MB-820 Q&As

Microsoft Dynamics 365 Business Central Developer

Pass Microsoft MB-820 Exam with 100% Guarantee

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

https://www.leads4pass.com/mb-820.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 developing an XMLport to export data from the parent Item table and a related child "Item Unit of Measure\\' table. The XMLport configuration must provide the following:

1.

Link the child table to its parent.

2.

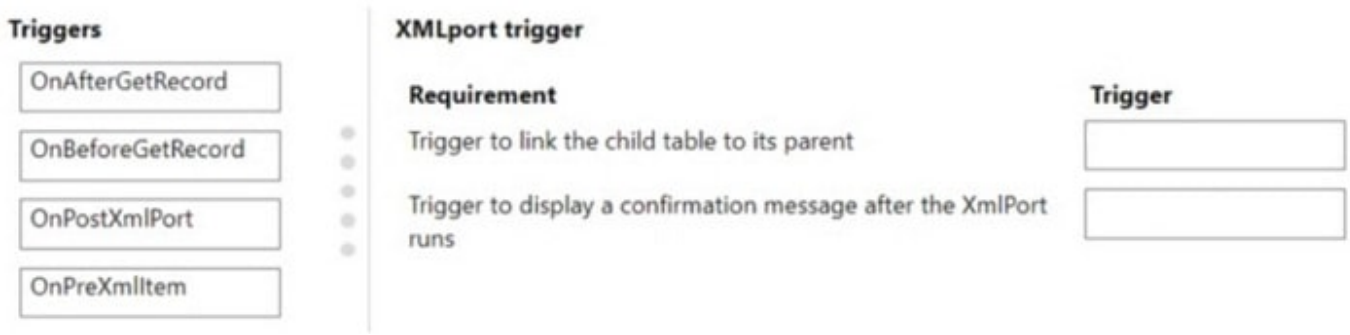
Display a confirmation message after the XMLport runs.

You need to generate the XMLport.

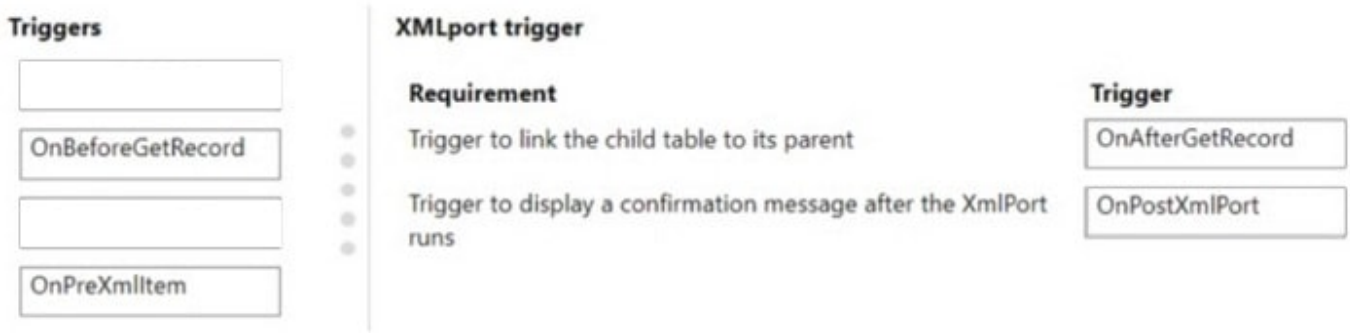
What should you do? To answer, move the appropriate triggers to the correct requirements. You may use each trigger once, more than once, or not at all. You may need to move the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Select and Place:



Correct Answer:



To meet the XMLport configuration requirements:

Link the child table to its parent: Use the OnAfterGetRecord trigger. Display a confirmation message after the XMLport runs: Use the OnPostXMLPort trigger.

In Business Central, when you are developing an XMLport for data export, triggers are used to perform actions at

different stages of the XMLport\\'s operation:

OnAfterGetRecord Trigger:This trigger fires after a record is retrieved from the database but before it is processed for output in the XMLport. It is the ideal place to link child table records to their parent because you have access to the current

record that can be used to set filters or modify data in the child table before it is written to the XML file.

OnPostXMLPort Trigger:This trigger fires after the XMLport has finished processing all records. It is the correct place to display a confirmation message because it ensures that the message will appear after the entire XMLport operation is

complete. Here, you can use application-specific functions to show the message, such as MESSAGE function in AL code.

By placing the appropriate triggers in these positions, you can ensure that the XMLport will link the child records to their parent records during the data export process and will notify the user with a confirmation message once the operation is

successfully completed.

QUESTION 2

You create a page with the PageType property set to RoleCenter.

You navigate through the different sections of the page.

You need to add functionalities to the page.

What should you do?

A. Define actions in the area (reporting) before actions in the area (creation).

B. Define the navigation menu in the area(processing).

C. Define the navigation bar in the area (embedding).

D. Add a source table on the Role Center page.

Correct Answer: A

When creating a page with the PageType property set to RoleCenter in Microsoft Dynamics 365 Business Central, it\\'s essential to organize the functionalities and actions in a manner that enhances user experience and efficiency. The best practice is to define actions in the area (reporting) before actions in the area (creation) (A). This organization allows users to access reporting and analytical features quickly, which are commonly used in Role Centers for overview and insight purposes, before moving on to creation or transactional tasks. This logical flow aligns with typical user workflows, where analysis and review precede the creation of new records or transactions. The other options, such as defining the navigation menu in the area(processing) (B), defining the navigation bar in the area (embedding) (C), or adding a source table on the Role Center page (D), do not directly address the need to add functionalities to the Role Center page in a user- friendly manner.

QUESTION 3

You plan to write unit test functions to test newly developed functionality in an app.

You must create a test codeunit to write the functions.

You need to select the property to use for the test codeunit.

Which property should you use to ensure that the requirements are fulfilled?

A. SubType

B. Access

C. Description

Correct Answer: A

When creating a test codeunit in Microsoft Dynamics 365 Business Central to write unit test functions, the SubType property (A) of the codeunit should be set to Test. This property is crucial for defining the codeunit\\'s purpose and behavior within the application. By setting the SubType property to Test, you are indicating that the codeunit contains test functions intended to validate the functionality of other parts of the application, such as customizations or new developments. This distinction ensures that the testing framework within Business Central recognizes the codeunit as a container for test functions, allowing it to execute these functions in a testing context, which can include setting up test data, running the tests, and cleaning up after the tests have completed.

QUESTION 4

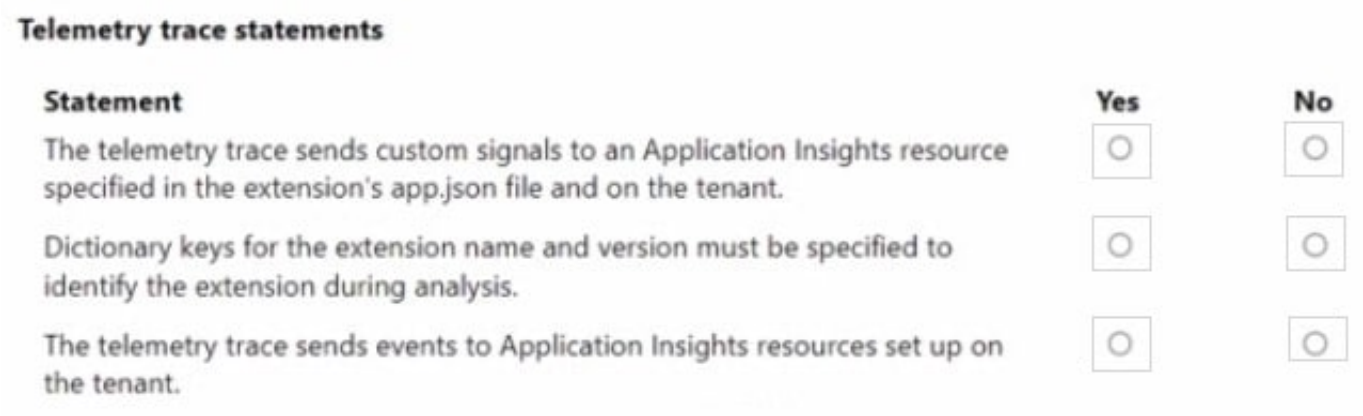
HOTSPOT

A company is setting up a custom telemetry trace signal to send traces on failed customer statement emails.

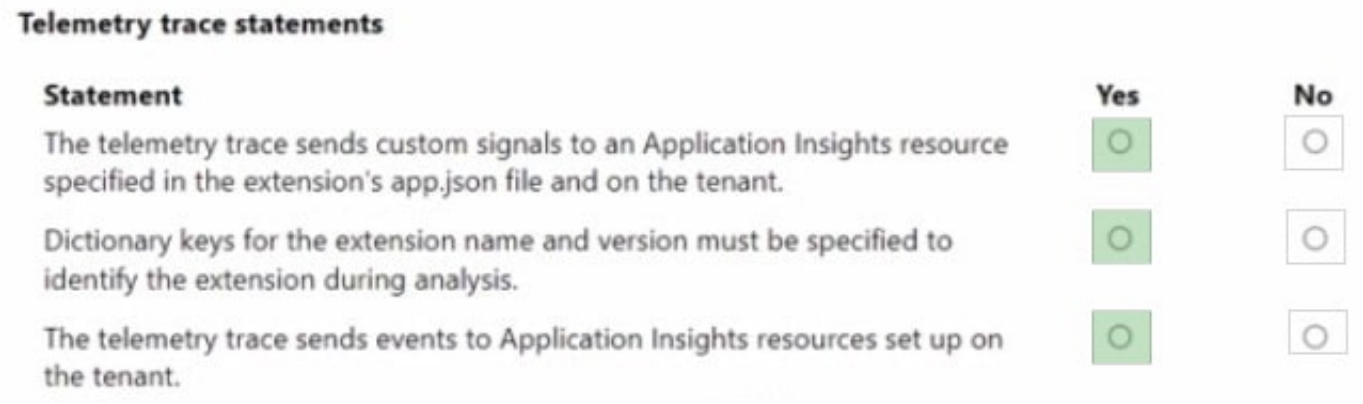
```
05  local procedure SendTraceOnFailedToEmailCustomerStatement(Customer: Record Customer)
06  var
07      Dimensions: Dictionary of [Text, Text];
08      FailedEmailLbl: Label 'Failed to email customer statement';
09  begin
10       Dimensions.Add('systemId', Customer.SystemId);
11      Session.LogMessage('FCUSTSTMT', FailedEmailLbl, Verbosity::Error,
12      DataClassification::SystemMetadata, TelemetryScope::ExtensionPublisher, Dimensions);
13  end;
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No.

Hot Area:



Correct Answer:



The telemetry trace sends custom signals to an Application Insights resource specified in the extension\\'s app.json file and on the tenant. = YES Dictionary keys for the extension name and version must be specified to identify the extension

during analysis. = YES

The telemetry trace sends events to Application Insights resources set up on the tenant. = YES

Telemetry in Business Central allows developers to collect custom telemetry for extensions using Application Insights. The telemetry trace is used to send custom signals to an Application Insights resource. This resource is typically specified

in the app.json file of the extension and must be configured on the tenant where the extension is installed. The use of dictionary keys for the extension name and version is a best practice to identify the extension during analysis in Application

Insights. These keys can be added to the telemetry trace to ensure that when the data is collected, it\\'s clear which extension the data is associated with.

Finally, it is correct that the telemetry trace sends events to Application Insights resources that are set up on the tenant, enabling the collection and analysis of telemetry at the tenant level.

QUESTION 5

HOTSPOT

You develop a test application.

You must meet the following requirements:

1.

Roll back changes to a test method after run time.

2.

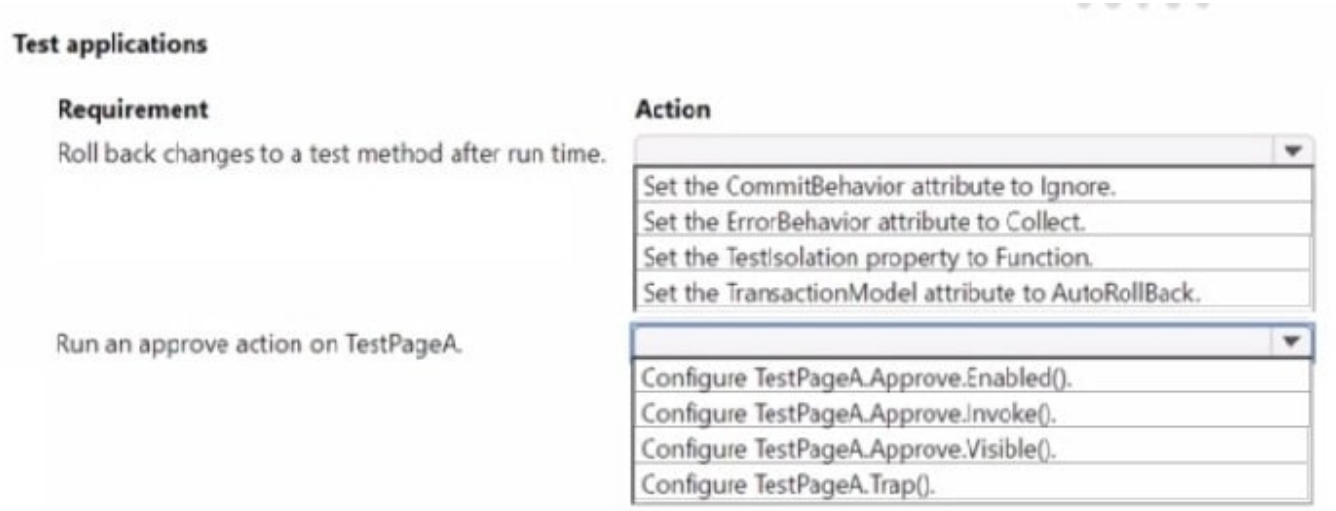
Run an approve action on a test page named TestPageA.

You need to implement the given requirements on the test codeunit

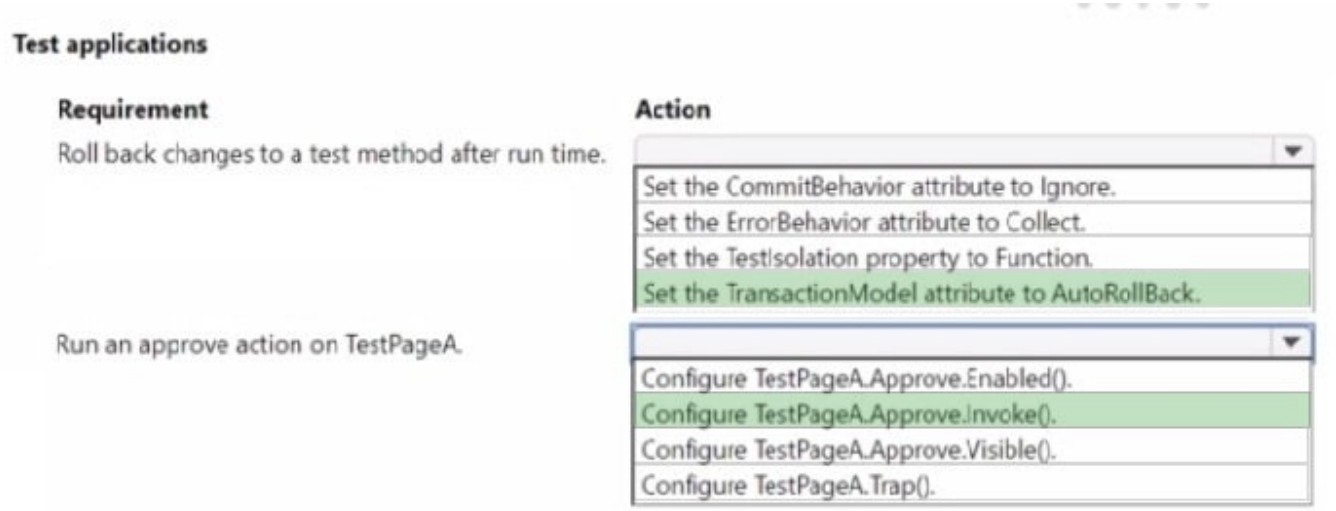
Which actions should you perform? To answer, select the appropriate options in the answer area

NOTE: Each correct selection is worth one point.

Hot Area:



Correct Answer:



To roll back changes to a test method after run time, you should:

Set the TransactionModel attribute to AutoRollback. To run an approve action on a test page named TestPageA, you should:

Configure TestPageA.Approve.Invoke().

In Business Central\\'s testing framework, the TransactionModel attribute can be set to AutoRollback. This ensures that any changes made during the test are rolled back after the test is complete, leaving the database in its original state. For

running an action on a test page, you would use the \\'Invoke\\' method on the action you wish to perform. In this case, to run an approve action on TestPageA, you would use TestPageA.Approve.Invoke() within your test codeunit. This simulates

the user action of approving something on the page.

These actions ensure that the testing environment is properly set up to test specific functionalities without persisting test data and to invoke actions as part of the test scenarios.