Exam MB-400: Microsoft Power Apps + Dynamics 365 Developer – Skills Measured

This exam was updated on May 22, 2020. Following the current exam guide, we have included a version of the exam guide with Track Changes set to "On," showing the changes that were made to the exam on that date.

Audience Profile

Candidates for this exam are developers who work with Microsoft Power Apps model-driven apps in Dynamics 365 to design, develop, secure, and extend a Dynamics 365 implementation. Candidates implement components of a solution that include application enhancements, custom user experience, system integrations, data conversions, custom process automation, and custom visualizations.

Candidates must have strong applied knowledge of Power Apps model-driven apps in Dynamics 365, including in-depth understanding of customization, configuration, integration, and extensibility, as well as boundaries and constraints. Candidates should have a basic understanding of DevOps practices for Power Apps model-driven apps in Dynamics 365. Candidates must expose, store, and report on data.

Candidates should have development experience that includes JavaScript, TypeScript, C#, HTML, .NET, Microsoft Azure, Office 365, RESTful Web Services, ASP.NET, and Power BI.

Skills Measured

NOTE: The bullets that appear below each of the skills measured are intended to illustrate how we are assessing that skill. This list is not definitive or exhaustive.

NOTE: In most cases, exams do NOT cover preview features, and some features will only be added to an exam when they are GA (General Availability).

Create a Technical Design (10-15%)

Validate requirements and design technical architecture

- design and validate technical architecture
- design authentication and authorization strategy
- determine whether requirements can be met with out-of-the-box functionality
- determine when to use Logic Apps versus Power Automate flows
- determine when to use serverless computing vs. plug-ins
- determine when to build a virtual entity data source provider vs. when to use connectors

Create a data model

- design a data model

Configure Common Data Service (15-20%)

Configure security to support development

- troubleshoot operational security issues
- create or update security roles and field-level security profiles

Implement entities and fields

- configure entities
- configure fields
- configure relationships

Create and maintain solutions

- configure solutions
- import and export solutions
- manage solution dependencies

Create and Configure Power Apps (10-15%)

Create model-driven apps

- configure a model-driven app
- configure forms
- configure views
- configure visualizations

Create Canvas Apps

- configure a Canvas App
- develop complex expressions

Configure business process automation (10-15%)

Configure Power Automate

- configure a Flow
- configure actions to use Common Data Service connectors
- develop complex expressions

Implement processes

- create and configure business process flows
- create and configure business rules

Extend the user experience (15-20%)

Apply business logic using client scripting

- configure supporting components
- create JavaScript or Typescript code
- register an event handler
- use the Web API from client scripting

Create a Power Apps Component Framework (PCF) component

- initialize a new PCF component
- configure a PCF component manifest
- implement the component interfaces
- package, deploy, and consume the component
- use Web API device capabilities and other component framework services

Create a command button function

- create the command function
- design command button triggers, rules, and actions
- edit the command bar using the Ribbon Workbench
- modify the form JavaScript library dependencies

Extend the platform (15-20%)

Create a plug-in

- debug and troubleshoot a plug-in
- develop a plug-in
- use the global Discovery Service endpoint
- optimize plug-ins for performance
- register custom assemblies by using the Plug-in Registration Tool
- create custom actions

Configure custom connectors for Power Apps and Flow

- create a definition for the API
- configure API security

- use policy templates

Use platform APIs

- interact with data and processes using the Web API
- optimize for performance, concurrency, transactions, and batching
- perform discovery using the Web API
- perform entity metadata operations with the Web API
- use OAuth with the platform APIs

Develop Integrations (10-15%)

Publish and consume events

- publish an event by using the API
- publish an event by using the Plug-in Registration Tool
- register a webhook
- create an Azure event listener application

Implement data synchronization

- configure and use entity change tracking
- configure the data export service to integrate with Azure SQL Database
- create and use alternate keys

The exam guide below shows the changes that were implemented on May 22, 2020.

Audience Profile

Candidates for this exam are developers who work with Microsoft Power Apps model-driven apps in Dynamics 365 to design, develop, secure, and extend a Dynamics 365 implementation. Candidates implement components of a solution that include application enhancements, custom user experience, system integrations, data conversions, custom process automation, and custom visualizations.

Candidates must have strong applied knowledge of Power Apps model-driven apps in Dynamics 365, including in-depth understanding of customization, configuration, integration, and extensibility, as well as boundaries and constraints. Candidates should have a basic understanding of DevOps practices for Power Apps model-driven apps in Dynamics 365. Candidates must expose, store, and report on data.

Candidates should have development experience that includes JavaScript, TypeScript, C#, HTML, .NET, Microsoft Azure, Office 365, RESTful Web Services, ASP.NET, and Power BI.

Skills Measured

NOTE: The bullets that appear below each of the skills measured are intended to illustrate how we are assessing that skill. This list is not definitive or exhaustive.

NOTE: In most cases, exams do NOT cover preview features, and some features will only be added to an exam when they are GA (General Availability).

Create a Technical Design (10-15%)

Validate requirements and design technical architecture

- design and validate technical architecture
- design authentication and authorization strategy
- determine whether requirements can be met with out-of-the-box functionality
- determine when to use Logic Apps versus ~~Microsoft Flow~~Power Automate flows
- determine when to use serverless computing vs. plug-ins
- determine when to build a virtual entity data source provider vs. when to use connectors

Create a data model

- design a data model

Configure Common Data Service ~~(CDS)~~ (15-20%)

Configure security to support development

- troubleshoot operational security issues
- create or update security roles and field-level security profiles

Implement entities and fields

- configure entities
- configure fields
- configure relationships

Create and maintain solutions

- configure solutions
- import and export solutions
- manage solution dependencies

Create and Configure Power Apps (10-15%)

Create model-driven apps

- configure a model-driven app
- configure forms
- configure views
- configure visualizations

Create Canvas Apps

- configure a Canvas App
- develop complex expressions

Configure business process automation (10-15%)

Configure ~~Microsoft Flow~~Power Automate

- configure a Flow
- configure actions to use ~~CDS~~ Common Data Service connectors
- develop complex expressions

Implement processes

- create and configure business process flows
- create and configure business rules

Extend the user experience (15-20%)

Apply business logic using client scripting

- configure supporting components
- create JavaScript or Typescript code
- register an event handler
- use the Web API from client scripting

Create a Power Apps Component Framework (PCF) component

- initialize a new PCF component
- configure a PCF component manifest
- implement the component interfaces
- package, deploy, and consume the component
- use Web API device capabilities and other component framework services

Create a command button function

- create the command function

- design command button triggers, rules, and actions
- edit the command bar using the Ribbon Workbench
- modify the form JavaScript library dependencies

Extend the platform (15-20%)

Create a plug-in

- debug and troubleshoot a plug-in
- develop a plug-in
- use the ~~Organization Service~~ global Discovery Service endpoint
- optimize plug-ins for performance
- register custom assemblies by using the Plug-in Registration Tool
- create custom actions

Configure custom connectors for Power Apps and Flow

- create a definition for the API
- configure API security
- use policy templates

Use platform APIs

- interact with data and processes using the Web API
- optimize for performance, concurrency, transactions, and batching
- perform discovery using the Web API
- perform entity metadata operations with the Web API
- use OAuth with the platform APIs

Develop Integrations (10-15%)

Publish and consume events

- publish an event by using the API
- publish an event by using the Plug-in Registration Tool
- register a webhook
- create an Azure event listener application

Implement data synchronization

- configure and use entity change tracking
- configure the data export service to integrate with Azure SQL Database

- create and use alternate keys