



# MB-500<sup>Q&As</sup>

Microsoft Dynamics 365 Finance and Operations Apps Developer

## Pass Microsoft MB-500 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.passapply.com/mb-500.html>

100% Passing Guarantee  
100% Money Back Assurance

Following Questions and Answers are all new published by Microsoft  
Official Exam Center

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers





## QUESTION 1

### DRAG DROP

You are a Dynamics 365 Finance developer for a company that sells pre-owned vehicles. When the company acquires a new vehicle, vehicle identification (VIN) numbers are added to the VIN field in the CustTable.

You need to create an event handler to validate that new VIN values are at least 10 characters long. Validation must occur when records are saved.

Which three actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.

Select and Place:

### Answer Area

#### Actions

Copy the validatingWrite event handler on the CustTable table object.

Copy the validateWrite event handler on the CustTable object.

Add code to perform the validation.

Copy the onModified event handler on the CustTable object.

Add the event handler method to a class.

Correct Answer:



### Answer Area

#### Actions

Copy the validatingWrite event handler on the CustTable table object.

Copy the onModified event handler on the CustTable object.

Copy the validateWrite event handler on the CustTable object.

Add the event handler method to a class.

Add code to perform the validation.

Reference: <https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/extensibility/customization-overlayering-extensions>

### QUESTION 2

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are adding a new field to the SalesTable form.

You must use an extension to add a status field onto the form.

You need to create the extension in the Application Object Tree (AOT) and add the extension to the demoExtensions model.

Solution: Add a new form object to the project and name the form SalesTable.Extension.

Does the solution meet the goal?

A. Yes

B. No

Correct Answer: B

Instead navigate to the user interface forms section for the SalesTable form and create an extension.

Note: In Dynamics 365 Finance and Operations, the new fields will need to be added via a table extension. Create the extensions on the SalesTable.

Reference:



<https://stoneridgesoftware.com/how-to-extend-sales-order-update-functionality-to-custom-fields-in-d365-finance-and-operations/>

---

### QUESTION 3

A company uses Dynamics 365 Supply Chain Management.

You need to list all classes in the Application Suite model that contain a cross-company query.

Which query string should you use?

- A. type:class, method name=run code:"crosscompany" model:"Application Suite"
- B. type:class, table code:"crosscompany" model:"Application Suite"
- C. type:class code:"crosscompany" model:"Application Suite"
- D. type:class, method name=insert code:"crosscompany" model:"Application Suite"

Correct Answer: A

The Query class embodies the structure of a query.

Note 1: Query class Objects of this kind are not used to fetch records from the database. Instead, use a QueryRun object that can be assigned a query object. The dynamic behavior of a query is defined by the QueryRun class. The static behavior is defined by the Query class. Queries contain one or more data sources that correspond to tables in the database. The data sources are specified by using QueryBuildDataSource objects. This class lets you create, read, update, and delete X++ code and metadata.

The following example creates a query object that is used to create a QueryRun object.

X++

```
{ Query q = new Query (QueryStr(Cust)); // Use the query to build a queryRun object. QueryRun qr = new QueryRun  
(q); // Traverse some records. while (qr.next()) {  
  
// ... } }
```

Note 2:

You can create a cross-company query by using X++ code. There are two ways to do this:

crossCompany keyword on the select statement

Query class methods

Reference:

<https://learn.microsoft.com/en-us/dynamicsax-2012/developer/cross-company-x-code-basics>

<https://learn.microsoft.com/en-us/dotnet/api/dynamics.ax.application.query>

---

### QUESTION 4



You create a new class.

Class compilation must fail if a specific method in the source code is called.

You need to mark the class with the appropriate attribute.

Which attribute should you use?

- A. SysAppSecurityAttribute
- B. SysTestCategory
- C. SysObsoleteAttribute
- D. SysAttribute

Correct Answer: C

Reference: <https://docs.microsoft.com/en-us/dynamicsax-2012/developer/overview-of-attribute-classes>

#### QUESTION 5

A company uses Dynamics 365 Finance.

You must create a process that updates the following:

1.  
A single record for customer number A0001 in the customer table.
2.  
The value of its customer group to 10. You need to implement the process. Which code segment should you use?

Name	Comments
AccountBase	Root EDT
AccountId	Derives from AccountBase

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Correct Answer: D

Update method.



The update method updates the current record with the contents of the buffer. It also updates the appropriate system fields. The optional where clause specifies a condition that the update method tests as it processes each row of the table.

Only those rows that test true against the condition are updated with the new values.

The following example selects the CustTable table for update. Only records where the value of the AccountNum field equals 4000 are updated. Because there is no call to next, and this example doesn't use a select while statement, only one

record is updated. The value of the CreditMax field is changed to 5000.

```
CustTable custTable; ttsBegin; select forUpdate custTable
```

```
where custTable.AccountNum == \"4000\";
```

```
custTable.CreditMax = 5000;
```

```
custTable.update();
```

```
ttsCommit;
```

Note: forUpdate Checking

This check ensures that no record can be updated or deleted if the record has not first been selected for update. A record can be selected for update, either by using the forUpdate keyword in the select statement, or by using the

selectForUpdate method on tables.

ttsLevel Checking

This check ensures that no record can be updated or deleted except from within the same transaction scope as it was selected for update. Integrity is ensured by using the following statements:

ttsBegin: marks the beginning of a transaction. This ensures data integrity, and guarantees that all updates performed until the transaction ends (by ttsCommit or ttsAbort) are consistent (all or none).

ttsCommit: marks the successful end of a transaction. This ends and commits a transaction. MorphX guarantees that a committed transaction will be performed according to intentions.

Incorrect:

Not A: Use ttsBegin and ttsCommit.

There is also a typo in the last line:

```
ustTable.Update(); instead of custTable.Update();
```

Not B: need to use

```
where custTable.AccountNum == \"A0001\"
```

to select the proper record.

Not C: Use select forupdate

Reference:



VCE & PDF

PassApply.com

<https://www.passapply.com/mb-500.html>

2024 Latest passapply MB-500 PDF and VCE dumps Download

---

<https://learn.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/dev-ref/xpp-data/xpp-update>

<https://learn.microsoft.com/en-us/dynamicsax-2012/developer/transaction-integrity>

[MB-500 PDF Dumps](#)

[MB-500 VCE Dumps](#)

[MB-500 Practice Test](#)