

Changes between Commerce Connect 8.1 and 8.2 Config Files

This document details the changes that have been made to the Sitecore Commerce Connect configuration files between version 8.1 and 8.2.

[Sitecore.Commerce.config](#)

The following ShopProvider and AuthenticationTokenProvider configuration has been added:

```
<shopProvider type="Sitecore.Commerce.Providers.ShopProvider, Sitecore.Commerce"
singleInstance="true"/>

<authenticationTokenProvider type="Sitecore.Commerce.Providers.AuthenticationTokenProvider,
Sitecore.Commerce" singleInstance="true"/>
```

The following commerce entity configuration has been added for the Shop and AuthenticationToken classes.

```
<commerce.Entities>
  <Shop type="Sitecore.Commerce.Entities.Shop, Sitecore.Commerce" />
  <AuthenticationToken type="Sitecore.Commerce.Entities.AuthenticationToken,
  Sitecore.Commerce" />
</commerce.Entities>
```

An analytics data class was added to record offline order synchronization:

```
<commerce.analyticsData>
  <OfflineOrdersSynchronizedAnalyticsData
  type="Sitecore.Commerce.AnalyticsData.OfflineOrdersSynchronizedAnalyticsData,
  Sitecore.Commerce"/>
</commerce.analyticsData>
```

[Sitecore.Commerce.Customers.config](#)

The following CustomerPaymentInfo configuration has been added.

```
<commerce.Entities>
  <CustomerPaymentInfo
  type="Sitecore.Commerce.Entities.Customers.CustomerPaymentInfo, Sitecore.Commerce"
  />
</commerce.Entities>
```

A spelling mistake was made for the customerSerializationProperty. The "o" in customer was missing.

```
<sitecoreCustomerRepository
type="Sitecore.Commerce.Data.Customers.SitecoreMembershipCustomerRepository,
Sitecore.Commerce">
  <param ref="entityFactory" />
  <param desc="customerSerializationProperty">__Serialization</param>
  <param desc="partySerializationProperty">__PartySerialization</param>
  <param desc="domain">CommerceCustomers</param>
```

```
<param desc="role">Customer Role</param>
</sitecoreCustomerRepository>
```

Three new pipelines have been added to support adding, removing and updating customer payment information.

```
<commerce.customers.addCustomerPaymentInfo>
  <processor
    type="Sitecore.Commerce.Pipelines.Customers.AddCustomerPaymentInfoToExternalSystem,
    Sitecore.Commerce"/>
  <processor type="Sitecore.Commerce.Pipelines.Customers.AddCustomerPaymentInfoToSitecore,
    Sitecore.Commerce">
    <param desc="customerRepository" ref="sitecoreCustomerRepository" />
  </processor>
</commerce.customers.addCustomerPaymentInfo>

<commerce.customers.removeCustomerPaymentInfo>
  <processor
    type="Sitecore.Commerce.Pipelines.Customers.RemoveCustomerPaymentInfoFromExternalSystem,
    Sitecore.Commerce"/>
  <processor
    type="Sitecore.Commerce.Pipelines.Customers.RemoveCustomerPaymentInfoFromSitecore,
    Sitecore.Commerce">
    <param desc="customerRepository" ref="sitecoreCustomerRepository" />
  </processor>
</commerce.customers.removeCustomerPaymentInfo>

<commerce.customers.updateCustomerPaymentInfo>
  <processor
    type="Sitecore.Commerce.Pipelines.Customers.UpdateCustomerPaymentInfoInExternalSystem,
    Sitecore.Commerce"/>
  <processor
    type="Sitecore.Commerce.Pipelines.Customers.UpdateCustomerPaymentInfoInSitecore,
    Sitecore.Commerce">
    <param desc="customerRepository" ref="sitecoreCustomerRepository" />
  </processor>
</commerce.customers.updateCustomerPaymentInfo>
```

[Sitecore.Commerce.ExperienceAnalytics.config](#)

This is a new configuration file to support the new Commerce Experience Analytics report feature. It defines the dimensions used during Commerce Experience Analytics reporting.

[Sitecore.Commerce.LicenseCheckPatch.config](#)

This configuration file has been removed.

[Sitecore.Commerce.Orders.config](#)

The following configuration adds a facet to a contact to allow syncing of offline orders.

```
<model>
  <elements>
    <element interface="Sitecore.Commerce.Entities.Facets.IOfflineOrders,
      Sitecore.Commerce" implementation="Sitecore.Commerce.Entities.Facets.OfflineOrders,
      Sitecore.Commerce" />
  </elements>
```

```

<entities>
  <contact>
    <facets>
      <facet name="OfflineOrders"
        contract="Sitecore.Commerce.Entities.Facets.IOfflineOrders, Sitecore.Commerce"
        help="Adds an OfflineOrders facet to a contacts information to allow for
        syncing of offline orders"/>
      </facets>
    </contact>
  </entities>
</model>

```

A new pipeline was added for processing of offline orders:

```

<!-- SYNCHRONIZE OFFLINE ORDERS
  This pipeline is used to synchronize offline orders (such as in-store orders) with
  the Sitecore xDB analytics system.
-->
<commerce.orders.synchronizeOfflineOrders>
  <!-- The CancelIfPreviewEditorOrExplorerMode processor is responsible for checking the
  current mode and setting the Result.Success flag to false if any of the following
  are true:
    Preview or Editor mode
    Explorer mode
  If any of the modes are enabled then the Success flag is set to false and a system
  message is added to indicate why
  -->
  <processor
type="Sitecore.Commerce.Pipelines.Orders.SynchronizeOfflineOrders.CancelIfPreviewEditorOrEx
plorerMode, Sitecore.Commerce">
    <param ref="entityFactory" />
  </processor>

  <!--
    The ExtractOfflineOrderDateFromXDb processor is responsible for reading the facet
    data of a contact and populating the LastOrderSynchronizedDate and
    LastOrderSynchronizedId property of the SynchronizeOfflineOrdersContext. The
    value is retrieved from the contact identified by the UserId
    property of the request. If xDB is not enabled, no action is taken.
  -->
  <processor
type="Sitecore.Commerce.Pipelines.Orders.SynchronizeOfflineOrders.ExtractOfflineOrderDateFr
omXDb, Sitecore.Commerce">
    <param ref="entityFactory" />
  </processor>

  <!--
    Integration point

    This is where you would add your custom component to retrieve the list of offline
    orders from the external commerce system.

    Get the list of offline orders from the external system and create a new Order
    entity for each one. Make sure to set the isOfflineOrder property to true for
    each Order.

    Update the list of orders in the SynchronizeOfflineOrdersContext with the newly
    retrieved orders to be processed by the following processors.

    Setting the Result.Success flag to false will cause for rest of the processors to
    do nothing.
  -->

  <!--

```

The SynchronizeOrders processor is responsible for synchronizing the offline orders found in the SynchronizeOfflineOrdersContext of the request with xDB.

This processor assumes that a previous processor has populated the list of orders. If the list is empty no sync will occur.

```
-->
<processor
  type="Sitecore.Commerce.Pipelines.Orders.SynchronizeOfflineOrders.SynchronizeOrders,
  Sitecore.Commerce">
  <param ref="entityFactory" />
</processor>

<!--
  The TriggerOfflineOrdersSynchronizedPageEvent processor is responsible for
  triggering a page event when offline orders have been synchronized
-->
<processor
type="Sitecore.Commerce.Pipelines.Orders.SynchronizeOfflineOrders.TriggerOfflineOrdersSynch
ronizedPageEvent, Sitecore.Commerce">
  <Name>Offline Orders Synchronized Details</Name>
  <Text>Offline orders have been synchronized for a user</Text>
  <param ref="entityFactory" />
</processor>

<!--
  The UpdateLastSyncInfo processor is responsible for updating the facet on the
  contact with a new date and order ID of the last offline order that was
  processed.
-->
<processor
  type="Sitecore.Commerce.Pipelines.Orders.SynchronizeOfflineOrders.UpdateLastSyncInfo,
  Sitecore.Commerce">
  <param ref="entityFactory" />
</processor>
</commerce.orders.synchronizeOfflineOrders>
```

[Sitecore.Commerce.Payments.config](#)

The following federated payment pipelines have been added:

```
<!-- GET PAYMENT SERVICE URL
  This pipeline is responsible for retrieveing the URL to a federated payment service
  provider payment accept page.
-->
<commerce.payments.getPaymentServiceUrl>
</commerce.payments.getPaymentServiceUrl>

<!-- GET PAYMENT SERVICE ACTION RESULT
  This pipeline is responsible for retrieveint the result of a federated payment
  service transaction / action.
-->
<commerce.payments.getPaymentServiceActionResult>
</commerce.payments.getPaymentServiceActionResult>
```

[Sitecore.Commerce.Products.DelayedSyncProductRepository.config.disabled](#)

This new configuration file has been introduced.