



Free Questions for MB-820

Shared by Dawson on 24-05-2024

For More Free Questions and Preparation Resources

[Check the Links on Last Page](#)



Question 1

Question Type: MultipleChoice

Case Study: Mix Questions

Mix Questions

MB-820 Mix Questions IN THIS CASE STUDY

A company uses Business Central.

The company plans to use a translation file in an extension. The extension has a caption that should not be translated.

You need to prevent the caption from being translated.

What should you do?

Options:

- A- Use the CaptionML property and copy the same caption for each language used.
- B- Set the GenerateLockedTranslations feature in the appjson file.
- C- Add the Locked = true parameter to the Caption.
- D- Delete the Caption property.
- E- Copy the same caption for each language in the translation file.

Answer:

C

Explanation:

To prevent a caption from being translated in an extension for Microsoft Dynamics 365 Business Central, you should add the Locked = true parameter to the Caption (C). This parameter explicitly marks the caption as locked for translation, ensuring that it remains unchanged across different language versions of the extension. This approach is useful for specific terms, brand names, or other elements within the application that should remain consistent regardless of the user's language settings. Unlike the other options, which involve manual manipulation of the translation file or properties, setting Locked = true directly in the AL code provides a clear, maintainable, and error-proof method to exclude specific captions from the translation process.

Question 2

Question Type: DragDrop

Case Study: Mix Questions

Mix Questions

MB-820 Mix Questions IN THIS CASE STUDY

A company is implementing Business Central.

In the per-tenant extension, TableA Header and TableA Line are document tables, and TableB Header and TableB Line are document history tables.

The company requires that the resulting dataset of query objects contain the following records:

- * All records from TableA Header even if no matching record value exists in the linked TableA Line
- * Records from TableB Header where a match is found in the linked TableB Line field

You need to configure the linked data item to generate the required dataset.

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

The screenshot shows the configuration interface for a linked data item. On the left, under 'SqlJoinTypes', there are four buttons: CrossJoin, InnerJoin, LeftOuterJoin, and RightOuterJoin. In the center, under 'Link data items', there are four empty slots, each with a small circle to its left. On the right, under 'Dataset requirement', there are two text boxes: 'Include all records from TableA Header.' and 'Include only matched records from TableB Header.' To the right of these requirements is a 'SqlJoinType' column with two empty input boxes.

Answer:

See the Answer in the Premium Version!

Question 3

Question Type: MultipleChoice

Case Study: Mix Questions

Mix Questions

MB-820 Mix Questions IN THIS CASE STUDY

You need to determine If you have unwanted incoming web service calls in your tenant during the last seven days.

Which two KQL queries should you use? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A.

```
traces | where timestamp > ago(7d) | where customDimensions has 'RT0008' | where customDimensions.category !in ('ODataV4', 'ODataV3', 'Api')
```

B.

```
traces | where timestamp > ago(7d) | where customDimensions has 'RT0008' | where customDimensions.category == 'SOAP'
```

C.

```
traces | where timestamp > ago(7d) | where customDimensions == 'RT0008' | where customDimensions.category == 'SOAP'
```

D.

```
traces | where timestamp > ago(7d) | where customDimensions has 'RT0008' | where customDimensions.category != 'ODataV4'
```

E.

```
traces | where timestamp > ago(7d) | where customDimensions has 'RT0008' | where customDimensions.category !in ('ODataV4', 'Api')
```

Options:

- A- Option A
- B- Option B
- C- Option C
- D- Option D
- E- Option E

Answer:

A, C

Explanation:

The task is to identify unwanted incoming web service calls during the last seven days. To do this, we need to look at KQL (Kusto Query Language) queries that would filter out web service calls based on the timestamp (to ensure the calls are within the last seven days) and by certain

characteristics that would indicate they are unwanted, such as the wrong type of protocol (SOAP in this case, as Contoso Ltd. plans to dismiss using it).

Looking at the options:

Option A: This query selects all traces where the timestamp is within the last 7 days and where the custom dimension has a value of 'RT0008', and where the category is either 'ODataV4', 'ODataV3', or 'Api'. This query would show all API calls except SOAP, so it does not directly answer the question about unwanted calls.

Option B: This query filters for traces with a timestamp within the last 7 days, where 'RT0008' is present, and specifically looks for the category 'SOAP'. This query is correct because it directly targets SOAP calls, which are the unwanted calls according to Contoso Ltd.'s plans.

Option C: Similar to option B, this query filters for traces within the last 7 days and looks for 'RT0008' but uses the equality operator for the category 'SOAP'. This would also correctly return the unwanted SOAP calls.

Option D: This query also filters for traces within the last 7 days, but it excludes the 'ODataV4' category, which doesn't necessarily target the unwanted SOAP calls.

Option E: This query selects traces where the timestamp is within the last 7 days and the custom dimension has 'RT0008'. However, it filters out categories 'ODataV4' and 'Api', which does not directly help in identifying the unwanted SOAP calls.

Therefore, the queries that should be used to determine if there are unwanted incoming web service calls (SOAP calls) in the tenant during the last seven days are Options B and C. These queries are specific to identifying SOAP protocol usage, which is what Contoso Ltd. considers unwanted.

Question 4

Question Type: Hotspot

Case Study: Mix Questions

Mix Questions

MB-820 Mix Questions IN THIS CASE STUDY

A company plans to customize its per tenant extension reports. The company has the following requirements for the customization:

- * Child data items must not be displayed on the request page for some master detail reports.
- * Selecting key filter fields takes users too much time. The customization must decrease the amount of time to select the fields.

You need to optimize the report request page.

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

NOTE: Each correct selection is worth one point.

Report request page

Observation

Child data items of some master detail reports must not be displayed on the request page.

Decrease the amount of time to select filter fields.

Action

Set the PrintOnlyIfDetail property to true.
Set the UseRequestPage property to true.
Set the DataItemTableView sorting property.
Set the DataItemLinkReference property to the parent data item.

Set the SaveValues Property to true.
Specify the request page options.
Specify the RequestFilterFields property.
Specify the RequestFilterHeading property.

Answer:

See the Answer in the Premium Version!

Question 5

Question Type: MultipleChoice

Case Study: Mix Questions

Mix Questions

MB-820 Mix Questions IN THIS CASE STUDY

You need to allow debugging in an extension to view the source code. In which file should you specify the value of the allowDebugging property?

Options:

- A- settings.json
- B- rad.json
- C- app.json
- D- launchjson

Answer:

C

Explanation:

To enable debugging in an extension and allow the source code to be viewed, the allowDebugging property should be specified in the app.json file (C). The app.json file serves as the manifest for an AL project in Microsoft Dynamics 365 Business Central, defining the project's properties, dependencies, and features. By setting the allowDebugging property to true in this file, developers enable the debugging of the extension's source code, facilitating troubleshooting and development. This is essential for analyzing the behavior of the extension and identifying issues during the development process.

Question 6

Question Type: Hotspot

Case Study: Mix Questions

Mix Questions

MB-820 Mix Questions IN THIS CASE STUDY

You need to create the codeunit to read the POS terminal APIs.

How should you complete the code segment? To answer, select the appropriate options in the answer area.

NOTE; Each correct selection is worth one point.

Create and access codeunits

```
codeunit 52102 "POS API Management"
{
    Access = Internal
    Access = Public
    Permissions = TableData "POS Information" = rdx
    Permissions = TableData "POS Information" = RMDX

    trigger OnRun()
    begin
        readAPI();
    end;

    procedure readAPI()
    procedure readAPI(PosNo: Integer)
    var procedure readAPI()

    begin
        // your code here
    end;
}
```

Answer:

See the Answer in the Premium Version!

Question 7

Question Type: Hotspot

Case Study: Mix Questions

Mix Questions

MB-820 Mix Questions IN THIS CASE STUDY

You develop a test application.

You must meet the following requirements:

- * Roll back changes to a test method after run time.
- * 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.

Test applications

Requirement

Roll back changes to a test method after run time.

Run an approve action on TestPageA.

Action

Set the CommitBehavior attribute to Ignore.
Set the ErrorBehavior attribute to Collect.
Set the TestIsolation property to Function.
Set the TransactionModel attribute to AutoRollBack.

Configure TestPageA.Approve.Enabled().
Configure TestPageA.Approve.Invoke().
Configure TestPageA.Approve.Visible().
Configure TestPageA.Trap().

Answer:

See the Answer in the Premium Version!

Question 8

Question Type: OrderList

Case Study: Mix Questions

Mix Questions

MB-820 Mix Questions IN THIS CASE STUDY

A company has the following custom permission set:

```

permissionset 50000 "Sales Person Permission Set"
{
    Assignable = false;
    Caption = 'Sales Person Permission Set';

    Permissions =
        tabledata Customer = RIMD,
        tabledata "Payment Terms" = RMD,
        tabledata Currency = RM,
        tabledata "Sales Header" = RIM,
        tabledata "Sales Line" = RIMD;
}

```

You need to make the permission set visible on the Permission Sets page.

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.

NOTE: More than one order of answer choices is correct. You will receive credit for any of the correct orders you select.

Actions

-
-
-
-
-
-
-
-

Process for making permission sets visible



Answer:

Change the Assignable property value to true. Add the ObsoleteState = No property. Publish the app with the permission set to an environment.

Question 9

Question Type: Hotspot

Case Study: Mix Questions

Mix Questions

MB-820 Mix Questions IN THIS CASE STUDY

You need to define the properties of the comments field of the Non-conformity page.

How should you complete the code segment? To answer, select the appropriate options in the answer area.

NOTE; Each correct selection is worth one point.

ExtendedDataType property

```

group(commentsGroup)
{
    field("comments"; NonConformityComments)
    {
        ApplicationArea = All;
        MultiLine = True;
        MultiLine = False;
        NotBlank = True;
        NotBlank = False;
        ExtendedDataType = {
            LongContent
            None
            RichContent
            TextRichContent
        };
        DataType
        ExtendDataType
        ExtendedDatatype
        RichDataType
    }
}
-----
var
    NonConformityComments: Text;
  
```

Answer:

See the Answer in the Premium Version!

Question 10

Question Type: Hotspot

Case Study: Mix Questions

Mix Questions

MB-820 Mix Questions IN THIS CASE STUDY

You create a table with fields.

You observe errors in the code

You need to resolve the errors.

```

01 field(1; "Job No."; Code[20])
02 {
03     Caption = 'Job No.';
04 }
05 field(2; Description; text[150])
06 {
07     Caption = 'Description';
08 }
09 field(3; "Sales Amount"; Decimal)
10 {
11     AutoFormatType = 1;
12     CalcFormula = sum("Job Task"."Recognized Sales Amount" where("Job No." = field("No.")));
13     Caption = 'Calc. Recog. Sales Amount';
14     Editable = false;
15 }
16 field(5; "Over Budget"; Boolean)
17 {
18     Caption = 'Over Budget';
19 }
20 field(6; "Project Manager"; Code[50])
21 {
22     Caption = 'Project Manager';
23     TableRelation = "User Setup";
24 }

```

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

NOTE: Each correct selection is worth one point.

Error resolution

Statement	Yes	No
In line 12, declare "Job Task" as a variable.	<input type="radio"/>	<input type="radio"/>
Add the property FieldClass = FlowField; for field 3.	<input type="radio"/>	<input type="radio"/>
Add the property FieldClass = FlowFilter; for field 3.	<input type="radio"/>	<input type="radio"/>
In line 23, assign the "User Setup" table to a field.	<input type="radio"/>	<input type="radio"/>

Answer:

See the Answer in the Premium Version!

To Get Premium Files for MB-820 Visit

<https://www.p2pexams.com/products/mb-820>

For More Free Questions Visit

<https://www.p2pexams.com/microsoft/pdf/mb-820>

20%
DISCOUNT

P2P
exams