



INSTALLATION / QUICK START

SITECORE E-COMMERCE DONE RIGHT. ➔ ACTIVECOMMERCE.COM





Active Commerce Installation

And Developer Quick Start

[Prerequisites](#)

[Installation](#)

[Troubleshooting](#)

[Creating a New Site](#)



Prerequisites

- [Sitecore CMS 7.2 and DMS 7.2 \(rev. 151021\)](#) *
or
[Sitecore CMS 7.5 \(rev. 150212\)](#)
or
[Sitecore XP 8.0 \(rev. 151127\)](#) *
or
[Sitecore XP 8.1 \(rev. 160302\)](#)
- [.NET Framework 4.5](#)

Please reference the applicable Sitecore Installation Guide for details on Sitecore installation requirements.

* Earlier revisions are also supported, but we strongly recommend utilizing the latest update.

Installation

1. If creating a new site:
 - a. Use the Sitecore installer as normal to install new instance of Sitecore.
 - b. Install DMS (Sitecore 7.2 **only**)
 - i. Follow Sitecore instructions to attach DMS databases, install Analytics configuration files, and configure analytics connection string.
2. If adding on to an existing site:
 - a. Ensure IIS app pool is set to .NET Framework v4.0 and all prerequisites are met.
3. Place the supplied Active Commerce license file, "license_activecommerce.xml", in the Data folder of your site.
4. In Sitecore, use the Installation Wizard to install the following modules, in this order:
 - a. Sitecore E-Commerce Services (version 2.2)
 - i. [Download from the Sitecore Developer Network](#)
 - ii. **NOTE: Though Sitecore does not support SES 2.2 on Sitecore 7.5 and 8.x, and has released open source versions for those versions, you should still install 2.2 for Sitecore 7.5 and 8.x. Active Commerce continues to support SES 2.2 on these versions of Sitecore.**
 - iii. Install the main package (Sitecore E-Commerce Services 2.2 rev. xxxxxx.zip).
 - iv. Install the Order Manager package (Sitecore E-Commerce Order Manager 2.2 rev. xxxxxx.zip).
 - v. **NOTE: Do not install SES Examples package, and do not install the SES SQL database.**
 - b. Active Commerce



- i. Sitecore install package can be obtained from your Active Commerce representative.
 - ii. Install the main package (Active Commerce-SCXX 3.3 rev. xxxxxx.zip).
 - 1. **Be sure to install the version appropriate to your Sitecore version.**
 - c. If you will be utilizing the Mobile device layer and device-specific CSS, you may also want to install a device-sniffing module such as [Mobile Device Detector](#). Starting with Sitecore version 8.1, built-in [Device Detection](#) is an option.
 - i. **NOTE: If installing on Sitecore 8.x, be sure to install Active Commerce before installing the Mobile Device Detector or any other Update Packages.**
5. Modify Web.config
- a. **You can skip this step if you are installing the Sherpa demo site.**
 - b. Add to <appSettings> section (after last node):


```
<add key="nhibernate-logger"
value="ActiveCommerce.Data.Logging.SitecoreLoggerFactory,
ActiveCommerce.Data"/>
```
 - c. Add to <log4net> section (Sitecore 7.2, 7.5 and 8.0 only):
 (after last "appender" node)


```
<appender name="NHibernateLogFileAppender"
type="log4net.Appender.SitecoreLogFileAppender,
Sitecore.Logging">
  <file value="$(dataFolder)/logs/NHibernate.log.{date}.txt"
/>
  <appendToFile value="true" />
  <layout type="log4net.Layout.PatternLayout">
    <conversionPattern value="%4t %d{ABSOLUTE} %-5p %m%n" />
  </layout>
  <encoding value="utf-8" />
</appender>
```

 (after last "logger" node)


```
<logger name="NHibernate" additivity="false">
  <level value="WARN" />
  <appender-ref ref="NHibernateLogFileAppender" />
</logger>
<logger name="NHibernate.SQL" additivity="false">
  <level value="WARN" />
  <appender-ref ref="NHibernateLogFileAppender" />
</logger>
```
 - d. Add to <system.webServer><modules> section (after last node):


```
<add name="CassetteHttpModule"
type="Cassette.AspNet.CassetteHttpModule, Cassette.AspNet"
/>
```



- e. Add to <system.webServer><handlers> section:

(after "WebDAVRoot64" node)

```
<add verb="*" path="skinned_resource.ashx"
type="ActiveCommerce.Skinning.SkinningHandler,
ActiveCommerce.Kernel, Version=1.0.0.0, Culture=neutral"
name="SkinnedResourceHandler" resourceType="Unspecified" />
```

(after last node)

```
<add name="CassetteHttpHandler" verb="*" path="cassette.axd"
preCondition="integratedMode" allowPathInfo="true"
type="Cassette.Aspnet.CassetteHttpHandler, Cassette.Aspnet"
/>
```

- f. Add to <system.web><httpModules> section (after last node, Sitecore 7.2, 7.5 and 8.0 **only**):

```
<add name="CassetteHttpModule"
type="Cassette.Aspnet.CassetteHttpModule, Cassette.Aspnet"
/>
```

- g. Add to <system.web><httpHandlers> section (Sitecore 7.2, 7.5 and 8.0 **only**):
(before first node)

```
<add verb="*" path="skinned_resource.ashx"
type="ActiveCommerce.Skinning.SkinningHandler,
ActiveCommerce.Kernel, Version=1.0.0.0, Culture=neutral" />
```

(after last node)

```
<add verb="*" path="cassette.axd"
type="Cassette.Aspnet.CassetteHttpHandler, Cassette.Aspnet"
/>
```

- h. Add to <system.web> section (after last node):

```
<catching>
  <outputCacheSettings>
    <outputCacheProfiles>
      <add name="AC/Catalog/GetProductForCategory"
enabled="true" duration="86400" varyByParam="*"
varyByHeader="host" location="Any" />
      <add name="AC/Utility/Countries" enabled="true"
duration="86400" varyByParam="none" varyByHeader="host"
location="Any" />
      <add name="AC/Utility/States" enabled="true"
duration="86400" varyByParam="id" varyByHeader="host"
location="Any" />
      <add name="AC/Translation" enabled="true"
duration="86400" varyByParam="lang" varyByHeader="host"
location="Any" />
    </outputCacheProfiles>
  </outputCacheSettings>
```



```
</caching>
```

- i. Add to `<runtime><assemblyBinding>` section (after last node, Sitecore 7.5 and 8.x only):

```
<dependentAssembly>
  <assemblyIdentity name="Stimulsoft.Report.Web"
    publicKeyToken="ebe6666cba19647a"/>
  <bindingRedirect oldVersion="1.0.0.0-2013.1.1600.0"
    newVersion="2013.1.1600.0"/>
</dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="Stimulsoft.Base"
    publicKeyToken="ebe6666cba19647a"/>
  <bindingRedirect oldVersion="1.0.0.0-2013.1.1600.0"
    newVersion="2013.1.1600.0"/>
</dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="Stimulsoft.Report"
    publicKeyToken="ebe6666cba19647a"/>
  <bindingRedirect oldVersion="1.0.0.0-2013.1.1600.0"
    newVersion="2013.1.1600.0"/>
</dependentAssembly>
```

6. Run your site, log into Sitecore, and make sure you don't receive any errors.
7. Create the Active Commerce database. This database will house all non-Sitecore entities, such as orders, product stock, and wish lists.
 - a. Create a new database on your database server
 - i. Name it whatever you'd like (e.g., *MySite_ActiveCommerce*)
 - ii. You can give it the same owner as your Sitecore databases (e.g., *sitecore*)
 - b. Add a connection string to `ConnectionStrings.config`
 - i. Use "activecommerce" for the name
 - ii. Set `MultipleActiveResultSets=true` on the connection string
 - iii. e.g.

```
<add name="activecommerce" connectionString="user
id=sitecore;password=sitecore;Data
Source=localhost;Database=MySite_ActiveCommerce;MultipleActiveResult
Sets=true" />
```
 - c. Generate the database schema
 - i. Go to the following address: `http://<site
host>/sitecore/admin/databaseutility.aspx`
 - ii. Click the "Generate Schema" button
 - iii. You should see a "Success!" message when completed
 - d. **NOTE:** By default, Active Commerce assumes use of SQL Server 2012. If using a different provider or version, please reference the *Active Commerce Configuration Guide - Active Commerce Database* section for details on how



this can be changed.

See *Active Commerce Configuration Guide - Scaling Active Commerce for information about setting up Active Commerce in a distributed environment.*

Troubleshooting

For assistance with issues during installation, refer to the knowledge base on the Active Commerce Help Center: <https://help.activecommerce.com>.

Creating a New Site

1. In the Sitecore Content Editor, select the sitecore\Content item.
2. Click “Active Commerce Website,” which is now an insert option.
3. Enter the name of your site and click OK.
NOTE: The name entered here will be used throughout as a default name for various auto-generated items.
4. Create/modify the site definition.
 - a. Utilize the App_Config\Include\SiteDefinition.config.example template config provided by Sitecore as a starting point by removing the “.example” extension.
 - b. Add/adjust the following attributes:
 - i. **name**="**<site name>**" - Required. The site name / identifier.
 - ii. **rootPath**="**<path to site root>**" - Required. The path to the site root item (parent of the Home item).
 - iii. **hostName**="**<site host>**" - Required. The site host to use.
NOTE: Don't forget to ensure the hostName used here has a binding in IIS and also a hosts file entry if necessary.
 - iv. **targetHostName**="**<site host>**" - Optional. If you are using wildcard (*) or pipe (|) characters in your hostName, then you'll need to add the targetHostName.
 - v. **publicHostName**="**<site host>**" - Optional. If present, this is used when generating links exposed publicly (e.g., email templates, product export). Fallback of targetHostName is used if absent.
 - vi. **content**="**master**" - Required. Various features use the configured content database of the site definition directly. *NOTE: You can omit this attribute in Content Delivery environments, unless using legacy orders, in which case this will need to remain "master".*
 - vii. **EcommerceSiteSettings**="**/Site Settings**" - Required. Specifies the Ecommerce Settings item root. This can be a relative path (relative to your site), absolute item path, or a GUID. By default, this is named “Site Settings”.
 - viii. **translationDictionary**="**/sitecore/commerce/translation**" - Required. The full path of the translation dictionary root to use for the site. By default, this is installed at /sitecore/commerce/translation.



- ix. **skin=""<skin name>"** - Optional. Specifies the skin to be used by the site (see the *Active Commerce Skinning System Guide* for full details). If this is omitted, the base skin will be used.
5. Add on-publish HTML cache clearing for your new site to your configuration patch. Add the following inside the `<sitecore>` node and edit for the site name you configured above.

```

<events>
  <event name="publish:end">
    <handler type="Sitecore.Publishing.HtmlCacheClearer, Sitecore.Kernel"
method="ClearCache">
      <sites hint="list">
        <site id="[site name]">[site name]</site>
      </sites>
    </handler>
  </event>
  <event name="publish:end:remote">
    <handler type="Sitecore.Publishing.HtmlCacheClearer, Sitecore.Kernel"
method="ClearCache">
      <sites hint="list">
        <site id="[site name]">[site name]</site>
      </sites>
    </handler>
  </event>
</events>

```

6. Create a shop context for the Order Manager.
 - a. In the Sitecore Content Editor, select the `/sitecore/system/Modules/SPEAK/ordermanager/Repositories/Shop Contexts` item.
 - b. Insert new "Shop Context".
 - c. For **Name**, enter the same value you used for the **name** attribute in your site definition.
7. Once complete, you should be able to preview the site (Home item).
8. Note that workflow is enabled on much of the default Active Commerce content tree, and it is necessary to complete workflow on these items in order to publish them.