

# Microsoft MB-820

**Microsoft Dynamics 365 Business Central Developer**

**For More Information – Visit link below:**

**<https://www.examsempire.com/>**

**Product Version**

- 1. Up to Date products, reliable and verified.**
- 2. Questions and Answers in PDF Format.**



**<https://examsempire.com/>**

**Visit us at: <https://www.examsempire.com/mb-820>**

# Latest Version: 7.0

## Question: 1

### HOTSPOT

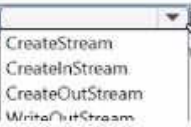
You need to download a stored picture from the Room Incident page.

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.

F

InStream and OutStream



```
local procedure DownloadIncidentPicture(Incident : Record Incident)
var
    TempBlob : Codeunit "Temp Blob";
    IncidentOutStream : OutStream;
    IncidentInStream : InStream;
    ImageFilter, FileName : Text;
begin
    TempBlob. (IncidentOutStream);
    Incident.Image.ExportStream(IncidentOutStream);
    TempBlob. (IncidentInStream);
    ImageFilter := 'Image Files (*.bmp;*.jpg;*.gif)|*.bmp;*.jpg;*.gif';
    FileName := 'Customer Picture';

    if not DownloadFromStream(, 'Download Incident Picture', '', ImageFilter, FileName) then
        exit;
```

**Answer:**

## InStream and OutStream

```

local procedure DownloadIncidentPicture(Incident : Record Incident)
var
    TempBlob : Codeunit "Temp Blob";
    IncidentOutStream : OutStream;
    IncidentInStream : InStream;
    ImageFilter, FileName : Text;
begin
    TempBlob. (IncidentOutStream);
    Incident.Image.ExportStream(IncidentOutStream);
    TempBlob. (IncidentInStream);

    ImageFilter := 'Image Files (*.bmp;*.jpg;*.gif)|*.bmp;*.jpg;*.gif';
    FileName := 'Customer Picture';

    if not DownloadFromStream(, 'Download Incident Picture', '', ImageFilter, FileName) then
        exit;
    end;

```

Explanation:

```

var TempBlob: Codeunit "Temp Blob";
IncidentOutStream: OutStream;
IncidentInStream: InStream;
ImageFilter, FileName: Text;
begin
    // Initialize the TempBlob and streams
    TempBlob.CreateOutStream(IncidentOutStream);
    Rec.Image.ExportStream(IncidentOutStream); // 'Rec' refers to the current Room Incident record
    TempBlob.CreateInStream(IncidentInStream);
    // Set the filters and filename for the image
    ImageFilter := 'Image Files (*.bmp;*.jpg;*.jpeg;*.gif)|*.bmp;*.jpg;*.jpeg;*.gif';
    FileName := 'Customer Picture';
    // Prompt the user to download the image
    if not DownloadFromStream(IncidentInStream, '', 'Download Incident Picture', '', ImageFilter,
        FileName) then
        Error('Unable to download the image.');
```

## Question: 2

HOTSPOT

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:**

## 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;  
}
```

Explanation:

codeunit 52102 "POS API Management"

```
{  
Access = Public;  
Permissions = TableData "POS Information" = rwdx;  
trigger OnRun()  
begin  
readAPI();  
end;  
procedure readAPI()  
begin  
// Your code here to read from the POS API  
end;  
}
```

## Create and access codeunits

```
codeunit 52102 "POS API Management"
{
    // ...

    trigger OnRun()
    begin
        readAPI();
    end;

    // ...

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

### Question: 3

#### HOTSPOT

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;

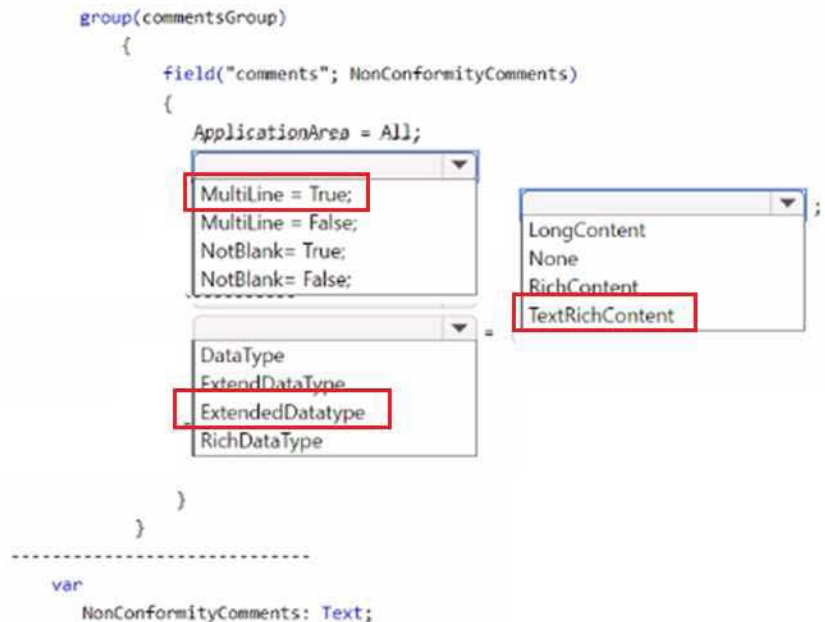
        DataType
        ExtendDataType
        ExtendedDatatype
        RichDataType

    }
}

var
    NonConformityComments: Text;
```

**Answer:**

#### ExtendedDataType property



#### ExtendedDataType property



### Question: 4

You need to define the data types for the fields of the Non-conformity table.  
Which two data types should you use? Each correct answer presents part of the solution.  
NOTE: Each correct selection is worth one point.

- A. Integer for the Non-conformity Number field
- B. Date Time for the Non-Conformity Date field
- C. Char for the Non-Conformity Number field
- D. Date for the Non-Conformity Date field
- E. Code for the Non-Conformity Number field

**Answer: C, E**

Explanation:

In Business Central, fields in tables are assigned specific data types that determine the kind of data they can store. For the Non-conformity table mentioned in the case study, the following data types should be used:

Date for the Non-Conformity Date field: This is because the Non-conformity Date field is required to store only the date when the non-conformity was recorded. The Date data type is appropriate for storing dates without times.

Code for the Non-Conformity Number field: The Non-conformity Number field is described to use alphanumeric values with a format that includes "NC" and the year, like "NC24-001". In Business Central, the Code data type is used for fields that store alphanumeric keys. It is a text field with a limited length, which makes it suitable for number series that contain letters and numbers.

Other options are not suitable:

A. Integer for the Non-conformity Number field: This would not be appropriate because the Nonconformity

Number includes alphanumeric characters and not just integers.

B. DateTime for the Non-Conformity Date field: This is not correct because there is no requirement to store the time alongside the date.

C. Char for the Non-Conformity Number field: Char data type is not typically used in Business Central for number series or identifiers. The Code data type is preferred for this purpose.

## Question: 5

HOTSPOT

You need to create the Install codeunit that is required in the extension used for installing or updating the Housekeeping app.

Which data type or declaration should you use? To answer, select the appropriate options in the answer area.

NOTE; Each correct selection is worth one point.

Data types or declarations for an Install codeunit

Requirement	Data type or declaration
Data type for information	<div>ModuleDependencyInfo</div> <div>ModuleInfo</div> <div>SessionInformation</div>
Start of the declaration of the method or procedure to perform the tasks	<div>global procedure</div> <div>local procedure</div> <div>procedure</div>

**Answer:**

Explanation:

For the Install codeunit required for the extension used for installing or updating the Housekeeping app, you should use the following data type and declaration:



Data type for information: ModuleInfo

Start of the declaration of the method or procedure to perform the tasks: local procedure

In AL language, which is used for developing extensions in Business Central, an Install codeunit is a special type of codeunit that is used to handle installation or upgrade logic for an extension.

ModuleInfo is a data type that contains information about the current extension, such as its version. It is typically used within the OnInstallAppPerCompany or OnUpgradePerCompany triggers of an Install codeunit to determine if the app is being installed for the first time or upgraded.

A local procedure within an Install codeunit is a method that is only accessible within the codeunit itself. It is not visible to other objects or extensions. This is suitable for tasks that are internal to the installation

process and should not be exposed globally.

These selections align with the requirements of handling installation and update procedures in a controlled and encapsulated manner within Business Central extensions.

**Thank You for Trying Our Product**

**Special 16 USD Discount Coupon: NSZUBG3X**

**Email:** [support@examsempire.com](mailto:support@examsempire.com)

**Check our Customer Testimonials and ratings  
available on every product page.**

**Visit our website.**

**<https://examsempire.com/>**