

The Magento Certified Developer Exam (Beta)

Self-Assessment Checklist

The Magento Certified Developer (MCD) Exam is a computer-based test that has two forms: Standard and Plus. The Standard exam consists of 80 multiple-choice questions and lasts 2 hours, and the Plus exam has an additional 20 questions and lasts 2.5 hours. Both forms of the exam are based on the objectives listed below. The Plus exam additionally poses questions that test knowledge of Magento Enterprise Edition plus challenge questions about the deep structure of Magento. Questions are based on Magento Community Edition v1.5 and Magento Enterprise Edition v1.9.

You will be taking a beta version of the exam. The statistical data from the beta exam candidates will allow the test designers to set the final passing score. Therefore, you will not know whether you passed until 4 to 6 weeks after the beta. Passing the Standard form of the beta exam will earn Magento Certified Developer (MCD) certification, and passing the Plus form of the beta exam will earn MCD Plus certification.

To prepare for the exam, read through the objectives in this guide to determine which areas you need to review. You will not have access to any resources or references during the exam.

Standard Form

The Standard form of the beta exam contains questions on these ten content areas (listed with the percent of the total exam each area comprises):

- 1- Basics (5%)
- 2- Request Flow (7%)
- 3- Rendering (8%)
- 4- Working with Database in Magento (12%)
- 5- Entity-Attribute-Value (EAV) Model (10%)
- 6- Adminhtml (8%)
- 7- Catalog (10%)
- 8- Checkout (15%)
- 9- Sales and Customers (12%)
- 10- Advanced features (13%)

These are the specific objectives included in those content areas:

1- Basics

This topic comprises approximately 5% of the Standard form of the exam. Questions are drawn randomly from the following objectives:

- Fundamentals
 - Describe and apply basic principles and processes of Object Oriented Programming (OOP) and Model-View-Controller (MVC) to build Magento websites
 - Identify and describe the principles of Event-Driven Architecture (EDA)
- Magento module-based architecture
 - Describe module architecture
 - List steps to add new modules
 - Describe module limitations
- Magento directory structure
 - Describe the Magento directory structure, including naming conventions, code pools, and namespaces
 - Describe the Magento module structure
 - Describe/identify the code pools used in Magento
 - Locate templates, layout XML, skins, javascript, and temporary directories within the Magento directory structure
- Configuration XML
 - Create and register a module
 - Specify option in the config file
 - Operate website/stores/stores views
 - Add different values for different stores
- Functional and factory class groups
 - Instantiate main Magento objects

- Get different instances from different places
- Class overrides in Magento
 - Replace Model, Helper, and Block classes
 - Override a class
- Event observer
 - Register an Observer
 - Configure an Observer to work in only frontend or backend
 - Configure an Observer to work in both frontend and backend
 - Set up a cron job
 - Identify the function and proper use of automatically available events, including *_load_after, etc.
- Describe methods for resolving module conflicts.
- Internationalization
 - Describe how to plan for internationalization of a Magento site
 - Describe the use of Magento translate classes and translate files
 - Describe the advantages and disadvantages of using subdomains and subdirectories in internationalization

2- Request Flow

This topic comprises approximately 7% of the exam. Questions are drawn randomly from the following objectives:

- Application initialization
 - Describe the steps for application initialization
 - Change a website from within index.php
- Front Controller
 - Locate Front Controller class
 - List all events that Front Controller fires
 - Explain Front Controller responsibilities
- URL rewrites
 - Describe URL structure/processing in Magento
 - Describe URL rewrite process
 - Rewrite a catalog/ product/ view to a different URL
- Request routing
 - Describe request routing/flow in Magento
 - Create a module with controller
 - Override an existing controller
- Module initialization
 - Describe/identify the steps needed to create and register a new module
 - Describe/identify module dependencies
 - Describe/identify the steps needed to create a controller in a module
 - Describe/identify the steps needed to enable and disable a module
- Design and layout initialization
 - Identify the steps in the request flow in which:
 - Design data is populated
 - Layout configuration files are parsed
 - Layout is compiled
 - Output is rendered
 - Describe the module layout XML schema
 - Describe layout fallback
 - Describe admin and frontend scopes
- Structure of block templates
 - Define root template, non-output block
 - Describe how to render a root template
 - Describe how to remove a child from the template page
- Flushing data (outputs)
 - Describe how and when Magento renders content to the browser
 - Describe how and when Magento flushes output variables using Front_Controller

3- Rendering

This topic comprises approximately 8% of the exam. Questions are drawn randomly from the following objectives:

- Template structure
 - Define, and describe the use of, themes in Magento
 - Define, and describe the use of, design packages

- Define, and describe the use of, fallbacks
- Blocks
 - Describe the structure of blocks
 - Describe the relationship of templates and blocks
 - Describe the stages in the lifecycle of a block
 - Describe events fired in blocks
 - Identify different type of blocks
 - Instantiate a block
 - Disable block output
 - Describe how a typical block is rendered
- Design layout, XML schema, CMS content directives
 - Describe the elements of the Magento layout XML schema, including the major XML directives
 - Register Layout XML files
 - Create and add code to pages
 - Pass variables from layout to block
 - Add and customize Javascript

4- Working with Database in Magento

This topic comprises approximately 12% of the exam. Questions are drawn randomly from the following objectives:

- Models, resource models, and collections
 - Define/describe the basic concepts of models, resource models, and collections
 - Configure a database connection
 - Create and register new entities
 - Load and save an entity from a database
 - Filter and implement group save for a set of records
 - Describe how to use the Zend_Db classes to query the Database
 - Describe how to filter, sort and specify the selected values for collections
 - Describe how the DB abstraction layer for Magento is implemented
 - Describe how to use and resolve existing table names without hardcoding them
- Magento Object Relational Mapping
 - Describe Magento Object Relational Mapping
- Install and upgrade scripts
 - Describe the Install/Upgrade Workflow
 - Write install and upgrade scripts using set-up resources
 - Write downgrade (rollback and uninstall) scripts
 - Identify how to use the DDL class in setup scripts

5- Entity-Attribute-Value (EAV) Model

This topic comprises approximately 10% of the exam. Questions are drawn randomly from the following objectives:

- EAV model concepts
 - Define basic EAV concepts
 - Describe the EAV hierarchy structure
 - Describe how EAV data storage works in Magento
 - Describe the EAV data access process in Magento
 - Describe the database tables for EAV entities, and how to create them
 - Describe the key differences between EAV and Flat Table collections
 - Given a new entity, identify the factors that influence the choice of the resource type to use (EAV vs. flat)
- EAV entity, load and save
 - Describe EAV entity structure and its difference from the standard core resource model
 - Describe the EAV Load and Save process and its differences from the regular Load and Save process
- Attributes management
 - Describe EAV attributes, including Frontend-Source-Backend structure
 - Identify the purpose and describe how to implement the interface of attribute frontend models
 - Identify the purpose and describe how to implement the interface of attribute source models
 - Identify the purpose and describe how to implement the interface of attribute backend models
 - Create and customize attributes

6- Adminhtml

This topic comprises approximately 8% of the exam. Items are drawn randomly from the following objectives:

- Common structure/architecture
 - Describe the difference between adminhtml and frontend
 - Describe the components of Admin structure
 - Create a controller for admin router
 - Describe how to operate with cache clearing
 - Identify how to clear the cache using code (instead of the User Interface)
- Form and Grid Widgets
 - Define form structure, form templates, grids in Magento, and grid containers and elements
 - Describe grid workflow
 - Create a simple form and grid for a custom entity
 - Describe how to implement advanced Adminhtml Grids and Forms, including editable cells, mass actions, totals, reports, custom filters and renderers, multiple grids on one page, combining grids with forms, and adding custom javascript to an admin form
- System configuration XML and configuration scope
 - Define basic terms and elements of system configuration XML, including scopes, website/store/store view
 - Create an option in system configuration and output its value in a form
 - Fetch a variable from System config
- Access Control Lists (ACL) permissions
 - Define/identify basic terms and elements of ACL
 - Use ACL to:
 - Set up a menu item
 - Create appropriate permissions for users
 - Check for permissions in permissions management tree structures
- Enabling and configuring extensions
 - Define Magento extensions and describe the different types of extension available (Community, Core, Commercial)
 - Describe how extensions are installed and configured
 - Describe how to decouple extensions from Magento
 - Describe how Magento extensions connect with and inherit functionality from the rest of the system, including Payment Methods, etc.

7- Catalog

This topic comprises approximately 10% of the exam. Items are drawn randomly from the following objectives:

- Product Types
 - Identify/describe standard Product Types (simple, configurable, bundled, etc.)
 - Identify/use standard Magento tools for Product Types
 - Create custom Product Types from scratch or modify existing Product Types
 - Identify how custom Product Types interact with Indexing, SQL, underlying data structures.
- Price Generation
 - Identify basic concepts of price generation in Magento
 - Modify and adjust price generation for products, e.g. during integration of third party software
- Categories Structure
 - Dynamically alter the category tree structure
 - Choose the best-practice approach to project implementations and upgrades
 - Describe the Category Hierarchy Tree Structure implementation (the internal structure inside the database), including: the meaning of parent_id 0, the construction of paths, and the attributes required to display a new category in the store
- Catalog Rules
 - Identify how to implement catalog Price rules
 - Troubleshoot catalog Price rule implementations
- Other Skills
 - Choose optimal catalog structure (EAV vs. Flat) for given implementation
 - Troubleshoot, customize, and extend the Magento Catalog DB Structure
 - Implement, troubleshoot and modify Magento tax rules
 - Modify, extend and troubleshoot the Magento layered (aka “filter”) navigation
 - Troubleshoot and customize the Magento Indexes
 - Use custom product options to implement new features

8- Checkout

This topic comprises approximately 15% of the exam. Items are drawn randomly from the following objectives:

- Quote, Quote Item, Address and Shopping cart rules in checkout:
 - Modify these models and effectively use them in customizations
- Total models schema:
 - Add custom Total models or modify existing ones
- Shopping Cart:
 - Implement shopping cart rules
 - Troubleshoot shopping cart price rules
 - Extend shopping cart price rules
- Shipping/payment methods in Magento
 - Implement shipping methods
 - Troubleshoot shipping methods
 - Troubleshoot payment methods
 - Perform advanced modifications to the shipping rate calculation
 - Identify possibilities and limitations of the Magento multishipping implementation
 - Implement extensions of the Magento multishipping implementation
 - Payment Method Architecture: Implement new payment methods

9- Sales and Customers (12%)

This topic comprises approximately 12% of the exam. Items are drawn randomly from the following objectives:

- Sales
 - Modify Order Processing and integrate it into ERP integrations
 - Integrate Magento sales architecture with other systems (SAP, NetSuite, etc.)
- Customer:
 - Add to, delete, or customize Customer and Address attributes

10- Advanced features (13%)

This topic comprises approximately 13% of the exam. Items are drawn randomly from the following objectives:

- Widgets
 - Create frontend Widgets
- API
 - Use the Magento API to implement third party integrations
 - Extend the existing Magento API to allow for deeper integrations into third party products
 - Describe the different Web Service APIs available within the Magento Core
 - Describe the advantages and disadvantages of the available Web Service APIs in Magento
 - Identify the configuration files used for the v2 SOAP API
 - Describe the purpose of the configuration files related to the API
- Other Skills
 - Integrate Google features (Google base, checkout, adword, analytics) into Magento implementation
 - Advanced Grid/Forms: Design, use, and implement new advanced grids and forms
 - Advanced system configuration: Implement interactive and highly customized system configuration interfaces

Additional Objectives for the Plus Form

The Plus form additionally includes 20 questions drawn from the following objectives and topics:

- Customize, extend, and troubleshoot Enterprise Edition catalog target rules
- Customize, extend, and troubleshoot the Enterprise Edition reward point system
- Implement, customize, and troubleshoot Enterprise Edition website restrictions
- Identify the elements and functioning of Enterprise Edition Full Page Cache
- Challenge questions on:
 - Advanced Grids/Form
 - Advanced System Configuration
 - EAV concepts
 - Form and grid widgets
 - Catalog product types
 - Total models schema
 - Magento indexes

Sample Items:

Answer Key on the next page

1- In which of the following methods would you log event names?

- A. `Mage::addObserver()`
- B. `Varien_Event::dispatchEvent()`
- C. `Mage::logEvent()`
- D. `Mage_Core_Model_App::dispatchEvent()`

2- In which order are translations loaded?

1. Module translation files under `app/locale/[locale]/*.xml`
2. Translations stored in the DB table `core_translate`
3. Theme translations in the `translate.csv` file located in the theme `locale/[locale]/directory`

- A. 1, 2, 3
- B. 1, 3, 2
- C. 2, 1, 3
- D. 2, 3, 1

3 - When specifying a custom connection resource for a module that handles the DB reads, which of the following is a possible resource name?

- A. `<modulename_readonly>`
- B. `<modulename_read>`
- C. `<modulename_default>`
- D. `<modulename_setup>`
- E. `<modulename_connection_read>`

4 - Assuming a setup class of `Mage_Sales_Model_Mysql4_Setup`, which of the following will correctly add an attribute to the sales/order entity?

- A. `$installer->addAttribute('sales_order_entity', 'foo', array('type' => 'int'));`
- B. `$installer->addAttribute('sales/order', 'foo', array('type' => 'int'));`
- C. `$installer->addAttribute('order', 'foo', array('type' => 'int'));`
- D. `$installer->addAttribute('sales_flat_order', 'foo', array('type' => 'int'));`

5 - Which of the following operations is most impacted (in time required) by a large number of products and stores?

- A. loading a simple product page on the front
- B. saving product in the admin area
- C. importing tax rates to the database
- D. adding new attributes to be used in flat catalog

ANSWER KEY:

- 1 - D
- 2 - B
- 3 - B
- 4 - C
- 5 - D