domain controller. A firewall/load balancer helps protect the deployment from malicious users on the Internet, and a firewall protects the Active Directory domain controller and the data tier from malicious users who manage to compromise the Web tier.

The staging tier is a computer that is running SQL Server and a business management server. This tier belongs to the same domain as the database servers. This tier stages data from the internal development/test/business management tier to the run-time environment. The business management server in this tier runs the Catalog and Inventory Web service, the Marketing Web service, and the Orders Web service.

The internal development/test/business management tier is a computer that is running SQL Server, MongoDB, a business management server, and an Active Directory domain controller. In this tier you can conduct pre-production development and testing. Business users can connect to the data-tier business management server, and to the Web services that are running on the staging server from this tier.

3.3 Can I create topologies other than the ones listed?

Yes. The topologies in the following chapters are examples, and exist to help you identify the topology that will most closely match your traffic and workflow needs. You can adjust the topology for security, load, additional applications, and any other concerns. Once you choose your topology, there are still questions you need to answer for your solution. For example:

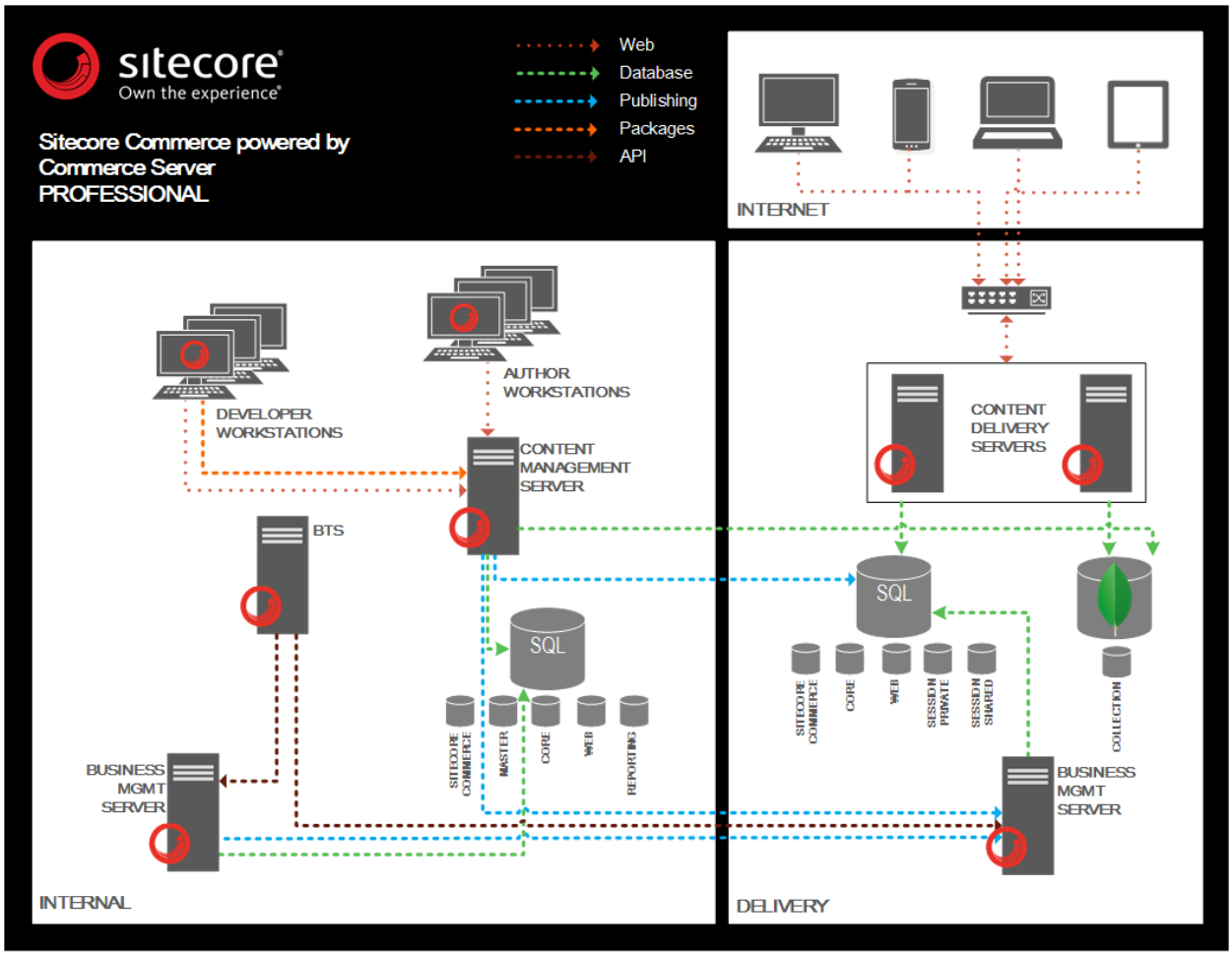
- Where should my domain controllers go?
- Do I want a different domain for my internal and delivery networks?
- Will there be underutilized Sitecore Commerce roles? If so, is there justification for putting them on separate servers?
- Will some specific Sitecore Commerce roles need to be scaled out horizontally or vertically?
- Do I want my internal and delivery networks to be merged?
- Do I want a SQL Server instance in my delivery network, or should I bring it into the internal network?
- Is it a better choice to manage MongoDB myself, or use the Sitecore xDB Cloud service?

These are not the only questions, but a starting list of things you should be aware of while reviewing topologies and how Sitecore Commerce roles will scale for your needs.

3.4 What is a Professional SCpbCS Deployment?

The Professional deployment is the most basic Sitecore Commerce deployment that provides load balancing and redundancy on the Content Delivery servers. This topology may have issues if there is high xDB activity, because the processing service may start to consume too much CPU as it reduces gathered data. If the load on the content delivery servers becomes an issue, you could always scale horizontally with additional servers. The reporting role is housed on the CM server, and each CD server has the processing service activated.

This topology is for conceptual purposes only. Please contact your Sitecore Representative for actual production requirements and licensing details.

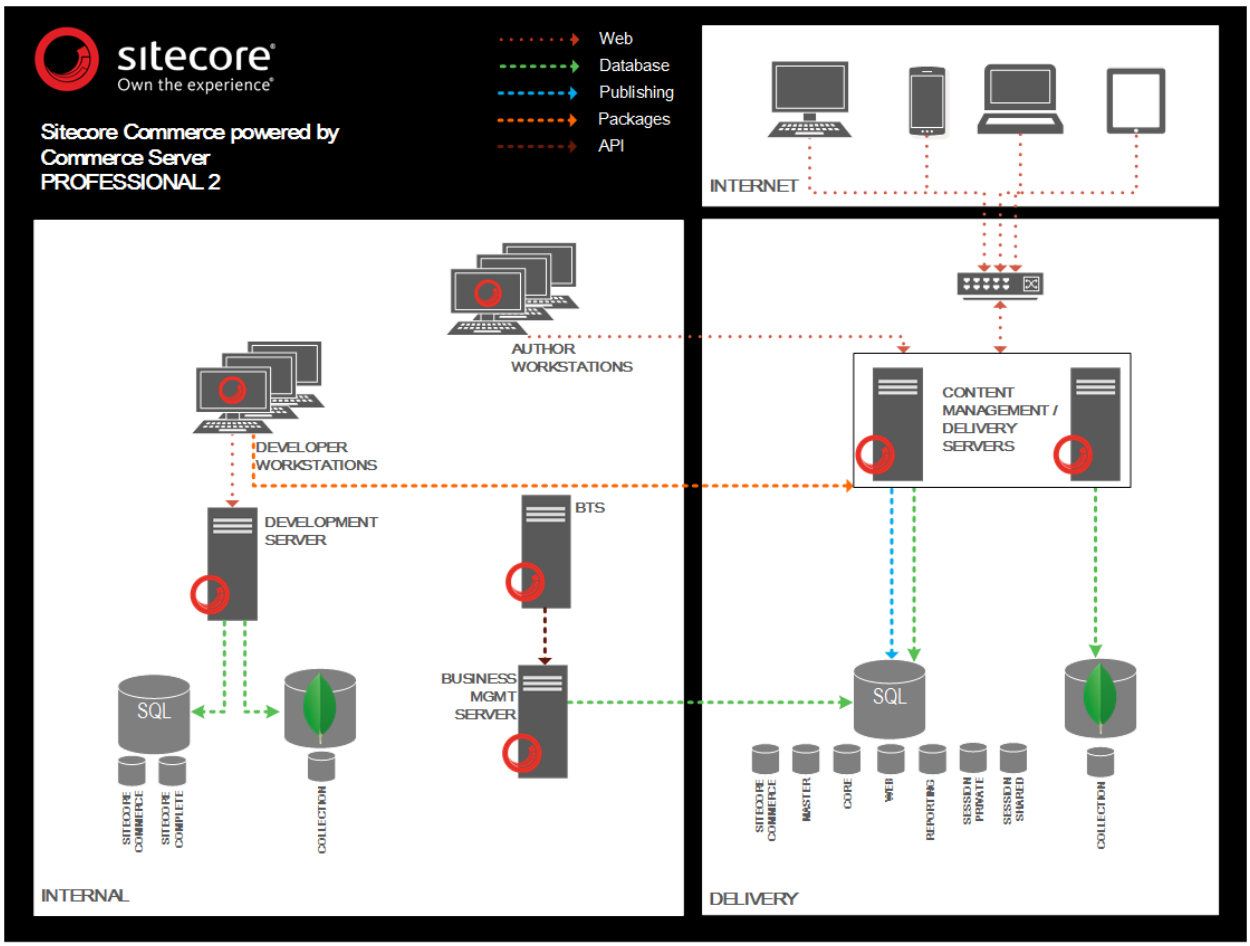


3.5 What is a Professional 2 SCpbCS Deployment?

The Professional 2 deployment removes the independent CM server, and merges it with the CD servers in the delivery network. If you have content that rarely changes, this role may meet your needs. There are risks with this topology, including that there is no environment to test changes, and if you are only using one Commerce Server site, any changes made in master site will be immediately available in the web site. There are also potential security risks to the delivery network, because you need to allow access to a greater number of people for content modification.

Much like the Professional topology, the Professional 2 deployment may have issues if there is high xDB activity, because the processing service may start to consume too much CPU as it reduces gathered data. If the load on the content delivery servers becomes an issue, you could always scale horizontally with additional servers. The reporting role is housed on the CM server, and each CD server has the processing service activated.

This topology is for conceptual purposes only. Please contact your Sitecore Representative for actual production requirements and licensing details.



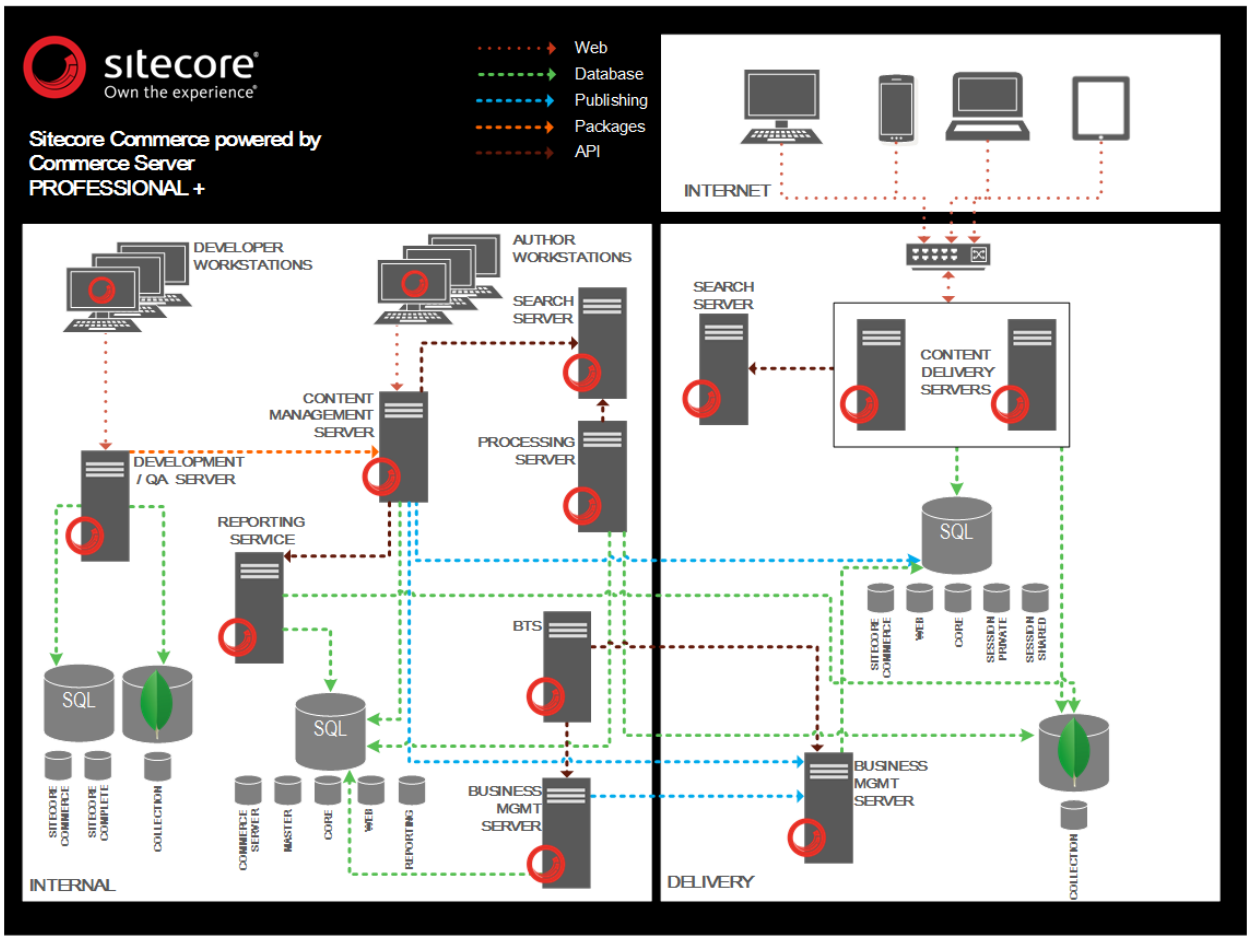
3.6 What is a Professional+ SCpbCS Deployment?

The Professional+ deployment is set up for environments with higher load. All roles have been broken out onto their own servers, leaving the CD server with more resources available for serving content. All roles on separate servers means that if any servers experiences an increase in load, you have the ability to easily scale horizontally or vertically.

This deployment also includes a separate search role, to be used with a search product that is hosted on a different server, such as Solr or Coveo. It is strongly recommended that you use Solr in all production scenarios, and try to use Lucene for development only. Having a separate search role and not using Lucene means you no longer need to worry about indexes needing to be copied to each CD server, or indexing on each CD server and the disk/CPU becoming completely consumed during indexing.

With larger traffic sites, xDB may have a high level of engagement activity that needs to be condensed into the Reporting database. This processing is performed by the Processing role and uses a large amount of CPU resources, so you may want to place the Processing role on a separate server.

This topology is for production usage scenarios. Please contact your Sitecore Representative for any questions concerning licensing details.



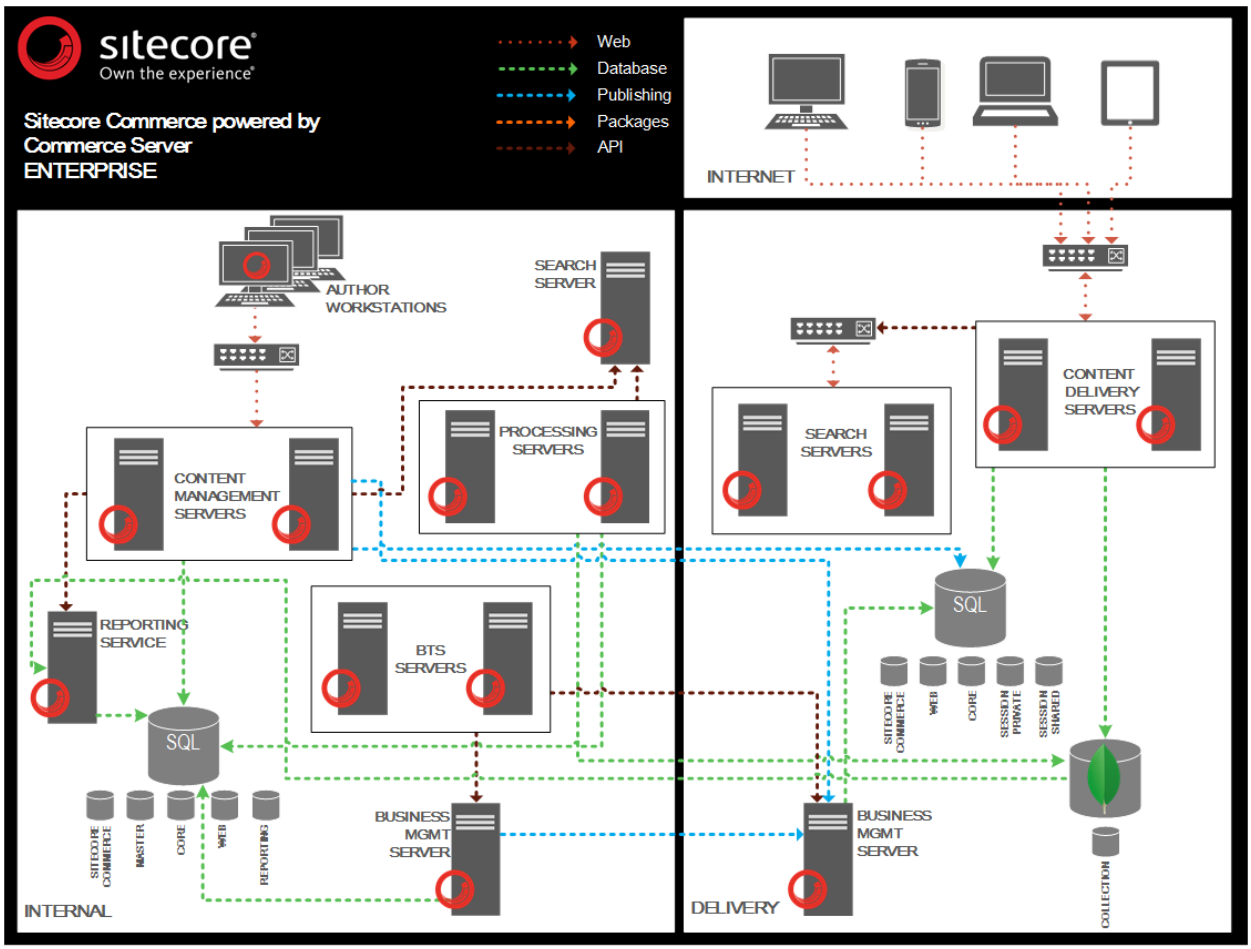
3.7 What is an Enterprise SCpbCS Deployment?

The Enterprise deployment builds on top of the Professional+ deployment by starting to scale out all of the services horizontally. This deployment is the base to for all sites with heavy load requirements. The load-balanced CM servers are used more for redundancy than load, and help ensure that the CM environment will always be available for content authors. Processing servers work by reading from an xDB queue of unprocessed data. Additional processing servers means a faster turn-around with the queue.

Be aware that the following image shows two servers in each grouping, but you can scale outwards with more servers as your demand increases.

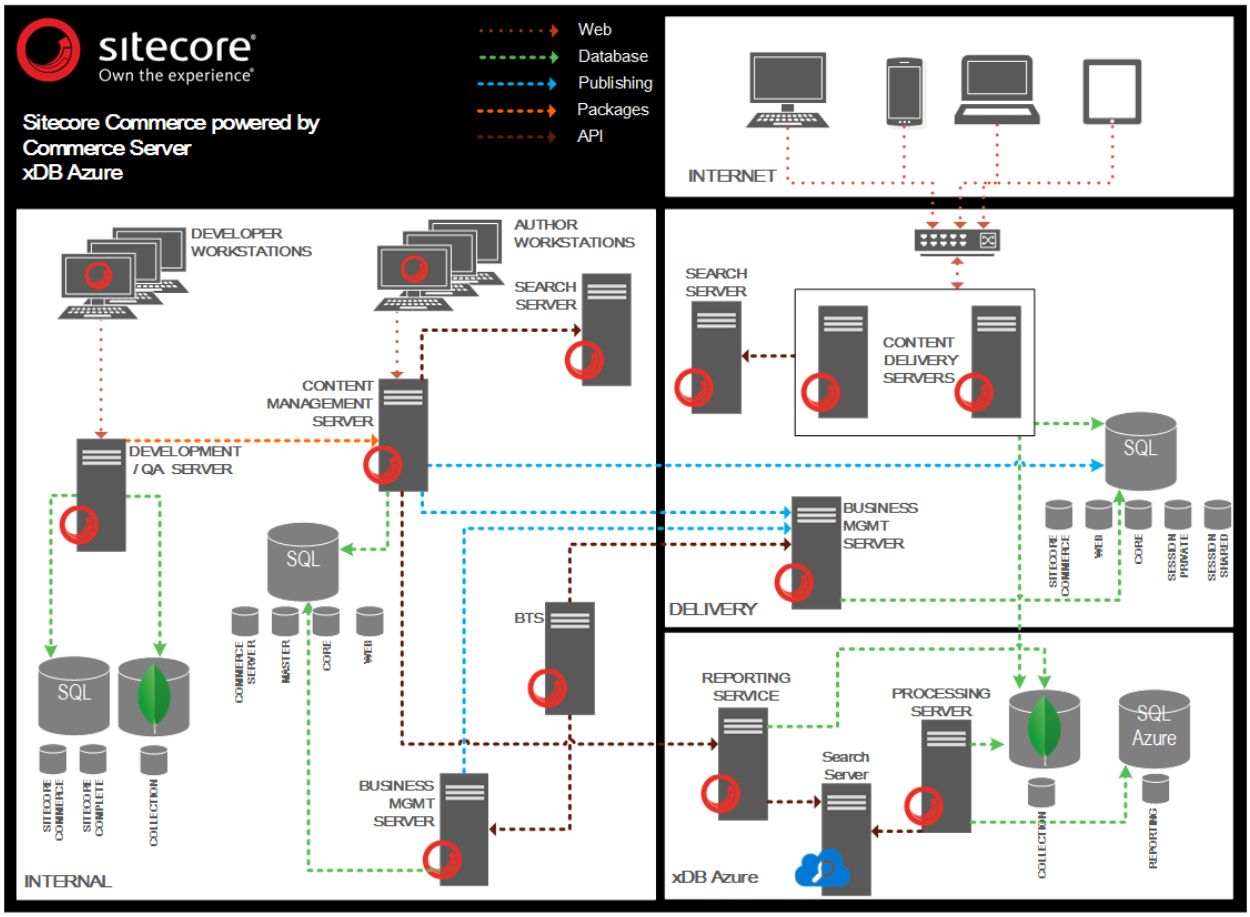
Having multiple BizTalk servers introduces redundancy, to ensure that important transactions, such as the transportation of new orders, never halts. Additional BTS servers also means that some servers can focus on frequent high processing tasks, and leave other servers for infrequent, low-processing tasks such as catalog imports.

This topology is for production usage scenarios. Please contact your Sitecore Representative for any questions concerning licensing details.



3.8 SCpbCS Deployment with xDB Cloud

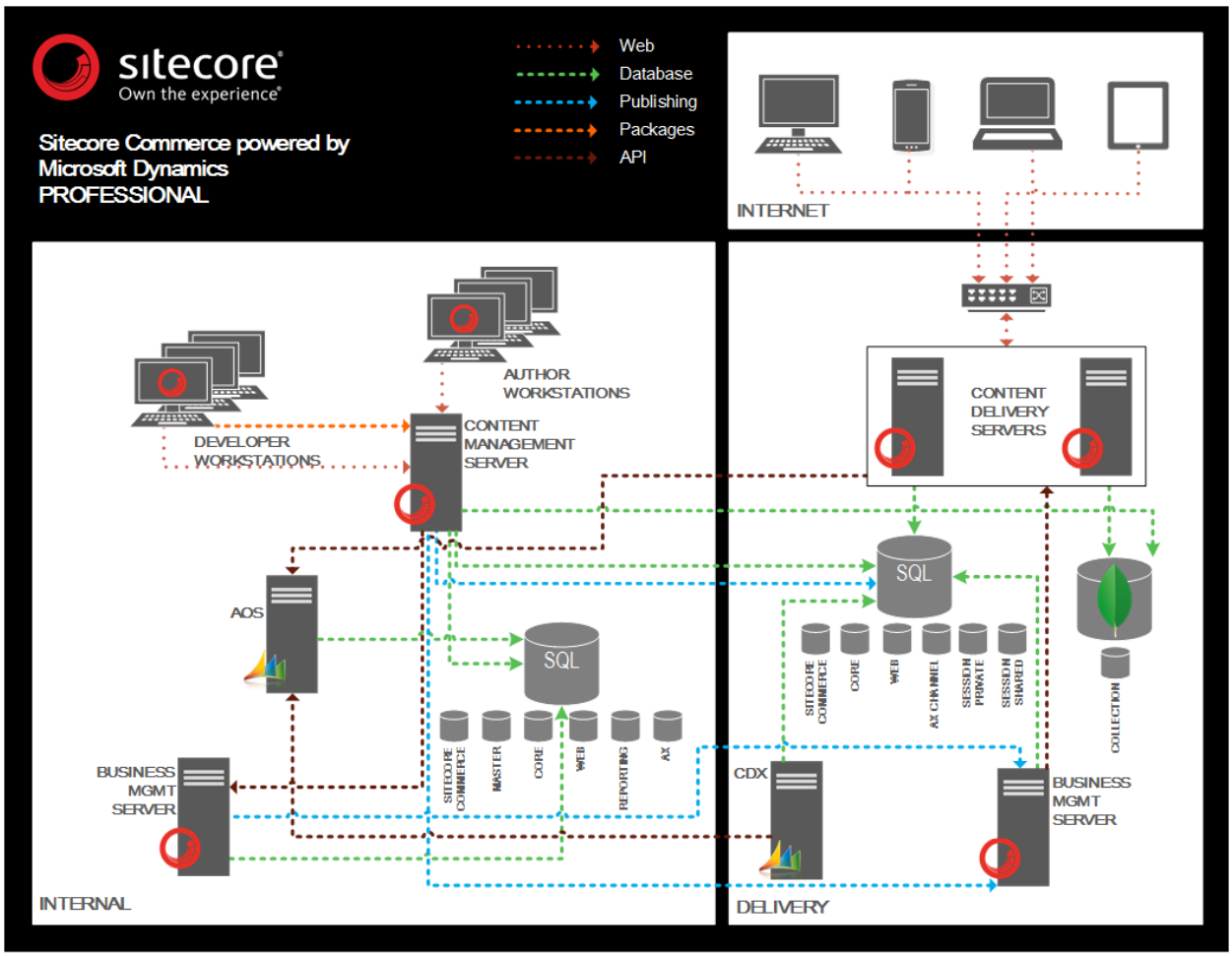
If you would prefer not to deal with local management of MongoDB, you have the option of using the Sitecore xDB Cloud offering, which will host all of xDB in Azure and you just need to connect your local topology up to it. The sample topology shown in the following image takes the Professional+ deployment and connects it with xDB Cloud, but you can connect any topology.



3.9 What is a Professional SCpbMD Deployment?

The Professional deployment is the most basic Sitecore Commerce deployment that gives you some load balancing and redundancy on the Content Delivery servers. This topology may have issues if there is high xDB activity, because the processing service may start to consume too much CPU as it reduces gathered data. If the load on the content delivery servers becomes an issue, you could always scale horizontally with additional servers. The reporting role is housed on the CM server, and each CD server has the processing service activated.

With this deployment, the routing service should be hosted on the Business Management Server in the internal network. The Business Management Server is a very low-activity server, and is ideal place to add low-activity roles. There is an instance of the Transaction service housed on each CD server, with each Sitecore site talking to its own local instance.

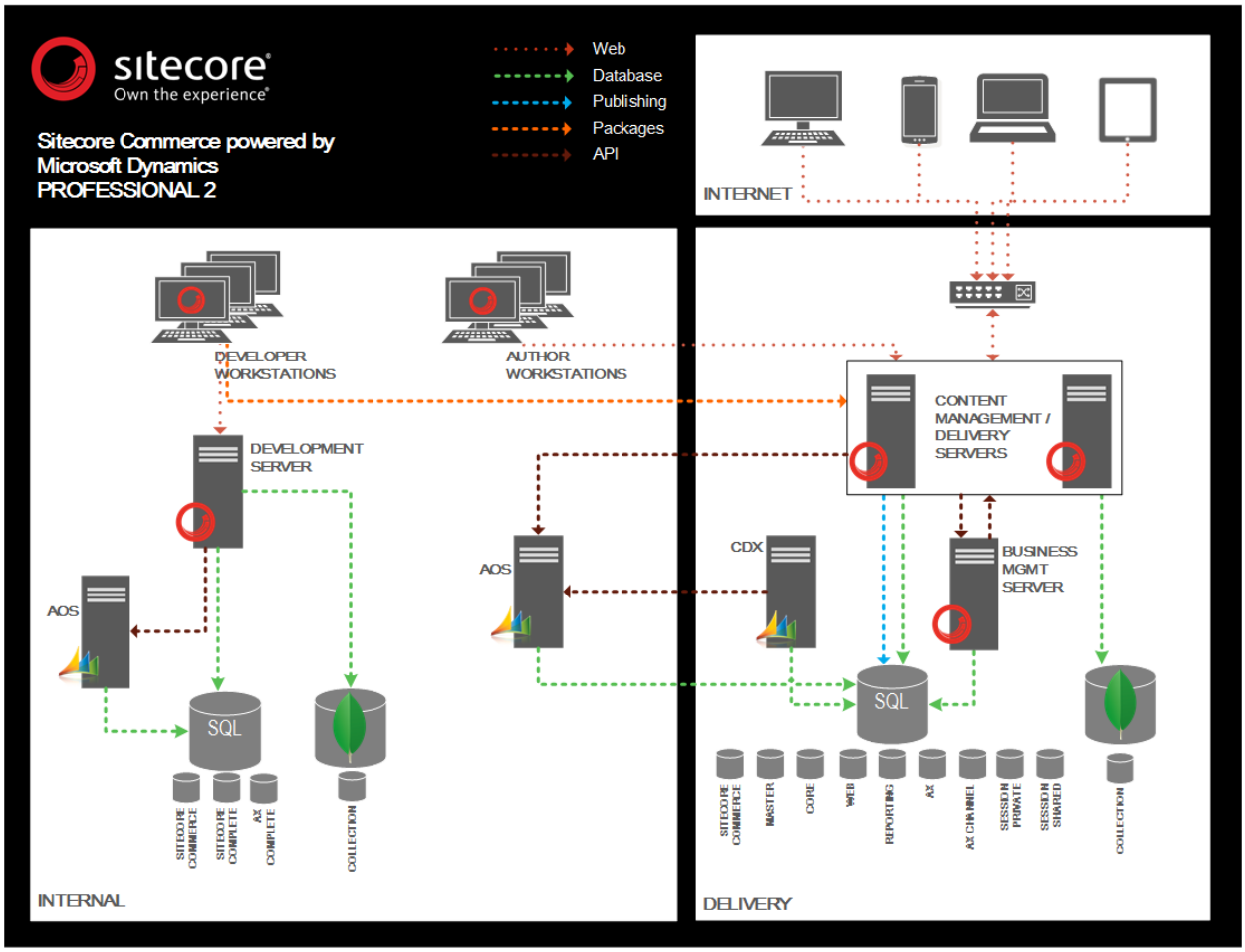


3.10 What is a Professional 2 SCpbMD Deployment?

The Professional 2 deployment removes the independent CM server, and merges it with the CD servers in the delivery network. If you have content that rarely changes, this role may meet your needs. There are risks with this topology, including that there is no environment to test changes, and if you are only using one Commerce Server site, any changes made in master site will be immediately available in the web site. There are also potential security risks to the delivery network, because you need to allow access to a greater number of people for content modification.

Much like the Professional topology, the Professional 2 deployment may have issues if there is high xDB activity, because the processing service may start to consume too much CPU as it reduces gathered data. If the load on the content delivery servers becomes an issue, you could always scale horizontally with additional servers. The the reporting role is housed on the CM server, and each CD server has the processing service activated.

As with the Professional deployment, the routing service should be hosted on the Business Management Server but this time in the Delivery network. There is an instance of the Transaction service housed on each CD server, with each Sitecore site talking to its own local instance.



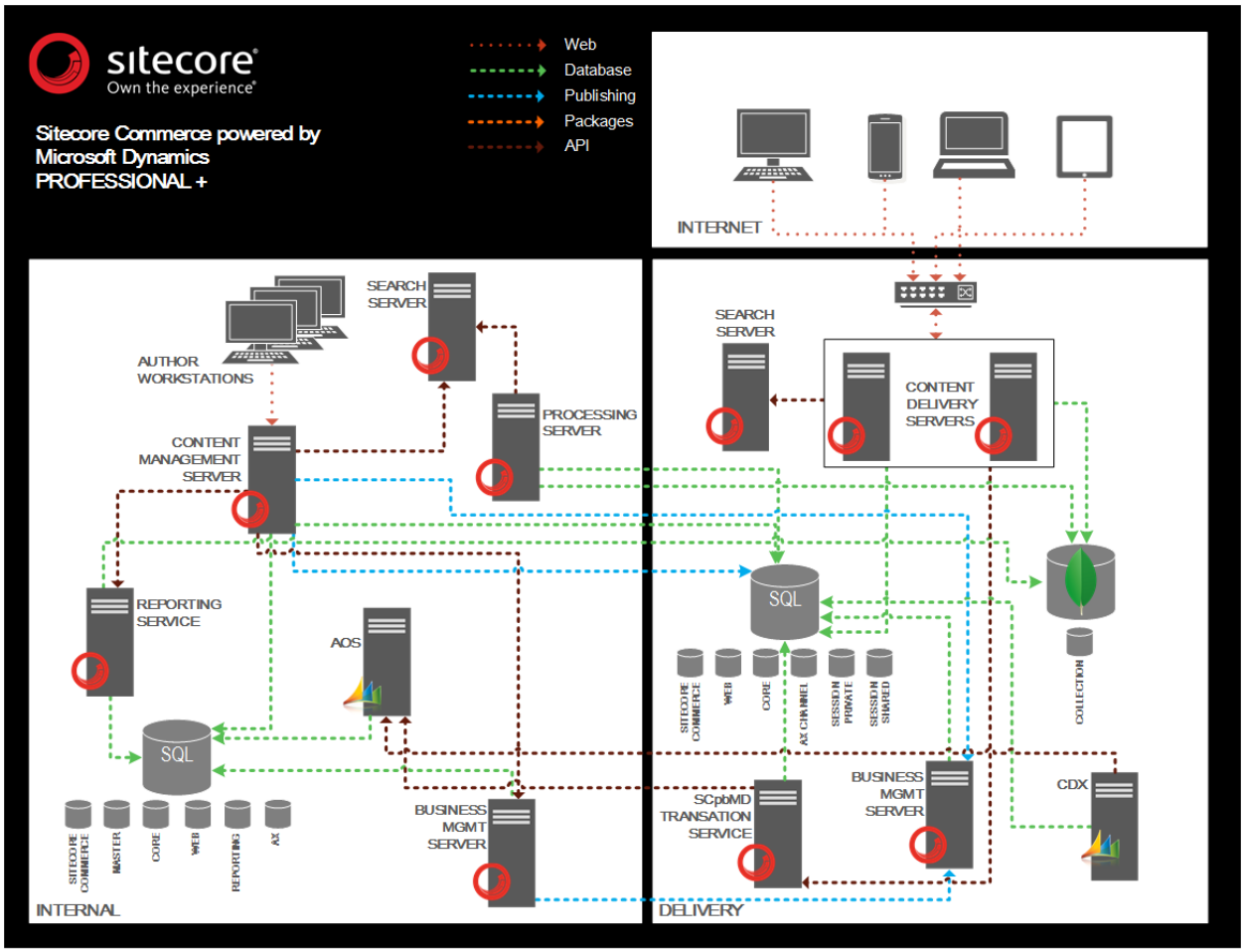
3.11 What is a Professional+ SCpbMD Deployment?

The Professional+ deployment is set up for environments with higher load. All roles have been broken out onto their own servers, leaving the CD server with more resources available for serving content. Having all roles on separate servers means that if any servers experiences an increase in load, you have the ability to easily scale horizontally or vertically.

This deployment also includes a separate search role, to be used with a search product like Solr or Coveo that is hosted on a separate server. It is strongly recommended that you use Solr in all production scenarios, and try to use Lucene for development only. Having a separate search role instead of using Lucene means you no longer need to worry about indexes needing to be copied to each CD server, or indexing on each CD server and the disk/CPU becoming completely consumed during indexing.

With larger traffic sites, xDB may have a high level of engagement activity that needs to be condensed into the Reporting database. This processing is performed by the Processing role and uses a large amount of CPU resources, so you may want to place the Processing role on a separate server.

With this topology, the Routing Service must be put on the same server as the Content Management server, but the Transaction service has been put on its own server. If you have a lot of traffic you will want to leave the Transaction service on its own server so that the CD servers can focus solely on serving content, and you can dedicate a full set of CPUs to work with Dynamics AX.

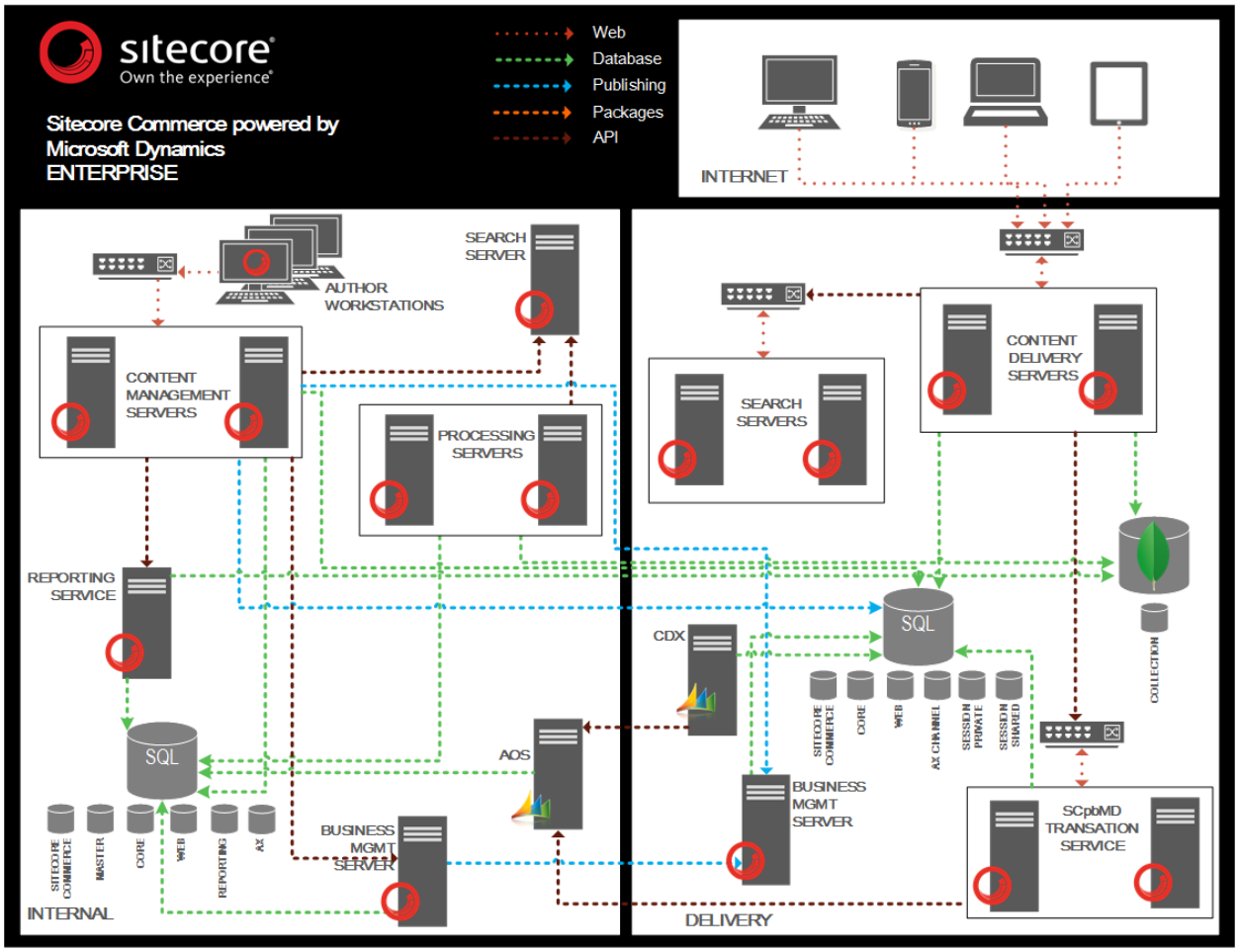


3.12 What is an Enterprise SCpbMD Deployment?

The Enterprise deployment builds on top of the Professional+ deployment by starting to scale out all of the services horizontally. This deployment is the base to use for all sites with heavy load requirements. The load-balanced CM servers are used more for redundancy than load, and help ensure that the CM environment will always be available for content authors. Processing servers work by reading from an xDB queue of unprocessed data. Additional processing servers means a faster turn-around with the queue.

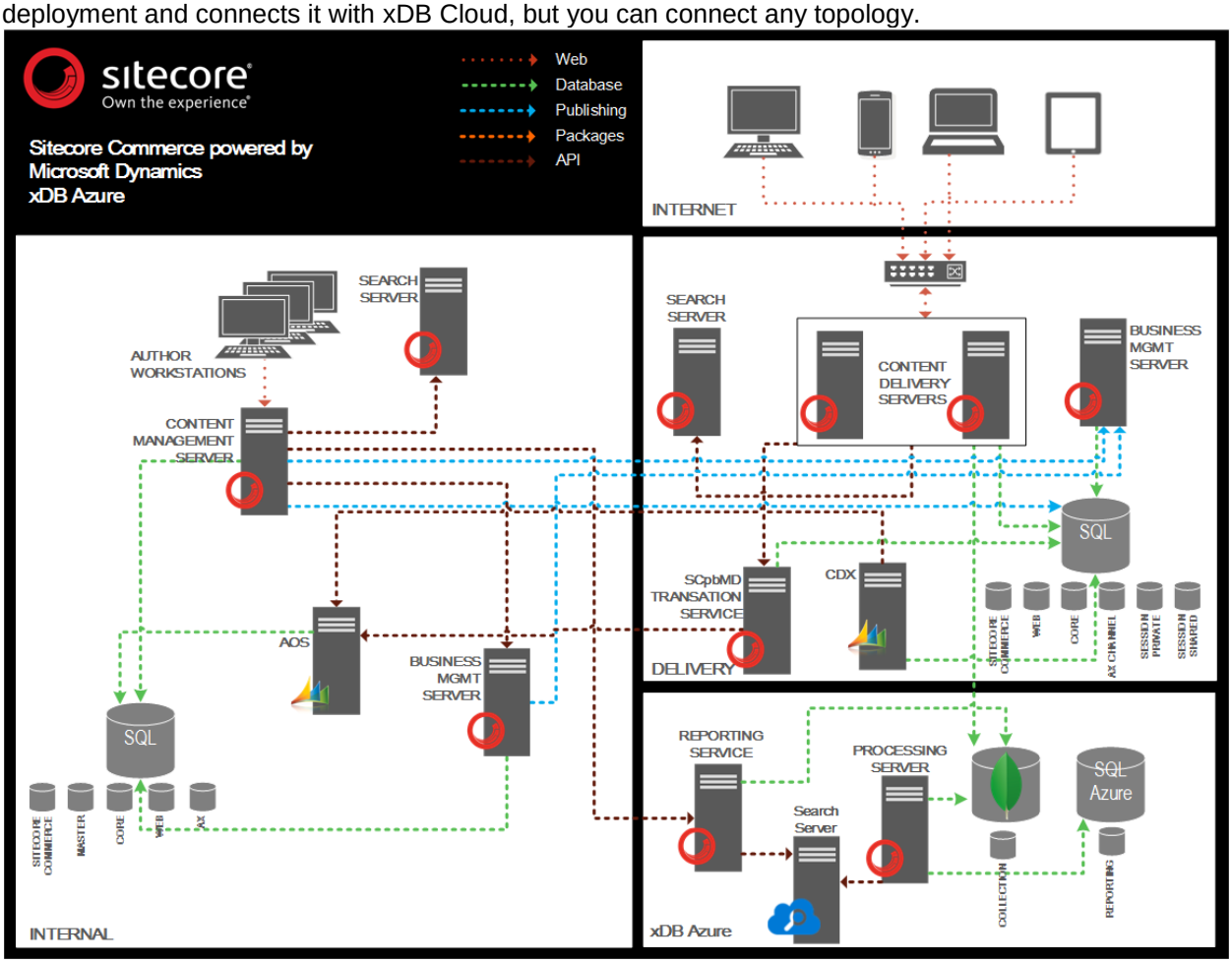
Be aware that the following image shows two servers in each grouping, but you can scale outwards with more servers as your demand increases.

With the increased load, there may be increased requests the AOS server. You may want to increase the number of Transaction Servers to handle the extra traffic and provide redundancy in case a server fails.



3.13 SCpbMD Deployment with xDB Cloud

If you would prefer not to deal with local management of MongoDB, you have the option of using the Sitecore xDB Cloud offering, which will host all of xDB in Azure and you just need to connect your local topology up to it. The sample topology shown in the following image takes the Professional+ deployment and connects it with xDB Cloud, but you can connect any topology.



Chapter 4 Deployment Requirements

This section lists the general requirements for Sitecore Commerce deployments. The requirements for your actual deployment may vary according to the needs of your environment.

4.1 What are the network deployment requirements?

When designing the network layout for your production environment, you should consider separating your Content Delivery and Content Management environments by having them in separate subnets, and using a firewall to control port and machine access. Separating the environments creates a more secure deployment, but does increase the complexity of the deployment, including the need to recreate some of the roles in both environments. In Azure, you can achieve this by using a Cloud Only Virtual Network and Windows Firewall. Azure does not offer the equivalent of a firewall appliance, but Barracuda Networks does offer a Virtual Machine equivalent.

Another thing to consider when designing the network layout is load balancing. The hardware and algorithm you use is your choice, but be aware that Content Management servers must use sticky sessions. Within Azure you have the choice of Traffic Manager or Load Balancer. Traffic Manager uses DNS redirection, and can redirect based on failover, round robin, or geographic location. Azure Load Balancer has internal and external flavors. External is targeted at only Virtual Machines, and internal is targeted at cloud services or virtual networks. The Azure Load Balancer redirects randomly, with no configuration options.

For a list of required ports, please see Firewall Ports.

Formatte

Deleted:

4.2 What are the hardware deployment requirements?

This section lists the hardware requirements for each computer in a Sitecore Commerce deployment. Although these guidelines represent the approximate amount of resources necessary to operate the recommended software, the actual specifications of the hardware that you choose deploy might vary depending on the resources and requirements of your particular environment.

While Sitecore does support 32bit, Commerce Server does not, so all computers that require Commerce Server must be 64bit.

Server Role	Processor	RAM	Azure Machine
Sitecore XP			
CM Server	8 virtual cores	14GB	A4
CD Server	8 virtual cores	14GB	A4
Search Server	4 virtual cores	7GB	A3
Processing Server	8 virtual cores	14GB	A4
Reporting Service	8 virtual cores	14GB	A4
Commerce Server			
Business Management server	4 virtual cores	7GB	A3
SCpbMD			
Transaction Service	4 virtual cores	7GB	A3
Routing Service	4 virtual cores	7GB	A3
Microsoft Dynamics AX			
Dynamics AX AOS Server*	4 virtual cores	28GB	A6
General			
SQL Server (Sitecore XP)	8 virtual cores	28 GB	D4
SQL Server (Commerce Server)	8 virtual cores	28 GB	D4
MongoDb	4 virtual cores	14 GB	D3
BizTalk**	4 virtual cores	7GB	A3

*More details on Dynamics AX hardware requirements are available at https://msdn.microsoft.com/en-us/library/aa834458(v=ax.10).aspx

**More details on BizTalk hardware requirements are available at https://msdn.microsoft.com/en-us/library/jj248697.aspx

4.3 What are the software deployment prerequisites?

The following is a list of all of the software prerequisites for all roles:

Main software

Role	Software
Server OS	Windows Server 2012 64bit
	Windows Server 2008 R2 64bit
Client OS	Windows 8 64bit
	Windows 7 64bit
Database	SQL Server 2014
	SQL Server 2012
	SQL Server 2008 R2
EAI	BizTalk 2010
	BizTalk 2013
	BizTalk 2013 R2

Runtime

Role	Software
Server OS	.NET Framework 4.5.1
	IIS 7+
	IIS 6 Management Compatibility
Client OS	.NET Framework 4.5.1
Sitecore	Sitecore XP 8+
	Sitecore Commerce Connect

Unsupported

Role	Software
Database	SQL Server Express
Misc.	Azure Roles
	Azure SQL

Chapter 5 Deployment Tools & Notes

5.1 Sitecore

5.1.1 Installation

A Sitecore site can be set up by using an .exe installer, or by manual process using a zip file. The .exe will create a site for you in IIS, copy all of the required files into the folder for the site, attach all of the databases to a SQL Server instance you specify, and update the connection strings in the ConnectionStrings.config. A drawback of using the .exe is that it does not have a silent install mode, so it is less useful for automated deployments. The zip installation is a complete zip file of the website, databases, and data folders, including the database files. The zip installation provides the most flexibility for setting up your site, because it allows you to automate the deployment to whatever degree you want. This also means that you must perform tasks such as setting up the IIS site, putting folders on the disk, placing files, deciding where the database files are placed, and setting up security.

Full details on the Sitecore 8 installation process are located at https://dev.sitecore.net/Downloads/Sitecore_Experience_Platform/8_0/Sitecore_Experience_Platform_8_0.aspx

5.1.2 Database setup

The two database servers that must be set up for Sitecore are SQL server and MongoDB. Unlike the SQL Server setup, MongoDB does not require that scripts are run or databases set up ahead of time, as long as the connection strings to the MongoDB are set up correctly in the ConnectionString.config. Sitecore will create the needed schema from that information.

More details on setting up MongoDB for xDB are located at https://doc.sitecore.net/products/sitecore%20experience%20platform/xdb%20overview/mongodb%20example%20architecture

5.1.3 Deployment of custom solutions

If you want to deploy custom solutions to a Sitecore instance, there are two choices: a Sitecore Package, and a Sitecore Update Package. Both package types allow you to bundle Sitecore Items, as well as files that need to go onto disk. You can also specify a post step class, which must implement IPostStep. This class can be used to perform some custom actions that cannot be performed by packages, for example, create security roles, or updates to the web.config.

When creating packages, consider if the items and files should be split into separate packages. The main reason for separating items is when items are installed, they are written to the database, and available to all servers using that database. However, when a package with files is installed, those files will only be available on the server the package was installed on. This means the package cannot be installed on only one server, but must be installed on all servers.

This may cause some servers running with new functionality, while other servers are broken until the files are installed. The recommended process is to create packages that install the files on all of the servers, and then install a package with items that enable the new functionality. This is less of a concern if your production environment only has a few servers. Large numbers of servers will require more time to install the packages on each one.

The following list describes the pros and cons of choosing a regular package vs. an update package:

Sitecore Packages

- Pros
  - The most popular way of distributing solutions.
  - Can be installed from the Sitecore Desktop (Start -> Development Tools -> Installation wizard).

        - A package designer is included in the website.
    - Cons
        - Packages can only be built with via a Sitecore site.
        - Not useful if you want to update specific item fields. The package contains a complete copy of an item from the source environment, and will overwrite the fields of the destination item.
        - There is limited integration with Visual Studio. To source control items, serialize the items to the “Data\serialization” folder, and then check the folder into your source control.
- Sitecore Update Packages
    - Pros
        - Packages can be created without having a Sitecore site present.
        - Individual fields on an item can be updated.
        - Better developer experience for creating packages.
    - Cons
        - There is no package designer included with the website
        - The package installation screen is available only from a hidden URL. For example, /sitecore/admin/UpdateInstallationWizard.aspx
        - There are no public Sitecore tools to create packages.
        - Team Development for Sitecore from Hedgehog is the only real way to create the packages, and has license requirements.

5.1.4 Setting up Solr

Sitecore has a flexible search framework that allows the development of plugins to integrate any search engine, and have them behave the same way. Sitecore’s default search engine is Lucene, but you can also download a plugin for Solr, which is a better solution for large installations.

The FXM component was shipped with some missing settings in the SOLR configuration file. A corrected file is located at https://kb.sitecore.net/articles/683462 .

Note: For SOLR version 4.9 or higher, schema.xml requires modification to make it compatible with Sitecore. For more information, go to https://kb.sitecore.net/articles/227897.

The Sitecore documentation for setting up Solr is located at:

http://sdn.sitecore.net/Reference/Sitecore%207/Sitecore%20Search%20and%20CMS%20Scaling%20Guide.aspx

The binary files required to set up Solr are located at:

https://dev.sitecore.net/Downloads/Sitecore_Experience_Platform/8_0.aspx

5.2 Commerce Server

5.2.1 Installation

Commerce Server has three installers, all of which have a silent install mode to be used in automated deployments:

- The all-in-one component installer, which installs components such as Commerce Server Core, Desktop Business Tools, and the BizTalk Adapters. This installer is targeted for development and production website environments.
- The Desktop Business Tools installer, which is targeted for desktop machines used by business users, merchandisers, and CSRs.
- The BizTalk installer, which is targeted for installation on BizTalk servers.

The all-in-one installer will also install Commerce Server Staging (CSS), which is installed as a Windows Service. When CSS is installed, it is set up as disabled, and uses Network Service as the running account. To configure and enable CSS, run the Commerce Server Configuration wizard.

5.2.2 Configuration

The Commerce Server Configuration wizard has the options of being run automatically after installation, or run independently later. The Configuration command line tool provides IT professionals a way to perform silent configuration and automate deployment configuration throughout an enterprise network. The Configuration wizard needs to be run on any machine that you plan to use to talk to Commerce Server, and performs the following tasks:

- Creates the Commerce Server admin database, or connects the machine to an existing admin database. The connection string for this admin database is stored in the registry, and used by the Commerce Server Core. A machine can only talk to one admin database.
- Configures Commerce Server Staging by enabling the Windows Service and assigning a proper service account to it.
- Registers Performance Counters.
- Registers COM & COM+ objects for all sub-systems, such as Catalog, Profiles, and Orders.
    - Commerce Server Config, for the Administration API.
    - Commerce Server Catalog Import Host, (or handling Catalog import and export.
    - Commerce Server Orders, for running the Commerce Server pipelines and Orders transactions.
    - Commerce Server Promocode Generator, for generating promotion codes.
- Creates the Widgets Virtual Directory in IIS, which is used by Commerce Server Manager to manage the Profile sub-system schema.

The Configuration wizard can also be run in silent mode with a configuration file to tell the wizard the various settings you need. Silent mode is the only way to avoid installing staging, accomplished by not adding the staging configuration node to the configuration file.

The switches for CSConfig are as follows:

/f Forces the configuration of all features even if they are already configured.

/l Specifies a custom log file. By default a log file with a timestamp based name is created in the temp directory.

/r Applies the configuration to a comma separated list of features (can be combined with /u).

/s Silently (progress bar only) configures the product features based on the settings in an xml file.

/i Silently (no progress bar) configures the product features based on the settings in an xml file.

/u Silently (progress bar only) removes the configuration for the product.

For example, the contents for a configuration xml file for the /s option:

```
<Configuration>
  <SQL ID="CommerceAdminDB">
    <Server>.</Server>
    <Database>MSCS Admin</Database>
    <WindowsSecurity>no</WindowsSecurity>
    <UserName>sa</UserName>
    <Password>sa password</Password>
  </SQL>
  <VirtualDirectory ID="Publishing" Create="True" />
</Configuration>
```

To start the unattended configuration, open a command prompt, navigate to the Commerce Server install folder, type the following, and then press ENTER:

CSConfig.exe /<option name> <parameters>

For example, the following statement uses the CSConfig.xml configuration file to configure all Commerce Server features:

CSConfig.exe /f /l %systemdrive%\install.log /s CSConfig.xml

You can create a CSConfig.xml file by running the Configuration wizard manually, and then saving your settings at the end of the wizard. Or you can create your own by adding the following settings to a file:

```
<Configuration>
  <VirtualDirectory ID="Publishing" Create="True" />
  <NTService ID="StagingService">
    <UserName>CSStageSvc</UserName>
    <Domain>MyDomain</Domain>
    <Password />
  </NTService>
  <SQL ID="CommerceAdminDB">
    <Server>SQLServer</Server>
    <Database>MSCS Admin</Database>
    <WindowsSecurity>yes</WindowsSecurity>
    <Password />
  </SQL>
</Configuration>
```

For the /r option, the following are the available features:

Feature.MS.CS.Core	Commerce Server Pipelines
Feature.MS.CS.Marketing	Marketing List Manager
Feature.MS.CS.Admin	Administration Database
Feature.MS.CS.Catalog	Catalog Import Host
Feature.MS.CS.PCGN	Marketing Coupon Generator
Feature.MS.CS.Misc	Profiles Schema Manager
Feature.MS.CS.Orders	Orders COM+ Applications
Feature.MS.CS.PerfCounters	Performance Counters
Feature.MS.CS.CsStaging	Staging Service

These can be specified with the /r switch for CSConfig. The list of features to be installed needs to be within quotes and comma delimited.

5.2.3 Database Upgrades

The Commerce Server Upgrade wizard handles database upgrades, and has the options of being run automatically after installation, or run independently later. The Upgrade wizard will connect to the admin database, search for all of the Commerce Server Sites and their subsystems, and check if all of the databases schemas have been updated. If databases are found with out-of-date schemas, you can choose to update some or all of the databases. The Upgrade wizard will update all schemas and preserve all data, so there is no worry about data loss.

The Commerce Server Core .dlls know the minimum database schema versions to use. If you install a new version of Commerce Server onto one server in the farm, and then run the Upgrade wizard to update the databases, then all other servers in the farm will probably be broken until the same version of Commerce Server has been installed.

5.2.4 Deployment of custom solutions

Commerce Server Site Packager is a deployment tool that you use to package your site and the accompanying applications and resources into a single file, and then move that file to another environment. You use it to package your Commerce Server site, files from the system file, resources from the Administration database, and SQL Server databases into a single file. You also use it to unpack the Commerce Server site, or sections of it, onto other computers.

For example, after developing or updating your site, you can use Site Packager to package and move your site from your development environment to your test environment, and then finally to your production environment. Site Packager is typically used for the initial deployment of your site. For future deployments, incremental updates, and rollbacks, use Commerce Server Staging.

The typical contents of a pup package are in the following list: Resources

- Catalog
  - Schema and data using the standard catalog xml format.
- Inventory
  - Schema and data using the standard catalog xml format.
- Profile
  - The profile schema xml that contains the profile schema and Site Terms.
  - A SQL file which can create the profile database schema in the target database.
  - An optional SQL file for populating the database with data.
- Orders
  - An xml file with metadata for the transactions and transactions_config database.
- Marketing
  - An xml file with metadata for the marketing and marketing_lists database.

- Applications
  - Resource Web Services
    - All files used by the IIS Applications that contain the web services for each sub-system used for the Desktop Business Tools and BizTalk.
  - Web Site
    - All of the files used by an IIS Website. Typically the main eCommerce site

Pup packages are the only solution for setting up your base Commerce Server site and subsystem databases. However, using pup packages to package and transport your customizations to a base site can be difficult and time consuming. The issue is that these packages are a complex structure of cab files, not zip files that you can open and update. If you make mistake, or realize that you need to make a change, then you must rebuild the entire package. When building out a Commerce Server site, we recommend that you use a version of the CSharp.pup to set up your site, export/import the various subsystem xml files and SQL files, and then move them using your own mechanism. We strongly suggest that you use the Sitecore packaging mechanisms instead of a pup package to bundle the Sitecore Site that is talking to Commerce Server.

5.2.5 How to unpack a pup package

This topic provides steps for unpacking your site on the production Web servers in a Web farm after you have unpacked the site on the business management server in the data domain. For information about unpacking a Web site package on the business management server, see How to Unpack a Site on the Business Management Server.

When you unpack the Web site on additional Web servers, you are adding Web servers to an existing Commerce Server Web site. You can add these resources by using Site Packager and the Custom unpack option, or by using Commerce Server Manager, which calls the Site Packager tool. For information about how to use Commerce Server Manager to add Web servers to an application, see How to Add a Web Server to an Application.

Note:

Before you unpack any Commerce Server package, verify that you are logged on as a member of the Administrators group.

Note:

If you cancel the unpacking process, Commerce Server may not delete the databases that you created during the unpacking process.

To unpack your site on the production Web servers

1. Click Start and navigate to Commerce Server 11, and click on Site Packager.
2. On the Welcome to Commerce Server Site Packager page, select Unpack from a package file (.pup), and then click Next.
3. On the Unpack page, click Browse and locate the package file to unpack. Select Custom unpack, and then click Next.
4. On the Unpack Method page, select Add a Web server to an existing application in an existing site, and then click Next.
5. On the Site Name page, select your site from Existing Sites and then click Next.
6. On the Select Applications page, clear the check boxes for Web services. Make sure that the check box for the site is selected, and then click Next.
7. On the Select IIS Computers, Web Sites and Paths page, click Next.
8. On the Profiling System page, click Next.
9. On the second Profiling System page, click OK.
10. In the Commerce Server Site Packager dialog box, click OK.
11. On the Unpacking is Complete page, click Finish.
12. Repeat steps 1 through 11 on each additional Web server in the Web tier.
13. After unpacking the site, perform the following security tasks for each Web server:
    - Set access permissions to help secure files and folders.
    - Create application pools and make assignments to the Internet Information Services (IIS) Worker Process group.
    - Authorize users and groups to access the Web services.
    - Enable Secure Sockets Layer (SSL) on each Web server.

For more information about these secure deployment tasks, see Configuring the Web Servers.

5.2.6 Maintenance

Inside the Commerce Server Documentation chm file that is part of the Commerce Server install, you will find a “Command Line Tools” section that contains a list of tools that will help you maintain and update your Commerce Server deployment. Additionally, you will want to read the “How to Delete Temporary Tables from the MSCS_CatalogScratch Database” section, which talks about a VBS script that is used to clean up old data in the CatalogScratch database.

5.3 Sitecore Commerce powered by Commerce Server (SCpbCS)

The section discusses the integration between the base Commerce Server installation and Sitecore. Currently, there is no single tool for installing and configuring Sitecore and Commerce Server in one action. Sitecore and Commerce Server must be set up separately before you can use the SCpbCS product.

5.3.1 Installation

The Commerce Connect package must be installed first, because SCpbCS is built on top of it. Commerce Connect is a Sitecore package and must be installed with the Installation wizard which, can be accessed via the Sitecore Desktop (Start -> Development Tools -> Installation wizard).

Once Commerce Connect has installed successfully, you can install SCpbCS. SCpbCS is two Sitecore Update Packages:

- Sitecore Commerce Server Connect, which manages the API level integration with Commerce Server.
- Merchandising Manager, which is the SPEAK -based application for merchandisers to manage their Commerce Server Catalogs from within Sitecore.

Sitecore Update Packages cannot be installed through the regular Sitecore Installation wizard . Instead they must use the Update Installation wizard , which is available at http://sitename/sitecore/admin/UpdateInstallationWizard.aspx. The Sitecore Commerce Server Package must be installed first, followed by Merchandising Manager. Greater step by step detail on the install can be found at Building a Single Server SCpbCS deployment.

5.3.2 Configuration

There is a minor amount of configuration required in the web.config to connect your Sitecore site with a Commerce Server site.

1) Add the Commerce Server configuration section group.
2) Add the Commerce Server http modules to the web.config.
3) Register the Commerce Server Profile Provider.
4) Add the caching configuration to the web.config.
5) Choose Lucene or Solr.

Once you deploy the Sitecore Commerce Server Connect package, you will find the snippets you need to add to the web.config file in \Website\MergeFiles\Merge.CommerceServer.config. You will need to make the following change to the snippets:

- Configuration/CommerceServer/Application[@siteName] must be changed to be the name of your Commerce Server site.
- If you are not using all of the Commerce Server sub-systems, you will need to comment out the http modules and config sections for the sub-systems you are not using.
- The properties under the Commerce Server Profile Provider assume the default fields that come with the sample SolutionStorefrontSite.pup. Update these properties if they do not match your profile schema.

With these configuration changes, the Sitecore site will communicate with Commerce Server at start up. Before you start using your site, be aware of the following information about the Commerce Server Catalog integration:

- When the Sitecore site starts, the Commerce Server Catalog DataProvider will iterate through all of the Inventory, Catalog, Product, and Category definitions and create an Item Template for each. If these Item Templates already exist, then the data provider will not re-create them, and instead will update them with any changes. You can manually force this update without restarting the site by using the “Update Item Templates” in the Sitecore Commerce section of the ribbon.

- The item templates that represent the Commerce Server definitions are only created in the Sitecore Master database. You will always need to publish these items to production.
- By default, no catalogs appear in the Sitecore tree. You must select the catalogs you want to appear. Select the catalogs by browsing to "/sitecore/Commerce/Catalog Management/Catalogs/", selecting the catalogs in the "Selected Catalogs" field, saving the item, and then collapsing and expanding the Catalogs node. You should now be able to browse through the catalog.
- If browsing the catalog tree is slower than expected, then enable Product Buckets by browsing to "/sitecore/Commerce/Settings/Catalog/Product Buckets", and checking the Value checkbox. This will hide all products from the tree, and you will only be able to access them through the Search tab in the Content Editing area. This imitates the behavior of Item Buckets. The Sitecore tree/API slows down when there are approximately 100 items under the same node. The Product Buckets feature hides product nodes to work around this performance issue. Another option here is to organize your catalog so that there are no more than 100 child items at any point in the catalog.
- When you expose a new catalog into the tree, it will not automatically be added to the search indexes. Go to the Indexing Manager (Start -> Control Panel -> Indexing -> Indexing Manager) and rebuild the indexes to add them.

5.3.3 Configure Solr

Either Lucene or Solr, with Lucene being the default, can index Commerce Server. To use Solr, disable the \Website\App_Config\Include\CommerceServer\CommerceServer.Index.Lucene.config file by adding ".disabled" at the end of the file extension, and then enable the \Website\App_Config\Include\CommerceServer\CommerceServer.Index.Solr.config.example by removing the ".example" from the file extension.

5.3.4 Special Sitecore config files

SCpbCS comes with two special config files, CommerceServer.SwitchMasterToWeb.config and CommerceServer.Processing.config, which are used to enable or disable specific functionalities for different Sitecore roles.

The CommerceServer.SwitchMasterToWeb.config config file works the same as the Sitecore SwitchMasterToWeb.config to disable all functionality around the master database so that the Sitecore instance can become a Content Delivery server. This means disabling all dataprovider and search indexes around the master database for SCpbCS.

The CommerceServer.Processing.config file is used for Processing server roles. When setting up a Processing server role, disable all other config files except for this. The CommerceServer.Processing.config file registers a pipeline processor in the Initialize pipeline, which registers types with MongoDB that may have been serialized into MongoDB via xDB.

Sitecore evaluates config files alphabetically, with files in the root of the Include folder being read first, followed by all of the files in the sub folders. The CommerceServer.SwitchMasterToWeb.config must be evaluated after all SCpbCS files and any files developed by 3rd parties. Therefore, if you create your own config files, and these files are listed alphabetically after the CommerceServer folder, you should take the CommerceServer.SwitchMasterToWeb.config out of the CommerceServer folder and add it into a folder named "Z", so that the file will be evaluated last.

These files were not shipped as part of the original 8.0 release, but will be added as part of a later release. If you cannot find them in the \App_Config\Include\CommerceServer\ folder, then you will be able to download them from https://dev.sitecore.net/Downloads/Sitecore_Commerce_powered_by_Commerce_Server/8,-d-,0/Sitecore_Commerce_powered_by_Commerce_Server_8.aspx

5.4 Sitecore Commerce powered by Microsoft Dynamics (SCpbMD)

The section discusses the deployment tools used to set up SCpbMD and connect Microsoft Dynamics AX, Commerce Server, and Sitecore. This section assumes that you have already built out a Microsoft Dynamics AX instance to connect to. This instance requires both Headquarters / AOS, full deployment of the Retail module, and creation of at least one Retail channel to be used for the online store.

5.4.1 Installation

Because SCpdMD is built on top of SCpbCS and Commerce Connect with dependencies on the catalog pieces of Commerce Server, be sure you are familiar with the information in the previous section about tooling.

Once Commerce Connect and SCpbCS have been installed successfully, you can install SCpbMD. SCpbMD contains four Sitecore Update Packages:

- Sitecore Commerce Connect for Dynamics Retail. The main Commerce Connect plugin that allows runtime connectivity with Dynamics AX through the transaction service.
- Sitecore Commerce Routing Framework. The base routing framework that supports extraction and translation of catalog data from Dynamics AX to Commerce Server.
- Routing Extensions for Dynamics Retail. An implementation of the routing framework that translates data between Dynamics AX and Commerce Server Catalog formats.
- Transaction Service for Dynamics Retail. A WCF service that facilitates communication between the Sitecore Commerce Connect and Dynamics AX

You may need to update your Windows hosts file to ensure that the computer running TransactionService is able to access the URL returned by your Microsoft Dynamics service. Sitecore Update Packages cannot be installed through the regular Sitecore Installation wizard, and must use the Update Installation wizard, which is available at http://sitename/sitecore/admin/UpdateInstallationWizard.aspx.

5.4.2 Configuration

There is a minor amount of configuration required in the web.config to connect your Sitecore site with a Microsoft Dynamics AX instance. Once you have finished deploying the packages, merge the contents of \Website\MergeFiles\Merge.DynamicsRetail.Connect.config into the web.config. From this point you will need to point the routing service to your Dynamics AX channel database and configure the routing schedule. More information about configuring the Routing service is located at Configure the Routing service. More information about the Transaction service is located at Building an SCpbMD Transaction Services role .

5.4.3 Configure the Routing service

1. Open Content Editor and edit the following items. Save and publish all item changes.
2. Open /sitecore/Commerce/Commerce Synchronization/Commerce Endpoints/Endpoint_SitecoreWebApi_Configuration_Dev
    a. Ensure the username and password is specified:
    b. Ensure host name is set to the Sitecore instance.
3. Open /sitecore/Commerce/Commerce Synchronization/Commerce Routes/Route Requests/Contoso Commerce Route Request
    a. Ensure the enabled checkbox is checked.
    b. Ensure the approved checkbox is checked.
    c. Clear last run and requested at fields.
4. Open /sitecore/Commerce/Dynamics Routing Extensions/Commerce Endpoints/Azure Contoso Commerce Endpoint - Dynamics Retail Crt
    a. Ensure the sync token is cleared.
    b. Ensure the server is specified. The value should be the Dynamics AX instance URL, which needs to be accessible from where the routing console application is running.

c. Ensure the database is specified. For example, the Dynamics AX Contoso Retail Store channel database name is retailcontososotre.
d. If Integrated Windows Authentication is used, ensure “IsTrusted” is checked.
e. Ensure the port is specified. For example, the Dynamics AX channel database port, 57500.
f. Ensure the username and password are specified. These are needed only if the SQL authentication is being used to access Dynamics AX channel database.

5. Open /sitecore/Commerce/Dynamics Routing Extensions/Commerce Endpoints/Endpoint_FileSystem_Contoso_Dev_Archive
   a. Ensure the path where the XML file is saved is valid.
6. Open Open /sitecore/Commerce/Dynamics Routing Extensions/Commerce Endpoints/Endpoint_FileSystem_Contoso_Dev_Working
   a. Ensure the path where the XML file is saved is valid.
7. By default, when the console app runs, it outputs a catalog XML file from Dynamics AX, and then imports the file into Sitecore Commerce. For situations when you want to import generated catalog XML into Sitecore Commerce at later stage, disable the console app by doing the following:
   a. Uncheck the "Enabled" checkbox for /sitecore/Commerce/Dynamics Routing Extensions/Dynamics Routes/Route_SyncSitecoreCommerceChannel_Contoso_AzureAxVm/Step_Publish Extract

5.4.4 Using Solr instead of Lucene

If you choose to use Solr instead of Lucene, you should first follow the Solr setup steps for SCpbCS and then you will need to copy the following snippet into the defaultSolrIndexConfiguration node:

```
        <fieldMap type="Sitecore.ContentSearch.FieldMap, Sitecore.ContentSearch">
         <fieldNames hint="raw:AddFieldByFieldName">
           <field fieldName="customerfavorites" storageType="NO" indexType="TOKENIZED"
vectorType="NO" boost="1f" returnType="text"
settingType="Sitecore.ContentSearch.SolrProvider.SolrSearchFieldConfiguration,
Sitecore.ContentSearch.SolrProvider"/>
           <field fieldName="stafffavorites" storageType="NO" indexType="TOKENIZED"
vectorType="NO" boost="1f" returnType="text"
settingType="Sitecore.ContentSearch.SolrProvider.SolrSearchFieldConfiguration,
Sitecore.ContentSearch.SolrProvider"/>
           <field fieldName="customeraveragerating" storageType="NO" indexType="TOKENIZED"
vectorType="NO" boost="1f" returnType="double"
settingType="Sitecore.ContentSearch.SolrProvider.SolrSearchFieldConfiguration,
Sitecore.ContentSearch.SolrProvider"/>
         </fieldNames>
        </fieldMap>
```

In addition, the following snippets will need to be copied into the schema file that is generated by Sitecore for Solr:

In the fields section:

```
        <dynamicField name="* tm" type="text general" indexed="true" stored="true"
multiValued="true"/>
        <dynamicField name="*_sci" type="string_ci" indexed="true" stored="true" />
```

In the types node:

```
        <fieldType name="string_ci" class="solr.TextField" sortMissingLast="true"
omitNorms="true">
               <analyzer>
                   <tokenizer class="solr.KeywordTokenizerFactory"/>
                   <filter class="solr.LowerCaseFilterFactory" />
               </analyzer>
```

```
            </fieldType>
```

Chapter 6 Building Deployments

This section provides information about how to build various Sitecore Commerce environments and deployment configurations. You should start by building a single server deployment to get your solution operational. You can then migrate or stage that solution to a multiple-computer deployment, such as a base or enterprise deployment.

6.1 Enabling MSDTC

Commerce Server uses MSDTC to guarantee transactions in the Orders and Catalog subsystems. You will need to enable MSDTC on all servers running Commerce Server, as well as the SQL Server, unless SQL Server and Commerce Server are running on the same machine.

Configuration steps:

1. Click Start, click Run, type dcomcnfg, and then click OK to open Component Services.
2. In the console tree, click to expand Component Services, click to expand Computers, click to expand My Computer, and then click to expand Distributed Transaction Coordinator.
3. Right-click Local DTC, and then click Properties to display the Local DTC Properties dialog box.
4. Click the Security tab.
5. In the Security Settings section, click Network DTC Access.
6. In the Client and Administration section, select Allow Remote Clients and Allow Remote Administration.
7. In the Transaction Manager Communication section, select Allow Inbound and Allow Outbound.
8. In the Transaction Manager Communication section, select Mutual Authentication Required, and select Incoming Caller Authentication Required if you are running MSDTC in a cluster. No Authentication Required is the recommended selection.
9. Select Enable XA Transactions, and then click OK.
10. Repeat steps 1 through 9 on the other Web servers.
11. Use DTCPing.exe to validate the connection between the Web server and the computer that is running SQL Server or an SQL cluster depending on your configuration. To obtain the DTCPing.exe tool, go to http://go.microsoft.com/fwlink/?LinkId=72166.
12. For more information about how to troubleshoot MSDTC, see http://go.microsoft.com/fwlink/?LinkId=72203.

6.2 Building a MongoDb server

MongoDB is used as the collection database for xDB. A single server unsharded install can be used development or testing, but will not scale or protect you from server failure or data loss. For guidance on setting up your MongoDB installation, go to:

https://doc.sitecore.net/products/sitecore%20experience%20platform/xdb%20configuration/configure%20a%20collection%20database%20server

For information on sharded MongoDB, go to:

http://docs.mongodb.org/manual/administration/sharded-clusters/

6.3 Building a SQL Server server

There are no specific requirements for SQL Server setup other than using a supported version. All Sitecore and Commerce Server databases can be split out on different SQL Server instances, except for MSCS_CatalogScratch, which must be on the same server as the Commerce Server Catalog database.

6.3.1 How to configure the Commerce Server Catalog and Inventory databases on different SQL servers

You can install the Commerce Server product catalog and inventory resources on different computers that are running SQL Server. Commerce Server uses SQL Server linked servers to join the resulting data. Although this configuration is possible, there will be some significant performance costs associated with doing this.

For performance reasons, the Commerce Server Catalog and Inventory databases should reside on the same SQL Server instance. The Commerce Server Catalog System frequently accesses inventory data from the inventory database, depending on your configuration. These databases should reside on the same computer that is running SQL Server as a best practice.

To deploy the catalog and inventory databases on different computers that are running SQL Server, you must give Microsoft Distributed Transaction Coordinator (MSDTC) network access on both computers

For more information about how to link computers that are running SQL Server, go to http://go.microsoft.com/fwlink/?LinkId=74207 .

Follow these steps to configure the SQL Server computers as linked servers:

To add the catalog server as a linked server to the inventory server

1. On the inventory server, open Microsoft SQL Server and then click SQL Server Management Studio.
2. Select the inventory database, and then click New Query on the toolbar.
3. In the query pane, type the following query:

   EXEC sp_addlinkedserver '<CatalogServerName>'
   Where <CatalogServerName> is the name of the computer that is the catalog server.

4. Click Execute on the toolbar.

To add the inventory server as a linked server to the catalog server

1. On the catalog server, open Microsoft SQL Server and then click SQL Server Management Studio.
2. Select the catalog database, and then click New Query on the toolbar.
3. In the query pane, type the following query:

   EXEC sp_addlinkedserver '<InventoryServerName>'
   Where <InventoryServerName> is the name of the computer that is the inventory server.

4. Click Execute on the toolbar.

To verify network access for MSDTC

1. Click Start, click Control Panel, double-click Administrative Tools, and then double-click Component Services.
2. In the Component Services console, expand Component Services, expand Computers, right-click My Computer, and then click Properties.
3. In the My Computer Properties dialog box, on the MSDTC tab, in the Transaction Configuration section, click Security Configuration.
4. In the Security Configuration dialog box, in the Security Settings section, verify that Network DTC Access is selected.

5. Click OK two times.
6. Close the Component Services console.

6.4 Building a Single Server SCpbCS deployment

A single server deployment contains all Sitecore roles, a full Commerce Server installation, and all packages. This setup is best for development or QA. It is not recommended for a production scenario.

The steps for setting up a single server are:

1. Create accounts to be used as app pools. For more information, go to Creating accounts and groups
2. Install SQL Server
3. Install MongoDB
4. Setup a Sitecore site
5. Secure Sitecore databases
6. Install Commerce Server
7. Configure Commerce Server
8. UnPup Commerce Server site
9. Adjust Commerce Server schemas
    a. If your pup package does not set up your profile or order database schemas fully, do so now.
10. Import Commerce Server data
    a. Import any catalog, inventory, profiles, orders, or marketing data.
11. Secure Commerce Server databases
12. Grant the Sitecore App Pool access to the Commerce Server Staging DCOM
    a. Using Windows Components Service Manager, assign the Sitecore site app pool user the following permissions on the staging CReplicationProject COM object, located at /Component Services/Computers/My Computer/DCOM Config/{7E95698D-CD3C-4C98-93C7-6510C31F7DDF}:
        i. Execute
        ii. Execute_Local
        iii. Activate_Local
13. Configure Profile System Encryption.
14. Install Commerce Connect
    a. From the Sitecore LaunchPad, click on Desktop -> Start -> Development Tools -> Installation wizard
    b. Upload the Sitecore Commerce Connect package and follow the wizard steps.
15. Install Commerce Server Connect
    a. Navigate to http://<your site>/sitecore/admin/UpdateInstallationWizard.aspx, and follow the steps in the wizard.
    b. Upload the Sitecore Sitecore Commerce Server Connect update file.
16. Copy the contents of \Website\MergeFiles\Merge.CommerceServer.config into the web.config
    a. Update the Commerce Server siteName on the CommerceServer node in the web.config
17. Install Merchandising Manager.
    a. Navigate to http://<your site>/sitecore/admin/UpdateInstallationWizard.aspx, and follow the steps in the wizard.
    b. Upload the Sitecore Merchandising Manager update file.
18. If you are using Solr, set it up now.
    a. Integrate Sitecore and Solr.
    b. Enable SCpbCS Solr indexes.
19. Expose Commerce Server catalogs in the tree.
20. Publish all content to Web.
21. Index all content.

6.5 Building a Single Server SCpbMD deployment

Because SCpbMD builds on top of SCpbCS, you must follow the steps to set up SCpbCS before going through this installation process. Microsoft Dynamics AX is a large install, and probably requires too many resources to run on a single machine along with Sitecore, SQL Server, MongoDB, and other programs. Our recommendation is to have Microsoft Dynamics AX running on a separate server that can be shared by a team, and install the SCpbMD and channel database on the local server. The following steps assume that Microsoft Dynamics AX has been fully set up, and the Channel database has been published to a known location.

To set up SCpbMD, use the following steps:

1. Go through the SCpbCS setup.
2. Install Sitecore Commerce Connect for Dynamics Retail.
    a. Navigate to http://<your site>/sitecore/admin/UpdateInstallationWizard.aspx, and follow the steps in the wizard.
    b. Upload the Sitecore Sitecore.Commerce.DynamicsIntegration.Connect update file.
3. Merge the contents of \Website\MergeFiles\Merge.DynamicsRetail.Connect.config into the web.config.
4. Install Sitecore Commerce Routing Framework.
    a. Navigate to http://<your site>/sitecore/admin/UpdateInstallationWizard.aspx, and follow the steps in the wizard.
    b. Upload the Sitecore.Commerce.RoutingFramework.update update file.
5. Install Routing Extensions for Dynamics Retail.
    a. Navigate to http://<your site>/sitecore/admin/UpdateInstallationWizard.aspx, and follow the steps in the wizard.
    b. Upload the Sitecore.Commerce.DynamicsRouting. Sitecore.Commerce.DynamicsIntegration.Routing.update.
    c. Navigate to http://<your site>/sitecore/admin/UpdateInstallationWizard.aspx, and follow the steps in the wizard.
    d. Upload the Sitecore.CommerceDynamicsIntegration.TransactionService update file.
    e. Configure the Transaction Service web.config file CommerceRunTime Connection String to point to your Dynamics AX for Retail Channel Database.
    f. Open the config file under [website]/App_Config\DynamicsRetail.Connectors.Services.config. Configure the Transaction service endpoints to correctly reflect the URI of service endpoints and identity to be used.
    g. In IIS Manager, locate the "TransactionService" directory in your Sitecore instance, and then click Convert to Application. You may also want to configure a corresponding application pool for the TransactionService application.
    h. You may need to update your Windows hosts file to ensure that the computer running TransactionService is able to access the URL returned by your Microsoft Dynamics service.
6. Configure the routing service.
7. Configure the Profile System Encryption. This only applies to SCpbCS.
8. Run the routing service to update Commerce Server with catalog data.
9. If you are using Solr, set it up now.
    a. Integrate Sitecore and Solr.
    b. Enable SCpbCS Solr indexes.
    c. Enable SCpbMD Solr settings.
10. Expose Commerce Server catalogs in the tree.
11. Publish all content to Web.
12. Index all content.

6.6 Building a Content Management (CM) server

A Content Management server serves up the master database to allow content authors to create and preview site changes before publishing the content to the web database. For example, the Content Delivery server. The CM server has the processing, reporting, and indexing roles disabled. These steps assume a SQL Server instance has already been set up, preferably on a separate machine from the CM server.

The following steps are for setting up the CM server:

1. Create accounts to be used as app pools. For more information, go to Creating accounts and groups
2. Setup a Sitecore site.
3. Secure Sitecore databases.
4. Install Commerce Server.
5. Install and configure SCpbCS in the same way as a single server setup.
6. If you are creating a Microsoft Dynamics AX solution then you now need to install and configure SCpbMD in the same way as a single server setup.
7. If you are using Solr, set it up now.
    a. Follow the Solr setup setting in the Single Server setup for SCpbCS and SCpbMD.
8. Configure the Profile System Encryption. This only applies to SCpbCS.
9. Once the CD server has been setup.
    a. Publish all content to the CD Web database.
    b. Index all content in CD Web or copy the search indexes from the CM to CD environment.

The step by step guide for setting up a Sitecore instance to be a CM server can be found at this link:

https://doc.sitecore.net/products/sitecore%20experience%20platform/xdb%20configuration/configure%20a%20content%20management%20server

6.7 Building a Content Delivery (CD) server

The role of the content delivery server is to deliver content from the web database to the public. A configured CD server should have no access to the master database or its search indexes. There is no access to any master resources. This could cause issues installing packages that contain items from the master database. A workaround for this package issue is to always install all packages before turning the site to be a CD server. These steps assume that SQL Server and MongoDB instances have already been setup, preferably on a separate machine from the CD server.

The following steps are for setting up the CD server:

1. Create accounts to be used as app pools. For more information, go to Creating accounts and groups
2. Setup a Sitecore site.
3. Secure Sitecore databases.
4. Install Commerce Server.
5. Install and configure SCpbCS in the same way as a single server setup.
    a. Exceptions
        i. Enable the \App_Config\Include\CommerceServer\CommerceServer.SwitchMasterToWeb.config.disabled file by removing the .disabled from the file name.
6. Configure the Profile System Encryption. This only applies to SCpbCS.
7. If you are creating a Microsoft Dynamics AX solution then you now need to install and configure SCpbMD in the same way as a single server setup.
    a. Exceptions.
        i. Do not setup the routing service, this is only required by the CM environment.
8. If using Solr now is the time to set it up.
    a. Follow the Solr setup setting in the Single Server setup for SCpbCS and SCpbMD.
9. Once the CD server has been setup.
    a. Publish all content from the CM Web database to the CD Web database.
    b. Index all content in CD Web or copy the search indexes from the CM to CD environment.

The step by step guide for setting up a Sitecore instance to be a Content Delivery server can be found at: https://doc.sitecore.net/products/sitecore%20experience%20platform/xdb%20configuration/configure%20a%20content%20delivery%20server .

A config file that disables all Sitecore related master configs, so you do not have to disable them manually, is available at http://sdn.sitecore.net/reference/sitecore%206/scaling%20guide.aspx. It is recommended that you enable this file as the last step of the process.

6.8 Building a Search role

This role is only relevant if you are using a search provider that can be hosted on a separate server, such as Solr or Coveo. If you are using Lucene as your search solution, then you will need to either index on every front-end server, or index on one server and copy that index to the rest. If you are using a search solution such as Solr, you have greater flexibility because the search engine can be centralized on the network, and data can be indexed once and available everywhere. It is strongly recommended that you use Solr in all production scenarios, and try to use Lucene for development only. Indexing is a very CPU intensive process that could affect the performance of your CM servers. If your search indexing is a long running process, you should consider setting up a separate Sitecore instance used solely for indexing.

The steps for setting up Solr 4.9 are located at: https://archive.apache.org/dist/lucene/solr/ref-guide/apache-solr-ref-guide-4.9.pdf

The Sitecore documentation for setting up Solr is located at:

http://sdn.sitecore.net/Reference/Sitecore%207/Sitecore%20Search%20and%20CMS%20Scaling%20Guide.aspx

The config files required to set up Solr are located at:

https://dev.sitecore.net/Downloads/Sitecore_Experience_Platform/8_0.aspx

6.9 Building a Processing Server role

The job of a processing server is to convert all of the xDB content from MongoDB into a compressed form, which is stored in the reporting database inside of SQL Server. If you have any custom types that are registered with MongoDB in the Initialize pipeline, then you will need to make sure that the registration also happens on the Processing Server. If the custom types are not registered on the Processing Server, some of the engagement data will not be reduced correctly. If you are unsure if you need to register types with MongoDB, check the Sitecore logs after the processor has run. If there are unregistered types, there will be deserialization errors in the Sitecore logs.

The following steps are for setting up the Processing Server:

1. Create accounts to be used as app pools. For more information, go to Creating accounts and groups
2. Setup a Sitecore site.
3. Install, but do not configure, Commerce Server.
4. Install Commerce Connect,
    a. From the Sitecore LaunchPad, click on Desktop -> Start -> Development Tools -> Installation wizard ,
    b. Upload the Sitecore Commerce Connect package and follow the steps in the wizard,
5. Install Commerce Server Connect,
    a. Navigate to http://<your site>/sitecore/admin/UpdateInstallationWizard.aspx, and follow the steps in the wizard.
    b. Upload the Sitecore Sitecore Commerce Server Connect update file.
6. If you are using Solr, set it up now.
    a. Integrate Sitecore and Solr.
    b. Enable SCpbCS Solr indexes.
7. In the \App_Config\Include\CommerceServer\ folder append all files with .disabled except for CommerceServer.config.
8. Disable all of the config files under \App_Config\Include\CommerceServer\, except for CommerceServer.Processing.config, by adding .disabled to the end of their file name
9. Convert Sitecore instance to a Processing Server role. The steps for setting up a processing server can be found at the link below: https://doc.sitecore.net/products/sitecore%20experience%20platform/xdb%20configuration/configure%20a%20processing%20server

6.10 Building a Reporting Service role

The Sitecore Reporting Service fetches reporting data from various data sources, for example, the collection or reporting databases, to use in Sitecore reporting applications, such as Experience Analytics. The Reporting Service can run on the same server alongside other server features, or can run independently on a dedicated server. If the Reporting Service is underutilized you could consider placing it onto the Business Management Server, which is also a low use server. This server requires no Commerce Server or Dynamics AX components.

The detailed steps for setting up a reporting server can be found at:

https://doc.sitecore.net/products/sitecore%20experience%20platform/xdb%20configuration/configure%20a%20reporting%20service%20server

6.11 Building the Business Management Server role

The Business Management Server role is used to host the Commerce Server sub-system web services that are used by the desktop business tools, BizTalk, and Commerce Server Staging. No Sitecore XP setup is required on this role.

6.11.1 Sub-system web services role

The following steps are for setting up the role:

Create accounts to be used as app pools. For more information, go to Creating accounts and groups

1. Install Commerce Server.
2. Configure Commerce Server.
3. UnPup Commerce Server site.
    a. You will only need to unpup the web service applications on this machine.
4. Configure the Profile System Encryption.
5. Update the web service application pools to use the service accounts.
6. Secure Commerce Server databases.

6.11.2 Commerce Server Staging role

Commerce Server Staging is used to move Commerce Server data between environments or different sites. If the master and web environment are pointing to the same Commerce Server site instance, as soon as data is changed on the master environment, the data will be live on web. To ensure that Commerce Server data does not unexpectedly go live, it is strongly recommended that you have one Commerce Server site for your CM environment, another for your CD, and then use Commerce Server to move the data.

Commerce Server Staging projects can be run from the staging tool, but also from the Publishing dialog in Sitecore. Running Commerce Server Staging projects from the Sitecore Publishing dialog has of advantages, because you can now publish your Sitecore and Commerce Server data to production at the same time. The recommended setup is to have Commerce Server Staging on a Content Management server, and install it on the Business Management Server in the production environment.

On the CM server

1. When installing Commerce Server make sure to install the Staging Service.
2. Configure the Staging Service using the Commerce Server Configuration wizard.

Production Business Management Server

1. Installing Commerce Server make sure to install the Staging Service.
2. Configure the Staging Service using the Commerce Server Configuration wizard.

6.12 Building a BizTalk role

This section provides instructions for configuring BizTalk Server in a Commerce Server enterprise deployment. The computer that is running BizTalk Server runs the Commerce Server line-of-business adapters to accept and transmit business-related messages, such as purchase orders and inventory requests.

You can install Commerce Server and BizTalk Server on the same computer, but you must run the configuration wizard for each product before you install and configure the next product. For example, first install and configure Commerce Server before you install and configure BizTalk Server. Alternatively, install and configure BizTalk Server before you install and configure Commerce Server. The Microsoft guide for BizTalk installation is located at: https://msdn.microsoft.com/en-us/library/jj248681.aspx.

If you install Commerce Server and BizTalk Server on the same computer, do not reference the same SQL Server databases for both products. You should run separate databases for each product.

Perform these tasks to configure the BizTalk Server.

2. Perform pre-installation requirements:
    - Enable clustering support by enabling MSDTC.
3. Install BizTalk Server.
4. Add the BizTalkAdmin group to the Administrator group, see How to Add the BizTalkAdmin Group to the Administrators Group.
5. Configure BizTalk Server.
6. Install Commerce Server Adapters for BizTalk Server, see How to Install Commerce Server Adapters for BizTalk Server.
7. Set authorization roles for the CSLOB service account, see How to Set Authorization Roles for the BizTalk Adapters.
8. Add Commerce Server adapters to your BizTalk Server. For more information, go to How to Add a BizTalk Adapter.
9. Configure the send and receive endpoints for each adapter, see Working with the BizTalk Adapters.

6.12.1 How to add a BizTalk adapter

After you have installed the BizTalk adapters for Commerce Server on the computer running BizTalk Server, you can create new adapters of the four types provided with Commerce Server. This topic describes the steps required to add Commerce Server adapters to your BizTalk Server configuration after you have installed the adapters on the BizTalk Server computer.

To add a Commerce Server adapter to BizTalk Server

1. In the BizTalk Server Administration Console, expand BizTalk Server Administration, expand the appropriate BizTalk Group, and then expand Platform Settings.

   Note:

   You may also perform these steps in Visual Studio.

   Note:

   You may optionally expand Adapters at this point. If one or more of the Commerce Server adapters have already been added to BizTalk Server, you will see them as child nodes of the Adapters node.

2. Right-click Adapters, click New, and then click Adapter.
   The Adapter Properties dialog box opens.
3. In the right pane of the Adapter Properties dialog box, in the Name box, type a descriptive name for this adapter.
4. In the Adapter box, select one of the following Commerce Server adapters from the drop-down list based on the type of adapter you want to add:
    - Commerce Server Catalog

- Commerce Server Inventory
- Commerce Server Orders
- Commerce Server Profiles

5. In the Description box, type a description for the adapter.
6. Click OK to complete the process of adding a particular Commerce Server adapter.
7. Repeat steps 2 through 6 as required to add other Commerce Server adapters.

Note:
You only need to create new adapters within the BizTalk Server Administration Console for the Commerce Server adapters that you intend to use.

6.13 Building an SCpbMD Transaction Services role

The Transaction Service is a wcf-based solution that encapsulates Dynamics AX transaction capabilities into a simple service endpoint. This provides a customization point for customers and partners. The Transaction Service is provided as source code and can be customized to implement specific business requirements. The transaction service can be hosted within Sitecore or as a standalone wcf service, but no matter if you are hosting inside or outside of Sitecore, the Transaction Service package must be installed on all CM and CD servers, because there are settings and dlls that are required to talk to the service.

Currently a Sitecore site and Transaction service are capable of talking to only a single Dynamics AX Channel database. If you need to talk to multiple Channel databases, then you will need to create a separate Sitecore site, Commerce Server site, and transaction service to talk to each one.

6.13.1 Hosting the transaction service in Sitecore

1. Perform the following on all CM & CD servers
2. Navigate to http://<your site>/sitecore/admin/UpdateInstallationWizard.aspx, and follow the steps in the wizard.
3. Upload the Sitecore.Commerce.DynamicsIntegration.TransactionService update file.
4. Configure the Transaction Service web.config file CommerceRunTime Connection String to point to your Dynamics AX for Retail Channel Database.
5. Open the config file under [website]\App_Config\DynamicsRetail.Connectors.Services.config. Configure the Transaction service endpoints to reflect the URI of service endpoints and identity to be used.
6. In IIS Manager, locate the "TransactionService" directory in your Sitecore instance, and then click Convert to Application. You may also want to configure a corresponding application pool for the TransactionService application.
7. You may need to update your Windows hosts file to ensure that the computer running TransactionService is able to access the URL returned by your Microsoft Dynamics service.

6.13.2 Hosting the transaction service outside of Sitecore

Even if you choose to host the transaction service outside of Sitecore in a separate IIS instance or some other form, you will still need to install the transaction package into Sitecore. Perform the steps in the previous task before attempting this one.

1. Create a separate IIS site & App Pool.
2. From one of the CM or CD servers:
    - Copy the contents of $\website\TransactionService\ into the root of the new website.
3. Configure the Transaction Service web.config file CommerceRunTimeConnectionString to point to your Dynamics AX for Retail Channel Database.

6.14 Building an SCpbMD Routing Services role

The Routing Framework provides the ability to define the needed steps to extract and transform catalog data from one system to another, in this case from Dynamics AX to Commerce Server Catalog database. This service can live within Sitecore and run as a job, or it can run outside of Sitecore as a Windows Service or Console app. If the service runs outside of Sitecore, then it uses the Sitecore Web Item API to communicate with Sitecore to retrieve route settings. If you are running the routing service outside of Sitecore, then you will still need to install the routing package on one of your CM servers so that the required item templates are created in Sitecore, and can be published to production.

In larger environments, with larger or more frequent synchronization needs, you may want to separate the Routing Service from the Sitecore installation to reduce the impact of synchronization on the Sitecore authoring environment. Depending on performance needs, this service can be deployed on the same machine as the Commerce Server Staging Service, or on a separate independent machine. The Routing Service can temporarily consume large amounts of memory as it gathers the extraction from Dynamics and generates the extraction file so you will need to ensure that the machine it is running on has adequate memory for the task.

In smaller environments, you can run the Routing Service to run inside of Sitecore as a Sitecore task to reduce the need for another machine.

6.14.1 Hosting the routing service in Sitecore

Regardless of whether you are going to be running the routing service inside or outside of Sitecore, you will still need to install this package into Sitecore, as it installs items and templates that are required to configure routes.

1. Navigate to http://<your site>/sitecore/admin/UpdateInstallationWizard.aspx, and follow the steps in the wizard.
2. Upload the Sitecore.Commerce.RoutingFramework update file and install.
3. Navigate to http://<your site>/sitecore/admin/UpdateInstallationWizard.aspx, and follow the steps in the wizard.
4. Upload the Sitecore.Commerce.DynamicsIntegration.Routing update file and install.
5. Configure the routing service.

6.14.2 Hosting the routing service in a Windows Service

The following steps will allow you to run the routing service as a Windows Service instead of a Sitecore job. The routing package must be installed in Sitecore, as this will remain the main source of configuration. Also, the Routing Service must be run on the same machine that is running Sitecore Content Management.

1. When you install the Routing Framework package, it will set up a Sync Task to run the routing synchronization service. Because this is running as a Windows Service, it can be disabled or deleted.
    - Delete Task
        - Browse to Open /sitecore/system/Tasks/Schedules/Commerce Synchronization Schedule.
        - Delete the Item.
    - Disable Task
        - Browse to Open /sitecore/system/Tasks/Schedules/Commerce Synchronization Schedule.
        - You cannot disable the task entirely. Instead, set the “Schedule” date to be sometime in the far future. For example, set the year to 2100.
2. Download the Routing Framework SDK package.
3. Unzip and follow the steps in the ReadMe.txt to build the solution.

4. Expand the Sitecore.Commerce.DynamicsIntegration.Routing update package and the package.zip contained within.
    - Inside the addedfiles folder take the Templates folder and copy it into the same folder as the Windows Service .exe.
    - Inside the addedfiles folder take the contents of the bin folder and copy it into the same folder as the Windows Service .exe.
    - From the Commerce Server install folder grab CommerceServer.Core.Catalog.dll and CommerceServer.Core.CrossTier.dll and add them into the same folder as the Windows Service .exe.
5. The Routing.SynchronizationService project contains the code for the Windows Service.
6. Inside the App.config (which becomes Sitecore.Routing.SynchronizationService.exe.config after the build), change below according to your Sitecore Commerce Catalog Web Web Service information:

```
        <client>
         <endpoint
address="http://localhost:2000/SolutionStorefrontSite_CatalogWebService/CatalogWebService.asmx
" binding="basicHttpBinding" bindingConfiguration="CatalogWebServiceSoap"
contract="Sitecore.Commerce.Catalog.CatalogWebServiceSoap" name="CatalogWebServiceSoap"/>
        </client>
```

   Note: Windows Authentication is required to access Sitecore Commerce catalog web service
7. Inside the SynchronizationService.cs class, in its OnStart method, update the code with the connection details to your Sitecore instance. Ideally, you should read these settings from a configuration file instead of hardcoding them.
8. Details on how to install the resulting .exe as a service are located at: https://msdn.microsoft.com/en-us/library/zt39148a%28v=vs.110%29.aspx#BK_Install

6.14.3 Hosting the routing service in a Console Application

The following steps will allow you to run the routing service as a Console Applications instead of a Sitecore job. You still need to install the routing package in Sitecore, as this will remain the main source of configuration.

1. When you install the Routing Framework package, it will set up a Sync Task to run the routing synchronization service. Because this will be running as a Console Application, it can be disabled or deleted.
    - Delete Task
        - Browse to Open /sitecore/system/Tasks/Schedules/Commerce Synchronization Schedule.
        - Delete the Item.
    - Disable Task
        - Browse to Open /sitecore/system/Tasks/Schedules/Commerce Synchronization Schedule.
        - You cannot disable the task entirely. Instead, set the “Schedule” date to be sometime in the far future. For example, set the year to 2100.
2. Download the Routing Framework SDK package.
3. Unzip and follow the steps in the ReadMe.txt to build the solution.
4. Expand the Sitecore.Commerce.DynamicsIntegration.Routing update package and the package.zip contained within.
    - Inside the addedfiles folder take the Templates folder and copy it into the same folder as the Windows Service .exe.
    - Inside the addedfiles folder take the contents of the bin folder and copy it into the same folder as the Windows Service .exe.
    - From the Commerce Server install folder grab CommerceServer.Core.Catalog.dll and CommerceServer.Core.CrossTier.dll and add them into the same folder as the Windows Service .exe.
5. The Routing.Console project contains the code for the console application

6. Inside the App.config, which becomes Sitecore.Commerce.DynamicsRetail.Synchronization.Console.exe.config after the build, change below according to your Sitecore instance information:

```
        <appSettings>
         <add key="Host" value="storefront" />
         <add key="UserName" value="sitecore\admin" />
         <add key="Password" value="b" />
         <add key="SitecoreDatabase" value="master" />
        </appSettings>
```

Change below according to your Sitecore Commerce Catalog Web Web Service information:

```
        <client>
         <endpoint
address="http://localhost:2000/SolutionStorefrontSite_CatalogWebService/CatalogWebService.asmx
" binding="basicHttpBinding" bindingConfiguration="CatalogWebServiceSoap"
contract="Sitecore.Commerce.Catalog.CatalogWebServiceSoap" name="CatalogWebServiceSoap"/>
        </client>
```

Note: Windows Authentication is required to access Sitecore Commerce catalog web service.

Once the routing settings have been set in Sitecore:

1. Run Sitecore.Commerce.DynamicsRetail.Synchronization.Console.exe.
2. It should output information in the console window, and eventually complete.
3. If it did not work, look at /sitecore/Commerce/Commerce Channels/Contoso/Channel History/ or /sitecore/Commerce/Dynamics Routing Extensions/Dynamics Routes/Route_SyncSitecoreCommerceChannel_Contoso_AzureAxVm/Route History for the latest history file. The Tracking Log property will display log output.
4. If there are no errors from console output, but there was no catalog XML generated, clear the "Last Run" and "Requested At" values, and set "Repeat Delay" to a smaller value, such as 2 instead of the default 120, under /sitecore/Commerce/Commerce Synchronization/Commerce Routes/Route Requests/Contoso Commerce Route Request.
5. Once the execution is completed, the catalog XML should be generated under “c:\Sitecore Synchronization Data\Working\Contoso_Dev\”. This location is defined by /sitecore/Commerce/Dynamics Routing Extensions/Commerce Endpoints/Endpoint_FileSystem_Contoso_Dev_Working (or _Archive).
6. If the Step_PublishExtract is enabled, which is the default, the catalog data should be imported. You can validate this in catalog manager by looking at the import status/history.

6.15 Building a Microsoft Dynamics AX AOS server

The following link from Microsoft will guide you through setting up Microsoft Dynamics AX and an AOS instance.

Microsoft Dynamics AX Retail for IT pros and developers [AX 2012]

https://technet.microsoft.com/en-us/library/jj710398.aspx

Chapter 7 Deploying Sample Sites

Both SCpbCS and SCpbMD have sample eCommerce sites that demonstrate some of the functionality that is possible with the two products. This chapter provides the additional steps required to get the sites up and running.

7.1 SCpbCS MVC Sample Site

Before you attempt this, ensure that already gone through the Single Server SCpbCS install, or set up SCpbCS on a CM or CD server.

The setup steps are as follows:

1. Install Sitecore Commerce Server Connect Sample MVC Site.
    a. Navigate to http://<your site>/sitecore/admin/UpdateInstallationWizard.aspx, and follow the steps in the wizard.
    b. Upload the Sitecore Commerce Server Connect Sample MVC Site update file.
2. Merge the $\Website\MergeFiles\Merge.MVCSite.config into the web.config.
3. Expand the MVC Site Deployment Scripts zip (i.e. Deploy.zip) file.
4. UnPup the Commerce Server site.
    a. Install the \Database\pup\SolutionStorefront.pup by double-clicking on it, or by opening the Commerce Server Site Packager and selecting the pup file.
    b. Select Quick Unpack.
    c. Set the Site name to SolutionStorefrontSite.
        i. If you choose a name other than SolutionStorefrontSite, you will need to update the siteName attribute on the CommerceServer element in the web.config to match this name.
    d. Set the database server name. It is strongly suggested that you use Windows Integrated Security.
5. Secure the Commerce Server resources.
    a. Update the web service app pools accounts.
    b. Add the expected Desktop Business Tool users into the various AzMan files.
    c. Lock down the Commerce Server databases with the accounts from the web service and site app pools.
        i. You can secure the Commerce Server databases manually, or you can use the \Database\Security\CommerceServer_Security.sql file as a helper. Search the document to replace the database names and user names with yours, and then run the SQL file against your database server.
6. Import Catalog
    a. Use Catalog Manager or ImportCatalog.exe to import:
        i. The catalog data and schema file \Database\Catalog\Catalog.xml.
        ii. The inventory data and schema file \Database\Catalog\Inventory.xml.
7. Import Profile
    a. Run the \Database\Profiles\AdjustSchema.sql file against the SolutionStorefrontSite_profiles database.
    b. Using Commerce Server Manager import the profile schema file \Database\Profiles\Schema.xml and point the database configuration to SolutionStorefrontSite_profiles.
8. Import Orders
    a. Run the \Database\Profiles\PaymentMethod.sql and \Database\Profiles\ShippingtMethod.sql files against the SolutionStorefrontSite_transactionconfig database.
9. Do an IISReset and then log into Sitecore.
10. Expose Commerce Server catalogs in the tree.
11. Publish all content to Web.
12. Index all content.

If you have set up a separate Processing server, then you also need to deploy the MVC Site update package on that server, because it has a processor in the Initialize pipeline that needs to register some types with MongoDB.

7.2 SCpbMD Reference Storefront

Before you attempt this, ensure that you already gone through the Single Server SCpbMD install, or set up SCpbMD on a CM or CD server.

The released package for SCpbMD is found on GitHub. Within the zip file:

1. The Database folder contains all of the artifacts related for configuring the Commerce Server site and
2. The Packages folder contains all of the artifacts related for configuring Sitecore.

The setup steps are as follows:

1. Install Contoso Images.
    a. Navigate to http://<your site>/sitecore/admin/UpdateInstallationWizard.aspx, and follow the steps in the wizard.
    b. Upload the ContososImages-1 update file.
        i. The Contoso images update package are located on dev.sitecore.net under the Microsoft Dynamics section.
2. Install Sitecore Reference Storefront powered by Microsoft Dynamics.
    a. Navigate to http://<your site>/sitecore/admin/UpdateInstallationWizard.aspx, and follow the steps in the wizard
    b. . Upload the Sitecore.Reference.Storefront.Common update file.
    c. Upload the Sitecore.Reference.Storefront.Powered.by.Microsoft.Dynamics update file.
3. Merge the $\Website\MergeFiles\Merge.Commerce.Storefront.config into the web.config.
4. Expand the Sitecore.Reference.Storefront.for.SCpbMD.<version>.zip file
5. UnPup the Commerce Server site.
    a. Install the \Database\pup\SolutionStorefront.pup by double clicking it, or opening the Commerce Server Site Packager and selecting the pup file.
    b. Select Custom unpack.
    c. Set the Site name to SolutionStorefrontSite.
        i. If you choose a name other than SolutionStorefrontSite. you will need to update the siteName attribute on the CommerceServer element in the web.config to match this name.
    d. Only select Product Catalog as the resource to unpack. SCpbMD does not interact any other part of Commerce Server.
    e. Click next on the Global Resource Pointers page
    f. On the Database Connection Strings page update the server name if you are not using a local SQL server, then click Next
    g. On the Select Applications page make sure SolutionStorefrontSite_CatalogWebService is the only application selected and then click Next
    h. On the Select IIS Computers, Web Sites and Paths page update the IIS Web Site target for the Catalog web service if you are not using the default, then click Next
6. Secure the Commerce Server resources
    a. Update the web service app pools accounts
    b. Add the expected Desktop Business Tool users into the various AzMan files
    c. Lock down the Commerce Server databases with the accounts from the web service and site app pools.
        i. You can secure the Commerce Server databases manually, or you can use the \Database\Security\CommerceServer_Security.sql file as a helper. Search the document to replace the database names and user names with yours, and then run the SQL file against your database server.
7. Import Catalog
    a. Use Catalog Manager or ImportCatalog.exe to import:
        i. The catalog data and schema file \Database\Catalog\Sitecore Commerce Reference Storefront for Dynamics Sample Catalog.xml.

8. Do an IISReset and then log into Sitecore.
9. Expose Commerce Server catalogs in the tree.
10. Publish all content to Web.
11. Index all content.

If you have a separate Processing server, then you need to deploy the Reference Storefront update package on that server, because it has a processor in the Initialize pipeline that needs to register some types with MongoDB.

7.3 SCpbCS Reference Storefront

Before you attempt this, ensure that you already gone through the Single Server SCpbCS install, or set up SCpbCS on a CM or CD server. Please note there is no need to perform the unpup portion of the Single Server SCpbSC install.

The released package for SCpbCS is found on GitHub. Within the zip file:

3. The Database folder contains all of the artifacts related for configuring the Commerce Server site and
4. The Packages folder contains all of the artifacts related for configuring Sitecore.

The setup steps are as follows:

1. Install Adventure Works Images.
    a. Log into the Sitecore site.
    b. From the Sitecore menu select Development Tools\Installation Wizard, and follow the steps in the wizard.
    c. Upload the Packages\Adventure Works Images.zip file.
2. Install Sitecore Reference Storefront powered by Commerce Server.
    a. Navigate to http://<your site>/sitecore/admin/UpdateInstallationWizard.aspx, and follow the steps in the wizard.
    b. Upload the Sitecore.Reference.Storefront.Common update file.
    c. Upload the Sitecore.Reference.Storefront.Powered.by.CommerceServer update file.
3. Merge the $\Website\MergeFiles\Merge.Commerce.Storefront.config into the web.config.
4. Expand the Sitecore.Reference.Storefront.for.SCpbCS.<version>.zip file.
5. UnPup the Commerce Server site.
    a. Install the \Database\pup\SolutionStorefront.pup by double-clicking on it, or by opening the Commerce Server Site Packager and selecting the pup file.
    b. Select Quick Unpack.
    c. Set the Site name to CSSolutionStorefrontSite.
        i. If you choose a name other than CSSolutionStorefrontSite, you will need to update the siteName attribute on the CommerceServer element in the web.config to match this name.
    d. Set the database server name. It is strongly suggested that you use Windows Integrated Security.
6. Secure the Commerce Server resources.
    a. Update the web service app pools accounts.
    b. Add the expected Desktop Business Tool users into the various AzMan files.
    c. Lock down the Commerce Server databases with the accounts from the web service and site app pools.
        ii. You can secure the Commerce Server databases manually, or you can use the \Database\Security\CommerceServer_Security.sql file as a helper. Search the document to replace the database names and user names with yours, and then run the SQL file against your database server.
7. Import Catalog
    a. Use Catalog Manager or ImportCatalog.exe to import:
        i. The catalog data and schema file \Database\Catalog\Catalog.xml.
        ii. The inventory data and schema file \Database\Catalog\Inventory.xml.
8. Import Profile
    a. Run the \Database\Profiles\AdjustSchema.sql file against the SolutionStorefrontSite_profiles database.
    b. Using Commerce Server Manager, import the profile schema file \Database\Profiles\Schema.xml and point the database configuration to SolutionStorefrontSite_profiles.
9. Import Orders
    a. Run the \Database\Profiles\PaymentMethod.sql and \Database\Profiles\ShippingtMethod.sql files against the SolutionStorefrontSite_transactionconfig database.
10. Do an IISReset and then log into Sitecore.

11. Expose Commerce Server catalogs in the tree.
12. Publish all content to Web.
13. Index all content.

If you have a separate Processing server, then you need to deploy the Reference Storefront update package on that server, because it has a processor in the Initialize pipeline that needs to register some types with MongoDB.

Chapter 8 Securing the Deployment

8.1 What are the secure deployment requirements?

Sitecore Commerce uses Internet Information Services (IIS), Windows Authorization Manager, SQL Server database role assignments, and Windows Component Services to create and help maintain a secure deployment. This section provides instructions for working with these software tools to help secure access to Sitecore Commerce resources, and establish the secure communication channels that are an intrinsic part of a commerce deployment.

8.1.1 Authentication and Secure Access Requirements

Authentication in Sitecore Commerce Server involves validating user credentials for internal business users and internal service accounts. Authentication of external site visitors should be done by Sitecore. Commerce Server uses a different authentication approach for each of these user segments. In all scenarios, users and services need access to the following types of site assets in the Web site application:

1. Web server resources such as pages, images, and other static files.
2. Database resources such as per-user, application-wide, or other forms of dynamic data.
3. Network resources such as remote file system resources and directory stores such as Active Directory.

In turn, the Web site application needs access to system resources such as the registry, event logs, and configuration files.

Review the following sections for a summary of the secure deployment methods that are used to support access by the different user segments.

External Access by Site Users

Review the Sitecore Installation Guide for instructions on how to set up a Sitecore site and with the accounts and permissions that are required. Controlled access to specific databases and the ability to read or write to the databases is controlled through SQL Server database login accounts and database user role mappings. To review the predefined database role mapping requirements, see What are the required accounts and groups to create? It is recommended to use integrated security in the Commerce Server database connections strings instead of explicit SQL logins. Commerce Server will then use the application pool account to talk to the Commerce Server databases, not the user account that is logged in.

Sitecore uses the ASP.NET membership provider framework for authentication and authorization. Using this framework gives you many advantages when it comes to xDB and site personalization. By default, Sitecore uses the basic SQL provider for membership and roles, which should work for your needs. However, you do have the option to have these work side by side with other providers, such as the Commerce Server Membership and Profile providers. By using the Sitecore switcher framework, Sitecore can authenticate users against the Commerce Server Profile database. With the Commerce Server Profile provider, you can use whatever membership provider you like and persist some or all of your profile data into the Commerce Server Profile database. Full details on profile integration are located at http://commercesdn.sitecore.net/SCpbCS80/SitecoreCommerceConnectGuide/en-us/#Concepts/c_ProfileStoreFundamentals.html .

Internal Access by Business Users and Commerce Server Services

Internal business users must have access to the Commerce Server Business Management applications and Commerce Server services must have access to various Commerce Server resources. Commerce Server implements two levels of granular security that address these areas.

The first level of security is implemented by using Windows Authorization Manager to define scopes, roles, tasks, and operations to manage access to the Catalog, Inventory, Marketing, Profiles, and Orders systems via the web services. Between four and ten security roles are defined per system.

The second level of security is implemented by mapping users to predefined database roles. This grants access to operations, not to resources, for the service identities. The back-end resource managers, such as databases, trust the application to authenticate users and grant access to the

trusted service identity. For example, a database administrator might grant access exclusively to a specific human resource application, but not to individual users.

Business User Access to Business Management Applications

The following list summarizes the secure deployment requirements used to support business users access to Business Management applications, and these applications access to Web services. These deployment tasks are performed on the production server where the Commerce Server Web services are run. In an Enterprise deployment, this is the business management server.

- Each business user is assigned a domain account.

Business management Windows or Active Directory groups are created according to your business needs. To review the set of predefined security roles, see What are the required accounts and groups to create?

- Each business user domain account is added to one or more business management group accounts.
- You assign business management groups to one or more predefined authorization roles. For more information, see What Are the Minimum Authorization Roles to Assign?
- Controlled access to specific databases and the ability to read or write to the databases is controlled through SQL Server database login accounts and database user role mappings. To review the predefined database role mapping requirements, see What are the required accounts and groups to create?
- IIS application pool and IIS worker process group assignments are made. You create an IIS application pool for each Web service and you assign the Web service account to its application pool and the IIS worker process group.
- You must grant Web application service identities write permissions to the temporary ASP.NET and Windows temporary folders. To enable user access to the Catalog Web service, you must assign the Catalog Web service identity write permissions to the Catalog authorization role.

Commerce Server Services Access to Commerce Server Resources

In addition to Commerce Server Web services, you can run many additional services, such as Staging, and Commerce Server adapters. The following summarizes the secure deployment requirements used to support these services access to Commerce Server resources:

- You assign each service a service identity. To review the service accounts that you need to create, see What are the required accounts and groups?
- Controlled access to specific databases and the ability to read or write to the databases is controlled through SQL Server database login accounts and database user role mappings. To review the predefined database role mapping requirements, see What are the required accounts and groups to create?

In addition, several additional deployment tasks are required to enable specific services to access specific resources. The following table summarizes these tasks.

Service	Requirements
Commerce Server Staging	Grant the service group, CSS_SG, permission to replicate the Internet Information Services (IIS) metabase. For more information, see How to Configure Access to the IIS Metabase.
Commerce Server Adapters	Provide the service identity, CSLOB, access to Web services by assigning them to predefined authorization roles. For more information, see How to Set Authorization Roles for the BizTalk Adapters.

Authorization Role-Based Access

Commerce Server provides several predefined roles to which you assign business users, so that they can perform specific tasks such as editing a catalog, creating a discount, and deleting an order. To restrict business users from performing all tasks, you assign them to specific roles such as the CatalogPropertyEditor role, where users can only manage individual catalog properties. By assigning user accounts or Windows groups to the administrator roles, such as MarketingAdministrator or OrdersAdministrator, these users can perform any operation associated with the corresponding Commerce Server system. For example, the MarketingAdministrator role lets users perform any operation in the Marketing System.

With role-based access control, you specify access control relative to the organizational structure of your company. You use Windows Authorization Manager to add individual users or user groups to a role. Before you assign business user access to the Authorization Manager roles, we recommend that you assign the business user to a Windows group, and then give the Windows group Authorization Manager permissions on the Web services.

Database Role Mapping and the Trusted System Model

To simplify authentication and eliminate the requirement of configuring database roles and permissions for each business user, Commerce Server uses the trusted system model. In this model, you configure database roles and permissions for groups of users according to user role, such as Catalog Editor or Marketing Manager, and then associate the individual business users with these user roles.

With this model, the Business Management server uses fixed identities to access the resources on the database server. The security context of the originating business user does not pass through the service at the operating-system level. The database trusts the Web server to authenticate users, and to let only authenticated users access the database using the trusted identity.

The advantage of using the trusted system model is that users cannot access back-end data directly without using the application and being subjected to application authorization. Only the Web tier service account has access to the back-end resources. Additionally, you only have to configure access control lists (ACL) for the Web tier service account instead of for every individual user identity. The trusted system model also supports connection pooling. This enables multiple clients to reuse available, pooled connections because all back-end resource authentication assumes the security context of the service account, regardless of user identity.

The disadvantage of using the trusted system model is that an attacker who manages to compromise the Web tier has broad access to back-end resources because the back-end resources rely completely on the Web tier to authenticate users. The trusted system model also makes auditing difficult because the Web tier service account masks the originating user identities.

Windows Authentication and Windows Integrated Security

Commerce Server supports Windows Authentication to SQL Server. This is also known as Windows Integrated Security. We recommend that you use Windows authentication for a Commerce Server installation. With Windows authentication, Windows Server uses Windows user accounts to authenticate to SQL Server. Commerce Server sets a tag in the connection string that tells the SQL Server to use Windows authentication when checking the security context of the user trying to access a given database.

When you use Windows Authentication, user names and passwords are not stored in the SQL Server connection string, and are not changed when the SQL Server password is reset.

SCpbMD communicates with Dynamics solely through the Commerce Run Time assemblies provided by Microsoft Dynamics. These assemblies require Windows Integrated Security access to a published Channel database. Once authentication has been established to the channel database. then the channel database contains authentication protocols to the Commerce Data Exchange (CDX) and Real Time Services (RTS). These security protocols are resolved transparently by the Commerce Run Time assemblies in communicating with these Dynamics Services.

Dynamics AX provides utilities within the product to configure and manage these access protocols.

8.2 Data Encryption

In an e-commerce site, you process sensitive information, such as customer credit card numbers and user profile information. Commerce Server recommends using these methods to encrypt data to protect this information:

- Use of Secure Sockets Layer (SSL) to encrypt non-profile data. SSL is a scheme for protocols such as HTTP and others to transmit data in a secure manner.
- Use of profile encryption keys to encrypt profile data. For more information, see How to configure encryption keys for Profiles System data.
- Use SQL Server's Transparent Data Encryption to encrypt the content of the log and data files of the databases. If this feature enabled, if anyone gets a copy of your databases, they will be unable to read it. For more information on Transparent Data Encryption, go to: https://msdn.microsoft.com/en-ca/library/bb934049.aspx.
- Encrypt the database connection between the website and databases by enabling Encrypted Connections in SQL Server. For more information on Encrypted Connections, go to: https://msdn.microsoft.com/en-ca/library/ms191192(v=sql.120).aspx

8.3 How to configure encryption keys for Profiles System data

This topic provides instructions to generate and install the encryption keys that Commerce Server uses to encrypt and decrypt sensitive profile information.

You use the Profile Key Manager to create and work with encryption keys. For more information about the Profile Key Manager, see Profile Key Manager.

To configure the encryption keys, follow these steps:

1. On one of the business management servers, generate the encryption keys. For more information about how to perform this step, see the procedure "To generate encryption keys".
2. Encrypt the encryption keys and store encrypted encryption keys in the registry. For more information about how to perform this step, see the procedure "To encrypt encryption keys".
3. Update the Web.config files to identify where in the registry the encrypted encryption keys are stored. For more information about how to perform this step, see the procedures "To update the Web.config file for the profiles Web service" and "To update the Web.config file for the application".
4. Repeat steps 2 and 3 on the other business management servers.
5. Repeat steps 2 and 3 on every Web server.

The following procedures contain detailed instructions to complete each of these tasks.

Note:

You must use the same encryption keys that you create in step 1 on all the servers.

You must perform steps 2 and 3 on each server. You cannot follow these steps only one time and copy the resulting files to the other servers.

To generate encryption keys

1. Log on to the computer as an administrator.
2. Open a Command Prompt and navigate to C:\Program Files (x86)\Commerce Server 11\Tools\
3. At the command prompt, type the following command:

   ProfileKeyManager.exe /kn /o keys.xml

   This command generates a file that is named keys.xml.

To encrypt encryption keys

1. Log on to the computer as an administrator.
2. Copy the keys.xml file that you generated by using the procedure "To generate encryption keys" to the local computer.
3. Open a Command Prompt and navigate to C:\Program Files (x86)\Commerce Server 11\Tools\
4. To encrypt the encryption keys and store the encrypted encryption keys in the registry, type the following command at the command prompt:

   ProfileKeyManager.exe /ke /kf keys.xml /o RegKeys.xml

5. If you have another copy of the keys.xml file, delete the keys.xml file that you copied in step 1.

   Note:

   The keys.xml file contains the encryption keys in plain text format. By using these keys, someone could decrypt sensitive profile information. You will have to keep one copy of the keys.xml file so that you can generate encrypted encryption keys for other servers. However, you should make sure that the keys.xml file is secure. You should delete the keys.xml file from every other computer.

To update the Web.config file for the profiles Web service

1. Edit the Web.config file for the profiles Web service. The default location of this file is C:\Inetpub\wwwroot\<SiteName>ProfilesWebService\Web.config, where <SiteName> is the name of the Commerce Server Web site.

2. Change the value of the publicKey attribute of the profilesWebService element to the value of the PublicKey element in the RegKeys.xml file. If the publicKey attribute does not exist, add it.
3. Change the value of the privateKey1 attribute of the profilesWebService element to the value of the PrivateKey element in the RegKeys.xml file. If the privateKey1 attribute does not exist, add it.
4. Change the value of the keyIndex attribute of the profilesWebService element to "1". If the keyIndex attribute does not exist, add it.

   A sample profilesWebService element will resemble this:

```
        <profilesWebService
            siteName="CSharpSite"
            authorizationPolicyPath="ProfilesAuthorizationStore.xml"
            disableAuthorization="false"

publicKey="registry:HKEY LOCAL MACHINE\SOFTWARE\CommerceServer\Keys\Default,PublicKey"

privateKey1="registry:HKEY LOCAL MACHINE\SOFTWARE\CommerceServer\Keys\Default,PrivateKey"
            keyIndex="1"
            searchResultsLimit="500"
            bypassProfileCache="true">
          ...
        </profilesWebService>
```

To update the Web.config file for the application

1. Edit the Web.config file for the Commerce Server Web application. The default location of this file is C:\Inetpub\wwwroot\<SiteName>\Web.config where <SiteName> is the name of the Commerce Server Web site.
2. Locate the add element that has a type attribute that has a value of "publicKey". Change the value of the value attribute of the add element to the value of the PublicKey element in the RegKeys.xml file.
3. Locate the add element that has a type attribute that has a value of "privateKey1". Change the value of the value attribute of the add element to the value of the PrivateKey element in the RegKeys.xml file.

   A sample keys element will resemble this:

```
        <keys keyIndex="1">
            <add type="publicKey" value="registry:HKEY LOCAL MACHINE\SOFTWARE\Commerce
Server\Keys\Default,PublicKey" />
            <add type="privateKey1" value="registry:HKEY_LOCAL_MACHINE\SOFTWARE\Commerce
Server\Keys\Default,PrivateKey " />
        </keys>
```

8.4 Creating accounts and groups

8.4.1 What are the required accounts and groups?

This topic summarizes the accounts and groups that you use to help secure a Commerce Server deployment. These accounts are required to run the various Commerce Server services, ASP.NET, and Web applications. Commerce Server creates some of these accounts when you configure a server. Other accounts require that you create them.

See the following sections for the account and group requirements for each of these areas:

- Commerce Server Installer and Staging User Accounts
- Commerce Server ASP.NET Account
- Commerce Server Windows Service Accounts and User Groups
- Commerce Server Web Application Accounts and User Groups
- Commerce Server Adapter and BizTalk Server Accounts and User Groups

Note the following:

- Commerce Server supports only <*domain name*>\<*user*> name formats for service accounts and Windows groups.
- We recommend that you use Active Directory domain groups and user accounts when you use multiple-computer configurations, including SQL Server. Domain groups include domain local groups, global groups, and universal groups, which are supported in both single-server and multiple-computer environments. You must manually create all the domain groups and accounts before you configure Commerce Server.

Commerce Server Installer and Staging User Accounts

The Commerce Server installer account, known as <*CS Installer*> in this deployment guide, must have the following rights to configure Commerce Server servers:

- Administrator rights on the local computer.
- SQL System Administrator rights on the computer that is running SQL Server.

Add the Commerce Server installer account to the Windows user groups indicated in the following table. This lets the installer access the Web services associated with these user groups.

Account name	Description	Windows User Group
<CS Installer>	Account of person logged on to install and configure Commerce Server.	Administrator, CatalogAdminGroup, MarketingAdminGroup, OrdersAdminGroup,ProfilesAdminGroup
<data domain>\Staging user>	Account of person who manages Commerce Server Staging.	Not applicable

Commerce Server Windows Service Accounts and User Groups

Each Commerce Server Windows service requires the definition of a Windows service account. The following table summarizes the default names that are used in this deployment guide.

Account name	Description
CSStageSvc	Account for running the Commerce Server Staging (CSS) service.

These three accounts must be created manually. The Commerce Server Configuration wizard configures Commerce Server to use these accounts specifically, but the Configuration wizard does not create these accounts.

Commerce Server Web Application Accounts and User Groups

You use Service user accounts for the Sitecore Commerce web applications to perform these tasks:

- To run Internet Information Services (IIS) application pools.
- To help secure folders.
- To establish anonymous access to the Web site.
- To access the Commerce Server databases.

Commerce Server creates the Web applications when you unpack a Commerce Server site, such as the SolutionStorefrontSite, and select the Web services that you want to install. Each Commerce Server Web application requires definition of a Windows user account and a Windows user group.

The following table summarizes the default names that are used in this deployment guide. You create these items and make assignments before or after you install Commerce Server. You create these accounts and user groups on the data tier domain controller. In addition, you create the RunTimeUser account on the Data tier domain controller.

Account name	Description
RunTimeUser	The account to run the Sitecore site application pool.
CatalogWebSvc	The account to run the Catalog web service application pool.
MarketingWebSvc	The account to run the Marketing web service application pool.
OrdersWebSvc	The account to run the Orders web service application pool.
ProfilesWebSvc	The account to run the Profiles web service application pool.

For each Web application, you create the associated administrative user groups and assign accounts as indicated in the following table. IIS automatically creates the IIS_IUSRS group.

Sitecore Commerce Web application default name	User account	User group
CatalogWebService	CatalogWebSvc	CatalogAdminGroup, IIS_IUSRS
MarketingWebService	MarketingWebSvc	MarketingAdminGroup, IIS_IUSRS
OrdersWebService	OrdersWebSvc	OrdersAdminGroup, IIS_IUSRS
ProfilesWebService	ProfilesWebSvc	ProfilesAdminGroup, IIS_IUSRS
<site_name>	RunTimeUser	IIS_IUSRS

Commerce Server Adapter and BizTalk Server Accounts and User Groups

Installing BizTalk Server creates the BizTalkAdmin and BizTalkSvc accounts. You must create the RunTimeUser and CSLOB accounts before you install Commerce Server. After installation, you create SQL Server login accounts and associate the user accounts with Windows user groups.

Account name	Description	Windows user group
CSLOB	Commerce Server adapters line-of-business service identity	No applicable

8.4.2 What are the required accounts and groups to create?

This topic provides a summary of the user accounts and groups that you create. In a multi-computer deployment, Sitecore Commerce accounts and user groups must be created on the domain controller. In a single-server deployment, you can create these accounts and groups on the computer where Sitecore Commerce is installed. For the internal test and development environments, create the same accounts and groups that you create for the data domain in the production environment.

See the following sections for the account and group requirements for each of these areas:

- Commerce Server and Service Accounts
- Commerce Server Groups and Account Assignments
- Additional User Groups for Granular Security
- SQL Server Database Instances, Accounts, and Role User Mappings

Commerce Server User and Service Accounts

The following table lists the accounts that you create or that are created when you install prerequisite software. You must create the <CS Installer>, <CS Staging User>, CSLOB, and RunTimeUser accounts before you install Commerce Server. Post-installation, you create SQL Server Login accounts and associate the user accounts together with Windows user groups.

Account name	Description	Windows user group	SQL Server login account
<CS Installer>	Account of person logged on to install and configure Commerce Server.	Administrator, CatalogAdminGroup, MarketingAdminGroup, OrdersAdminGroup, ProfilesAdminGroup	not applicable
<CS Staging User>	Account of person who manages Commerce Server Staging.	not applicable	<CS Staging User>
CSLOB	Commerce Server Adapters identity.	not applicable	not applicable
CSStageSvc	Commerce Server Staging (CSS) service identity.	CSS_SG, CSS Administrators, CSS Operators	CSStageSvc

Commerce Server Groups and Account Assignments

Commerce Server Administrator Groups

Create the four administrator groups summarized in the following table. These represent the minimum number of groups to define. You should create distinct user groups based on your business needs. You then assign these groups to authorization roles through the Authorization Manager. For more information, see Authorizing Users and Groups to Access Web Services.

User group	Description	Accounts to assign
CatalogAdminGroup	Administrator group for the Catalog and Inventory Web services.	<CS Installer>, Business User Accounts
MarketingAdminGroup	Administrator group for the Marketing Web services.	<CS Installer>, Business User Accounts
OrdersAdminGroup	Administrator group for the Orders Web services.	<CS Installer>, Business User Accounts
ProfilesAdminGroup	Administrator group for the Profiles Web services.	<CS Installer>, Business User Accounts

For a production deployment, you will want to define more groups in order to take full advantage of the role assignment roles available. For descriptions about each predefined role, see the next section.

Commerce Server Web Services and Account and Application Pool Assignments

Commerce Server installs the Web services when you unpack a site, and selects the Web services to install. Each Commerce Server Web service requires definition of a Windows user account, Windows user group, SQL Server login account, and application pool. The following table summarizes the default names Commerce Server and the installation guide use. You create the Windows user accounts before you unpack a site, and you create the SQL Server login accounts and application pools after you unpack the site.

Commerce Server Web service	Default name	Windows/SQL Login account	Windows user group	Application pool
Catalog	CatalogWebService	CatalogWebSvc	CatalogAdminGroup, IIS_IUSRS	CatalogWebSvcAppPool
Marketing	MarketingWebService	MarketingWebSvc	MarketingAdminGroup, IIS_IUSRS	MarketingWebSvcAppPool
Orders	OrdersWebService	OrdersWebSvc	OrdersAdminGroup, IIS_IUSRS	OrdersWebSvcAppPool
Profiles	ProfilesWebService	ProfilesWebSvc	ProfilesAdminGroup, IIS_IUSRS	ProfilesWebSvcAppPool

For each site that you unpack, we recommend that you create unique Web service account names, SQL Server login account names, Windows user groups, and application pools. You can share application pools, but we do not recommend this action.

Web Service Administrator Role Assignments

The following table lists the Web services and their corresponding authorization stores and administrator roles. You must assign each Web service account to its corresponding authorization role.

Authorization Store	Role	Account Assignments
CatalogAuthorizationStore.xml	Administrator	CatalogWebSvc, <CS Installer>
MarketingAuthorizationStore.xml	MarketingAdministrator	MarketingWebSvc, <CS Installer>
OrdersAuthorizationStore.xml	OrdersAdministrator	OrdersWebSvc, <CS Installer>

Authorization Store	Role	Account Assignments
ProfilesAuthorizationStore.xml	ProfileAdministrator	ProfilesWebSvc, <CS Installer>

After you assign write permissions to the authorization stores, in order to perform any operation in the Business Management applications, you assign users to the administrator roles for each Web service. By adding <CS Installer> to each administrator role, you can open and use each Business Management application.

BizTalk Adapters Role Assignments

The following table lists the role assignments to which CSLOB, the BizTalk adapters identity, must be added.

Authorization Store	Role	Description
CatalogAuthorizationStore	CatalogAdministrator	Gives the catalog adapter permission to import catalog changes and export catalogs.
MarketingAuthorizationStore	InventoryAdministrator	Gives the inventory adapter permission to import inventory catalog changes and export inventory catalogs.
OrdersAuthorizationStore	OrdersAdapter	Enables the orders adapter to perform all basic functions, such as Update Purchase Order, Save Purchase Order, Accept Basket, Orders Query, and Orders Export.
ProfilesAuthorizationStore	UserObject, ProfileWriter_Adapter	Enables the profiles adapter to update profile objects when it uses the following operations: Profile Delete, Profile Update, Profile Import, Profile Query, and Profile Export.

Additional User Groups for Granular Security

The following sections summarize the various authorization roles that are predefined for the Commerce Server systems. For each authorization role of interest, create an associated user group on the domain controller. You can then add business user accounts to the user group.

- Catalog and Inventory Systems
- Marketing System
- Orders System
- Profiles System

For each user group you create, you must assign the groups to authorization roles through the Authorization Manager. For more information, see Authorizing Users and Groups to Access Web Services.

Catalog and Inventory Systems

The following table describes the predefined authorization roles for the Catalog System and the Inventory System.

Role	Description
CatalogAdministrator	Members can manage the Catalog System.
CatalogViewer	Members have read access to the Catalog System.
CatalogManager	Members can manage all the catalogs in the Catalog System.
SchemaManager	Members can manage the catalog and inventory schema, including property, category, and product definitions.
CatalogSetsAdministrator	Members can manage all the catalog sets.
CatalogSetsViewer	Members can view all the catalog sets in the Catalog System.
InventoryAdministrator	Members can manage the Inventory System.
InventoryViewer	Members can view all the catalogs in the Inventory System.
InventorySynchronizationManager	Members can synchronize the run-time Inventory System with the management system.
Administrator	Members can manage the Catalog System and the Inventory System.

Marketing System

The following table describes the predefined authorization roles for the Marketing System.

Role	Description
MarketingAdministrator	Members have full access to every operation in the Marketing System.
MarketingApprover	Members can approve or reject marketing items, such as campaigns, discounts, and expressions.
MarketingAuthor	Members can create marketing-related items, including customers, campaigns, discounts, and expressions.
MarketingViewer	Members can view and search marketing items, including campaign event logs.
GlobalExpressionAuthor	Members can create, edit, and delete global expressions across multiple discounts.
RuntimeSiteManager	Members can refresh the Discounts and Advertisements caches of the run-time site.

Orders System

The following table describes the predefined authorization roles for the Orders System.

Role	Description
OrdersAdministrator	Members can manage data integrity and cleanup issues.
OrdersConfigurationEditor	Members can manage orders configuration data for the site.
OrdersViewer	Members have read access to view orders.
OrdersAdapter	Members can search orders for order processing and updates.

Profiles System

The following table describes the predefined authorization roles for the Profiles System.

Role	Description
ProfileAdministrator	Members have complete access to the Profiles System.
ProfileWriter_BusinessManager	Members of this scope-level role have access to the profile definition within the scope. There are six profile definitions: UserObject, Address, Organization, BlanketPO, CreditCard, and Currency.
ProfileWriter_CSR	Members of this scope-level role have access to the profile definition within the scope.
ProfileWriter_Adapter	Members of this scope-level role have access to the profile definition within the scope.

Users of the scope-level roles have access only to the profile type within the scope name. For example, members of the ProfileWriter_BusinessManager role in the UserObject scope have access to the UserObject profile definition only. You must add users to each scope-level role individually.

SQL Server Database Instances, Accounts, and Role User Mappings

SQL Server Database Instances Created for Commerce Server

The following table summarizes the Commerce Server databases and default database names that Commerce Server and the installation guide use.

Commerce Server SQL database instance	Default database name	How the database is created
CS Administration	MSCS_Admin	Created by the Commerce Server Configuration wizard.
CS Catalog Scratch	MSCS_CatalogScratch	Created by unpacking the catalog site resource.
Site Catalog	<site_name>_productcatalog	Created when you unpack the site resource.
Site Marketing	<site_name>_marketing	Created when you unpack the site resource.
Site Marketing List	<site_name>_marketing_lists	Created when you unpack the site resource.

Commerce Server SQL database instance	Default database name	How the database is created
Site Profiles	<site_name>_profiles	Created when you unpack the site resource.
Site Transaction Configuration	<site_name>_transactionconfig	Created when you unpack the site resource.
Site Transactions	<site_name>_transactions	Created when you unpack the site resource.

SQL Database Account, Database, and Database Role User Mapping

The following table lists the accounts on the computers that are running SQL Server that you must add to the specified roles. By default, the database names start with StarterSite. However, you might have specified different database names when you unpacked your site.

Database Account	Database	Roles
CatalogWebSvc	MSCS_Admin	admin_reader_role, admin_cache_poller_role
	MSCS_CatalogScratch	db_datareader, db_datawriter, db_ddladmin
	StarterSite_ProductCatalog	ctlg_CatalogWriterRole, db_datareader, db_datawriter, db_ddladmin, db_securityadmin, Inventory_ReaderRole, Inventory_WriterRole
MarketingWebSvc	MSCS_Admin	admin_reader_role, admin_cache_poller_role
	StarterSite_Marketing	mktg_MarketingService_role, mktg_promoCodeGenerator_role
	StarterSite_MarketingLists	db_owner
	StarterSite_ProductCatalog	ctlg_catalogReaderRole
	StarterSite_Profiles	Profile_Reader, Profile_Schema_Reader, ctlg_catalogReaderRole
OrdersWebSvc	MSCS_Admin	admin_reader_role, admin_cache_poller_role
	MSCS_CatalogScratch	db_datareader, db_datawriter, db_ddladmin
	StarterSite_Marketing	mktg_runtime_role
	StarterSite_ProductCatalog	ctlg_catalogReaderRole, Inventory_ReaderRole
	StarterSite_Profiles	Profile_Reader, Profile_Schema_Reader
	StarterSite_TransactionConfig	Orders_Management
	StarterSite_Transactions	Orders_Management, Orders_Runtime

Database Account	Database	Roles
ProfilesWebSvc	MSCS_Admin	admin_reader_role, admin_cache_poller_role
	StarterSite_Profiles	Profile_Schema_Manager, Profile_Runtime, ctlg_CatalogWriterRole
RunTimeUser	MSCS_Admin	admin_reader_role, admin_cache_poller_role
	MSCS_CatalogScratch	db_datareader, db_datawriter, db_ddladmin
	StarterSite_Marketing	mktg_runtime_role
	StarterSite_MarketingLists	db_datareader
	StarterSite_ProductCatalog	ctlg_catalogReaderRole, ctlg_CatalogWriterRole, Inventory_RuntimeRole, db_ddladmin, db_securityadmin, Inventory_ReaderRole, Inventory_WriterRole, db_datareader, db_datawriter
	StarterSite_Profiles	Profile_Schema_Reader, Profile_Runtime
	StarterSite_TransactionConfig	Orders_Runtime
	StarterSite_Transactions	Orders_Runtime
CSStageSvc	MSCS_Admin	admin_reader_role
	MSCS_CatalogScratch	db_datareader, db_datawriter, db_ddladmin
	StarterSite_Marketing	db_ddladmin, mktg_staging_role
	StarterSite_MarketingLists	db_datareader
	StarterSite_ProductCatalog	ctlg_CatalogWriterRole, db_datareader, db_datawriter, db_ddladmin, db_securityadmin, Inventory_ReaderRole, Inventory_WriterRole
	StarterSite_Profiles	Profile_Schema_Manager
	StarterSite_TransactionConfig	Orders_Management
<CS Staging User>	MSCS_Admin	db_datareader
	MSCS_CatalogScratch	db_datareader, db_datawriter, db_ddladmin
	StarterSite_ProductCatalog	ctlg_CatalogWriterRole, Inventory_ReaderRoles

8.5 Setting file and folder access permissions

This section provides instructions on how to help secure files and folders within a Sitecore Commerce deployment by setting access permissions. The following assignments must be made on the computers where Sitecore Commerce Web applications are run.

The specific file and folder access permissions you assign depend on whether you are configuring a Business Management server or a production Web server.

Perform the following steps on the Business Management server in a deployment:

Setting these access permissions are required to allow business users access to the Web services.

1. For the Catalog role assignment file, assign write permissions to the Catalog Web service account. These permissions must be made on the computer where Commerce Server Web applications are run. see How to Assign Write Permissions to the Catalog Authorization Role.
2. For the Temporary ASP.NET folder, assign write permissions to the Web application accounts. See How to Assign Write Permissions to the Temporary ASP.NET Folder.
3. For the Windows Temporary folder, assign access permissions to all Web application accounts. see How to Assign Permissions for the Windows Temporary Folder.

Follow these steps on each production Web server in a deployment:

1. For the Temporary ASP.NET folder, assign write permissions to the Web site application account, which is the RunTimeUser account. See How to Assign Write Permissions to the Temporary ASP.NET Folder.
2. For the Windows Temporary folder, assign access permissions to the Web site application account, which is the RunTimeUser account. See How to Assign Permissions for the Windows Temporary Folder.

8.5.1 How to Assign Write Permissions to the Catalog Authorization Role

You must follow these steps before your users can access the Catalog Web service. You assign write permission for the catalog Web service account, CatalogWebSvc, that you created as explained in the What are the required accounts and groups to create? topic. You assign write permissions so that business users can access catalogs through the Business Management applications.

To assign write permissions to the catalog authorization role

1. Click Start, click Run, type explorer, and then click OK.
2. In Windows Explorer, move to the directory where you installed the Catalog Web service. For example, <drive:>\Inetpub\wwwroot\CatalogWebService\.
3. In the <Catalog Web service name> folder, right-click the XML file CatalogAuthorizationStore.xml, and then click Properties.
4. In the CatalogAuthorizationStore.xml Properties dialog box, on the Security tab, click Add.
5. In the Select Users, Computers, or Groups dialog box, in the Enter the object names to select text box, type <Domain or Computer name>\CatalogWebSvc, and then click OK. As appropriate, substitute the user name for the account you created for "CatalogWebSvc".
6. In the CatalogAuthorizationStore.xml Properties dialog box, in the Group or user names list, select the CatalogWebSvc account.
7. In the Permissions for CatalogWebSvc list, select Write in the Allow column. (Read & Execute and Read should already be selected in the Allow column.) Click OK.

8.5.2 How to Assign Write Permissions to the Temporary ASP.NET Folder

Follow these steps to assign write permissions to the Temporary ASP.NET folder. You assign write permissions for the RunTimeUser and all Web service accounts that you created and as specified in What Are the Accounts and Groups to Create? You will be unable to run the Business Management applications if you omit these steps.

To assign write permissions to the Temporary ASP.NET folder

1. Click Start, click Run, type explorer, and then click OK.
2. In Windows Explorer, move to the <drive:>\Windows\Microsoft.NET\Framework64\v<version> folder.
3. Right-click Temporary ASP.NET Files, and then click Properties.
4. In the Temporary ASP.NET Files Properties dialog box, on the Security tab, click Add.
5. In the Enter the object names to select text box, type <Domain or Computer name>\RunTimeUser, and then click OK. As appropriate, substitute the user name for the anonymous user account you created for "RunTimeUser".
6. In the Temporary ASP.NET Files Properties dialog box, in the Groups or user names list, select the RunTimeUser account.
7. In the Permissions for RunTimeUser list, select Write in the Allow column. (Read & Execute and Read should already be selected in the Allow column Click OK.
8. In the Security dialog box, click Yes.
9. Repeat steps 3 through 8 for the Web service accounts: CatalogWebSvc, MarketingWebSvc, OrdersWebSvc, and ProfilesWebSvc.

8.5.3 How to Assign Permissions for the Windows Temporary Folder

Follow these steps to assign write permissions for the Windows Temporary folder on computers hosting Commerce Server Web services. You assign write permissions for the RunTimeUser and all Web service accounts that you created as explained in the What are the required accounts and groups to create? topic. You will be unable to run the Business User applications that are hosted by the Business Management server if you omit these steps.

To assign permissions for the Windows Temporary folder

1. Click Start, click Run, type explorer, and then click OK.
2. In Windows Explorer, move to the <drive:>\Windows folder.
3. Right-click Temp, and then click Properties.
4. In the Temp Properties dialog box, on the Security tab, click Add.
5. In the Enter the object names to select text box, type <Domain or Computer name>\RunTimeUser, and then click OK. As appropriate, substitute the user name for the anonymous account you created for "RunTimeUser".
6. In the Temp Properties dialog box, in the Groups or user names list, select the RunTimeUser account.
7. In the Permissions for RunTimeUser list, select Write in the Allow column. (Read & Execute, List Folder Contents, and Read should already be selected in the Allow column.) Click OK.
8. In the Security dialog box, click Yes.
9. Repeat steps 3 through 8 for the CatalogWebSvc, MarketingWebSvc, OrdersWebSvc, and ProfilesWebSvc accounts.

8.6 Authorizing users and groups to access web services

Commerce Server provides several predefined roles to which you assign business users so that they can perform specific tasks such as editing a catalog, creating a discount, and deleting an order. Authorization Manager, which is a Windows Server security tool, provides a role-based security model that you use to set permissions. With role-based access control, you can set permissions according to the organizational structure of your company. For more information about Authorization Manager, see http://go.microsoft.com/fwlink/?LinkId=16923 .

When you assign user accounts or groups to roles such as MarketingAdministrator or OrdersAdministrator, you enable users to perform any operation associated with the corresponding Commerce Server system. In addition, the Commerce Server Adapters service account, CSLOB, requires authorization role assignments.

Follow these steps to authorize accounts and groups to access Web services:

1. Create the group that you want to assign to an authorization role. You can assign both Windows or Active Directory domain accounts and groups to the authorization roles. See the following topics as needed:
    - For a summary of the predefined roles, see the "Additional User Groups for Granular Security" section in What are the required accounts and groups to create?
2. Add one or more users to the group you created in step 1. See How to Add Business User Accounts to Active Directory Groups.
3. On the computer where the Web services are run, assign users or groups to the authorization roles by using Authorization Manager. See How to Add Users or Groups to Authorization Roles
4. If you are using Commerce Server Adapters for BizTalk Server, assign the CSLOB service account to its required authorization roles by using Authorization Manager. See How to Set Authorization Roles for the BizTalk Adapters.

8.6.1 What Are the Minimum Authorization Roles to Assign?

Only authorized users can access Business Management applications. You can simplify the administration of authorization role assignment by creating an administrator group for each Commerce Server system, and authorizing that group according to the corresponding administrator roles.

The following table lists the Web services and their corresponding administrator roles. These represent the minimum number of roles to assign and the groups to authorize. You assign business users to one or more of the administrative groups based on the Business Application they need to access.

Web service	Authorization	Windows group	Business Management application
Catalog Web service	Administrator	CatalogAdminGroup	• Catalog Manager • Catalog and Inventory Schema Manager
Marketing Web service	MarketingAdministrator	MarketingAdminGroup	Marketing Manager
Orders Web service	OrdersAdministrator	OrdersAdminGroup	Customer and Orders Manager
Profiles Web service	ProfileAdministrator	ProfilesAdminGroup	Customer and Orders Manager

For more information about authorization roles, see What are the required accounts and groups to create?.

8.6.2 What Are the Required Authorization Store Permissions for BizTalk Adapters?

Additional authorization store permissions are necessary for the Commerce Server adapters line-of-business service identity, CSLOB. The following table lists the role assignments to which CSLOB must be added.

Authorization store	Role assignment	Description
CatalogAuthorizationStore	CatalogAdministrator	This role gives the catalog adapter permission to import catalog changes and lets the adapter export catalogs.
	InventoryAdministrator	This role gives the inventory adapter permission to import inventory catalog changes and lets the adapter export inventory catalogs.
OrdersAuthorizationStore	OrdersAdapter	This role lets the orders adapter perform all of its basic functionality, such as Update Purchase Order, Save Purchase Order, Accept Basket, Orders Query, and Orders Export.

ProfilesAuthorizationStore	• UserObject
	  ○ ProfileWriter_Adapter
	• Address
	  ○ ProfileWriter_Adapter
	• Organization
	  ○ ProfileWriter_Adapter
	• BlanketPO
	  ○ ProfileWriter_Adapter
	• CreditCard
	  ○ ProfileWriter_Adapter
	• Currency
	  ○ ProfileWriter_Adapter
		These roles let the Profiles adapter update to specific profile types with the following operations: Profile Delete, Profile Update, Profile Import (for example, Create), Profile Query, and Profile Export. The granularity of the Profile Type roles for User Object, Address, Organization, and so on, lets you manage BizTalk receive and send endpoints for specific types of profiles.

8.6.3 How to Add Users or Groups to Authorization Roles

You use the Authorization Manager to add individual users or user groups to a role.

Various roles exist for the Commerce Server business management features. You can simplify management of the roles by giving appropriate permissions to Windows groups and adding individual users to those groups. For a list of the groups needed to use the Commerce Server business management features, see What Are the Accounts and Groups to Create?

The following procedure describes how to add business users or groups to the authorization policy roles. Perform this procedure on computers where the Commerce Server Web services are run.

To add users to the authorization roles

1. Click Start, click Run, type azman.msc, and then click OK.
2. In the Authorization Manager screen, right-click Authorization Manager, and then click Open Authorization Store.
3. In the Open Authorization Store dialog box, verify that the XML file option is selected, and then click Browse to locate the authorization policy for the Web service. For example, the catalog authorization policy XML file is located at <drive:>\Inetpub\Wwwroot\CatalogWebService.
4. Select <authorization policy name>.xml, and then click Open.
5. In the Open Authorization Store dialog box, click OK.
6. Expand the authorization policy to \<authorization policy name>.xml\<System name>\Role Assignments\<Role name>. For example, \CatalogAuthorizationStore.xml\CatalogandInventorySystem\Role Assignments\CatalogAdministrator.
7. Right-click <Role name>, and then click Assign Windows Users and Groups.
8. In the Select Users, Computers, or Groups dialog box, in the Enter the object names to select box, type the name of the catalog administrator Windows group that you defined in What Are the Accounts and Groups to Create? For example, CatalogAdminGroup. Alternatively, you can type the name of an individual business user account. Click OK.
9. Repeat steps 2 through 8 for each authorization policy role in each Web service authorization store.

8.6.4 How to Set Authorization Roles for the BizTalk Adapters

The following procedure describes how to assign the Commerce Server Adapters service account, CSLOB, to its required authorization roles by using Authorization Manager. Perform this procedure on computers where the Commerce Server Web services are run.

To add the CSLOB account to the required authorization roles

1. Click Start, click Run, type azman.msc, and then click OK. If CatalogAuthorizationStore.xml, OrdersAuthorizationStore.xml, and ProfilesAuthorizationStore.xml appear in the left pane, go to step 6.
2. In the Authorization Manager screen, right-click Authorization Manager, and then click Open Authorization Store.
3. In the Open Authorization Store dialog box, verify that the XML file option is selected, and then click Browse to locate the role assignments for the Web service. For example, the catalog authorization policy XML file is located at <drive:>\Inetpub\wwwroot\CatalogWebService.
4. Select <authorization policy name>.xml, and then click Open.
5. In the Open Authorization Store dialog box, click OK.
6. Set the Catalog System authorization store role permissions:
    a. Expand the role assignments for CatalogAuthorizationStore.xml. For example, \CatalogAuthorizationStore.xml\CatalogandInventorySystem\Role Assignments.
    b. Right-click CatalogAdministrator, and then click Assign Windows Users and Groups.
    c. In the Select Users, Computers, or Groups dialog box, in the Enter the object names to select box, type CSLOB or the name that you created for the Commerce Server Adapters service account. Click OK.
    d. Repeat steps b and c for the InventoryAdministrator role.
7. Set the Orders System authorization store role permissions:
    a. Expand the role assignments for OrdersAuthorizationStore.xml. For example, \OrdersAuthorizationStore.xml\OrdersSystem\Role Assignments.
    b. Right-click OrdersAdapter, and then click Assign Windows Users and Groups.
    c. In the Select Users, Computers, or Groups dialog box, in the Enter the object names to select box, type CSLOB or the name that you created for the Commerce Server Adapters service account. Click OK.

8. Set the Profiles_Adapter role permissions for the UserObject, Address, Organization, BlanketPO, CreditCard, and Currency profile definitions in the Profiles System authorization store:
    a. Expand the role assignments for ProfilesAuthorizationStore.xml. For example, \ProfilesAuthorizationStore.xml\ProfilesSystem\<profile definition>\Role Assignments.
    b. Right-click ProfileAdministrator, and then click Assign Windows Users and Groups.
    c. In the Select Users, Computers, or Groups dialog box, in the Enter the object names to select box, type CSLOB or the name that you created for the Commerce Server Adapters service account. Click OK.
    d. Repeat steps b and c for each profile definition in the Profiles System authorization store.

8.7 Granting access to the Commerce Server Databases

This section provides instructions for configuring the database roles on the computers that are running SQL Server.

The database SQL login accounts that you create must match the same account names that you created for the Web application and Commerce Server service accounts that you created or that Commerce Server created. For the list of services and default names, see What are the required accounts and groups? .

Follow these steps to grant Commerce Server accounts access to the Commerce Server databases. All tasks are to be performed on computers that are running SQL Server.

1. On the production database server:
   a. Create SQL login accounts for the following:

      <data domain name>\RunTimeUser

      <Web domain name>\RunTimeUser

      <data domain name>\CatalogWebSvc

      <data domain name>\MarketingWebSvc

      <data domain name>\OrdersWebSvc

      <data domain name>\ProfilesWebSvc

      <data domain name>\CSStageSvc

   b. For each account that you created in Step 1a, associate the database accounts together with the database roles. For the required mappings, see What are the required accounts and groups?.
2. On the Staging database server, create SQL login accounts for the following:
   a. Create SQL login accounts for the following:

      <internal domain name>\CatalogWebSvc

      <internal domain name>\MarketingWebSvc

      <internal domain name>\OrdersWebSvc

      <internal domain name>\ProfilesWebSvc

   b. For each account that you created in Step 2a, associate the database accounts together with the database roles. For the required mappings, see What are the required accounts and groups? .

8.8 Securing Merchandising Manager and the Sitecore Content Tree

Commerce Server data can be edited in Sitecore via the Content Tree and the Merchandising Manager tool. You can control editing, creation, and deletion of these items with the normal Sitecore security mechanisms. You can find full details on these security mechanisms and the custom commerce security roles in Sitecore at http://commercesdn.sitecore.net/SCpbCS80/SitecoreCommerceConnectGuide/en-us/#Concepts/c_AuthorizationFundamentals.html

8.9 Granting access to Sitecore databases

The permissions required for the application pool user of the Sitecore site (i.e. RunTimeUser) can be found in the Sitecore installation guide, see https://dev.sitecore.net/Downloads/Sitecore_Experience_Platform/8_0/Sitecore_Experience_Platform_8_0.aspx .

8.10 Granting access to Microsoft Dynamics AX databases

In order for the Transaction service to talk to the AOS server via the Real Time Service you will need to secure the connection by using the settings listed in the following article http://blogs.msdn.com/b/axsupport/archive/2012/12/18/ax-for-retail-2012-r2-installing-the-real-time-service.aspx

Chapter 9 Appendices

9.1 Firewall Ports

Product & Functionality	Comments	Port Range
Sitecore		
-Website		80/443
-Reporting Service		80/443
Solr	https://cwiki.apache.org/confluence/display/solr/Running+Solr	
Default		8983
Commerce Server		
Staging	Commerce Server Staging (CSS). CSS uses this port to deploy site updates (such as Web content and business data) between different servers.	507
MSDTC	Microsoft Distributed Transaction Coordinator (MSDTC). The base for MSDTC is on the OLE Transactions interface protocol. This provides a simple, object-oriented interface to initiate and control transactions.	5000-5030
Business Web Services	Used by BizTalk and Desktop Business tools to access Commerce Server data. This port is changeable in IIS.	80
SQL Server	Full details can be found at http://support.microsoft.com/en-us/kb/287932	
Default	The default SQL Server port is 1433, and client ports are assigned a random value between 1024 and 5000.	1433, 1024-5000
Microsoft Dynamics AX	Full details can be found at https://technet.microsoft.com/en-us/library/gg731780.aspx	
Default	Strongly suggest going through the link above as there are a lot of entry points.	2712, 8101, 8201, 4712, 1433, 16750, 1239
BizTalk	Full details can be found at https://msdn.microsoft.com/en-us/library/aa577684.aspx	
Adapters	This completely depends on the adapters, please review the previously mentioned link.	
Administration Server		50000-50200, 1433, 135, 1164
Tracking Server		5000-5020, 1433, 135, 2383, 2725

Product & Functionality	Comments	Port Range
Processing Servers		5000-5020, 1433, 135, 2393, 2725, 445
Enterprise Single Sign-On		5000-5020, 1433, 135
BAM Portal		5000-5020, 1433, 135, 80/443, 2393, 2725,
MongoDB	Full details can be found at http://docs.mongodb.org/manual/reference/default-mongodb-port/	
Mongod & mongos	The default port for mongod and mongos instances. You can change this port with port or --port.	27017
Shard Server	The default port when running with --shardsvr runtime operatio,n or the shardsvr value for the clusterRole setting in a configuration file.	27018
Config server	The default port when running with --configsvr runtime operation or the configsvr value for the clusterRole setting in a configuration file.	27019
Web status page	The default port for the web status page. The web status page is always accessible at a port number that is 1000 greater than the port determined by port.	28017
Active Directory	Full details can be found at https://technet.microsoft.com/en-us/library/dd772723%28v=ws.10%29.aspx	
-Default		49152-65535

9.2 Download locations

Product	Package
Sitecore XP	
Solr support package	Sitecore.Solr.Support 1.*.* rev. ******.zip
Commerce Connect	Sitecore Commerce Connect 8.* rev. ******.zip
Commerce Server	
Commerce Server Core	CommerceServer-11.*.*.*.exe
Desktop Business Tools	BusinessTools-11.*.*.*.exe
BizTalk Adapters	BTSAdapters-11.*.*.*.exe
Sitecore Commerce powered by Commerce Server	
Sitecore Commerce Server Connect	Sitecore Commerce Server Connect 8.* rev. 8.*.*.*.update
Sitecore Merchandising Manager	Sitecore Merchandising Manager 8.0 rev. 8.*.*.*.update
Sitecore Commerce powered by Commerce Server – Sample Site	
Sitecore Commerce Server Connect Sample MVC Site	Sitecore Commerce Server Connect Sample MVC Site.8.*.*.*.update
MVC Site Deployment Scripts	Deploy.zip
Sitecore Commerce powered by Microsoft Dynamics	
Sitecore.Commerce.DynamicsIntegration.Connect	Sitecore.Commerce.DynamicsIntegration.Connectupdate
Sitecore.Commerce.DynamicsIntegration.Routing.	Sitecore.Commerce.DynamicsIntegration.Routing.update
Sitecore.Commerce.RoutingFramework.	Sitecore.Commerce.RoutingFramework.update
Sitecore.Commerce.DynamicsIntegration.TransactionService	Sitecore.Commerce.DynamicsIntegration.TransactionService.update
Sitecore Commerce powered by Microsoft Dynamics- SDK	
Routing.Framework SDK	Routing.Framework.Sdk.8.*.*.*.zip
DynamicsRetail.Routing SDK	Sitecore.Commerce.DynamicsIntegration.Routing.update.zip
TransactionService SDK	TransactionService.8.*.*.*.zip

Product	Package
Sitecore Commerce powered by Microsoft Dynamics AX – Sample Site	
Sitecore Reference Storefront Powered by Microsoft Dynamics	Sitecore.Reference.Storefront.Powered.by.Microsoft.Dynamics.8.*.*.*.update
ContosoImages	ContosoImages-1.update
Sitecore Commerce Powered By Microsoft Dynamics Deploy	Deploy.zip
MongoDB (v2.6.1)	http://www.mongodb.org/downloads#previous