

# Free PDF Quiz 2026 Adobe AD0-E724: Valid Commerce Developer Professional Valid Exam Fee



## Adobe AD0-E724 Commerce Developer Professional

**Questions & Answers PDF  
(Demo Version – Limited Content)**

For More Information – Visit link below:

<https://p2pexam.com/>

Visit us at: <https://p2pexam.com/ad0-e724>

What's more, part of that PracticeVCE AD0-E724 dumps now are free: [https://drive.google.com/open?id=1mt1OM1iXFLf4wbu1KnNB\\_WtfVTDOGI](https://drive.google.com/open?id=1mt1OM1iXFLf4wbu1KnNB_WtfVTDOGI)

There may be customers who are concerned about the installation or use of our AD0-E724 training questions. You don't have to worry about this if you have any of this kind of trouble. In addition to high quality and high efficiency of our AD0-E724 Exam Questions, considerate service is also a big advantage of our company. We will provide 24 - hour online after-sales service to every customer to help them solve problems on our AD0-E724 learning guide.

If you want to get satisfying result in Adobe AD0-E724 practice test, our online training materials will be the best way to success, which apply to any level of candidates. We guarantee the best deal considering the quality and price of AD0-E724 Brindumps Pdf that you won't find any better available. Our learning materials also contain detailed explanations expert for correct AD0-E724 test answers.

>> AD0-E724 Valid Exam Fee <<

## AD0-E724 Prep Training - AD0-E724 Study Guide & AD0-E724 Test Pdf

The latest Commerce Developer Professional AD0-E724 exam and exam study guide is reliable, Commerce Developer Professional AD0-E724 with reasonable exam price and guaranteed questions answers. Adobe offers actual Commerce Developer Professional to sure your success in AD0-E724 Exam. Don't worry, this Commerce Developer Professional AD0-E724 test price is benefit and content is 365 days updates!

## Adobe Commerce Developer Professional Sample Questions (Q24-Q29):

### NEW QUESTION # 24

An Adobe Commerce developer is working on a custom gallery extension.

The module uses the `Magento\Catalog\Model\ImageUploader` class for image uploading. The admin controller for custom image uploads is `Vendor\CustomGallery\Controller\Adminhtml\Image\Upload`.

The images need to be stored in different `basePath` and `baseTmpPath` than the default ones.

How can the default `imageuploader` class be extended and used without affecting the other modules that are already using it?

- A.
  1. Create a Virtual Type and configure the `basePath` and `baseTmpPath`.
  2. Inject the virtual type `Vendor\CustomGallery\GalleryImageUpload` into admin controller:

```
<virtualType
  name="Vendor\CustomGallery\GalleryImageUpload"
  type="Magento\Catalog\Model\ImageUploader"
>
  <arguments>
    <argument name="baseTmpPath" xsi:type="string">customgallery/tmp/images</argument>
    <argument name="basePath" xsi:type="string">customgallery/images</argument>
  </arguments>
</virtualType>

<type name="Vendor\CustomGallery\Controller\Adminhtml\Image\Upload">
  <arguments>
    <argument name="imageUploader" xsi:type="object">
      Vendor\CustomGallery\GalleryImageUpload
    </argument>
  </arguments>
</type>
```

- B.
  1. Create a Virtual Type and configure the `basePath` and `baseTmpPath`.
  2. Add virtual type `Vendor\CustomGallery\GalleryImageUpload` as a preference for `Magento\Catalog\Model\ImageUploader`:

```
<virtualType
  name="Vendor\CustomGallery\GalleryImageUpload"
  type="Magento\Catalog\Model\ImageUploader"
  <arguments>
    <argument name="baseTmpPath" xsi:type="string">customgallery/tmp/images</argument>
    <argument name="basePath" xsi:type="string">customgallery/images</argument>
  </arguments>
</virtualType>

<preference
  for="Magento\Catalog\Model\ImageUploader"
  type="Vendor\CustomGallery\GalleryImageUpload"
/>
```

- C.
  1. Configure the `basePath` and `baseTmpPath` Of `Magento\Catalog\Model\ImageUploader`.
  2. Inject the type `Magento\Catalog\Model\ImageUploader` into admin controller:

```
<type name="Magento\Catalog\Model\ImageUploader">
  <arguments>
    <argument name="baseTmpPath" xsi:type="string">customgallery/tmp/images</argument>
    <argument name="basePath" xsi:type="string">customgallery/images</argument>
  </arguments>
</type>

<type name="Vendor\CustomGallery\Controller\Adminhtml\Image\Upload">
  <arguments>
    <argument name="imageUploader" xsi:type="object">
      Magento\Catalog\Model\ImageUploader
    </argument>
  </arguments>
</type>
```

**Answer: A**

Explanation:

To extend the `Magento\Catalog\Model\ImageUploader` class and configure it with custom `basePath` and `baseTmpPath` without affecting other modules that utilize this class, the best approach is to use a Virtual Type

. Virtual types allow you to create a customized version of a class for a specific context without altering the original class or its usage elsewhere.

Option A is correct for the following reasons:

\* Creating a Virtual Type:By defining a virtual type (Vendor\CustomGallery\Gallery\ImageUpload) in the di.xml, you create a custom version of the ImageUploader class. This virtual type can have unique configurations for basePath and baseTmpPath, which are specific to the custom gallery module.

\* Explanation: Virtual types are particularly useful in Magento when you need to use a slightly modified version of an existing class in a specific context. In this case, the virtual type allows you to define custom paths without altering the base ImageUploader class, ensuring that other modules using ImageUploader are unaffected.

### NEW QUESTION # 25

The value of a product attribute in the Adobe Commerce system needs to have a different format before it is displayed. Which attribute model class is responsible for this?

- A. Frontend
- B. Backend
- C. Source

**Answer: A**

Explanation:

The frontend attribute model class is responsible for formatting the value of a product attribute before it is displayed on the storefront. This class can implement methods such as `getValue`, `getLabel`, `getInputType`, and `getOptionText` to modify the attribute value.

The backend attribute model class is responsible for saving and loading the attribute value to and from the database. The source attribute model class is responsible for providing the list of options for an attribute.

Verified References: [Adobe Commerce Developer Guide - Attribute models] In Adobe Commerce (Magento 2), the formatting of product attribute values before they are displayed is handled by the attribute's frontend model. The frontend model, typically extending

`\Magento\Eav\Model\Entity\Attribute\Frontend\AbstractFrontend`, is responsible for preparing the attribute value for display in the frontend, including formatting dates, adding pricing structures, and more.

The backend model deals with saving and loading attribute data, while the source model provides options for select attributes.

### NEW QUESTION # 26

For security reasons, merchant requested to a developer to change default admin url to a unique url for every branch/environment of their Adobe Commerce Cloud project.

Which CLI command would the developer use update the admin url?

- A. `bin/magento adminurl:set <admin_url>`
- B. `ece-tools variable:update ADMIN_URL`
- C. `magento-cloud variable:set ADMIN_URL`

**Answer: C**

Explanation:

The CLI command that the developer would use to update the admin url is `magento-cloud variable:set ADMIN_URL`. This command sets an environment variable called `ADMIN_URL` with a custom value for the admin url on a specific environment.

Environment variables are configuration settings that affect the behavior of the Adobe Commerce Cloud application and services. By setting an environment variable for `ADMIN_URL`, the developer can change the default admin url to a unique url for every branch/environment of their Adobe Commerce Cloud project. Verified References: [Magento 2.4 DevDocs]

### NEW QUESTION # 27

Which is a correct CMS content element in Adobe Commerce?

- A. Image
- B. Sheet
- C. Widget

**Answer: C**

Explanation:

A widget is a CMS content element that can be used to display dynamic content on a page. Widgets can be used to display things like product reviews, social media feeds, or even custom content.

In Adobe Commerce, widgets are a correct CMS content element. Widgets allow merchants to add dynamic data or content blocks to CMS pages, static blocks, and various other locations throughout the store's layout without needing to directly edit the site's code. Options B (Sheet) and C (Image) are not recognized CMS content elements in the context of Adobe Commerce's terminology, making option A the correct answer.

### NEW QUESTION # 28

An Adobe Commerce developer was asked to provide additional information on a quote. When getting several quotes, the extension attributes are returned, however, when getting a single quote it fails to be returned.

What is one reason the extension attributes are missing?

- A. The developer neglected to provide a plugin `Magento\Quote\Api\CartRepositoryInterface::get`.
- B. The developer neglected to implement an observer on the `collection_load_after` event.
- C. The developer neglected to add `collection=true` to their attribute in `etc/extension_attributes.xml` file.  
attribute code="my\_attributes" type="MyVendor\MyModule\Api\Data\

BTW, DOWNLOAD part of PracticeVCE AD0-E724 dumps from Cloud Storage: [https://drive.google.com/open?id=1mt1OM1iXFLf4wbu1KnNB\\_WtfVTDOGI](https://drive.google.com/open?id=1mt1OM1iXFLf4wbu1KnNB_WtfVTDOGI)