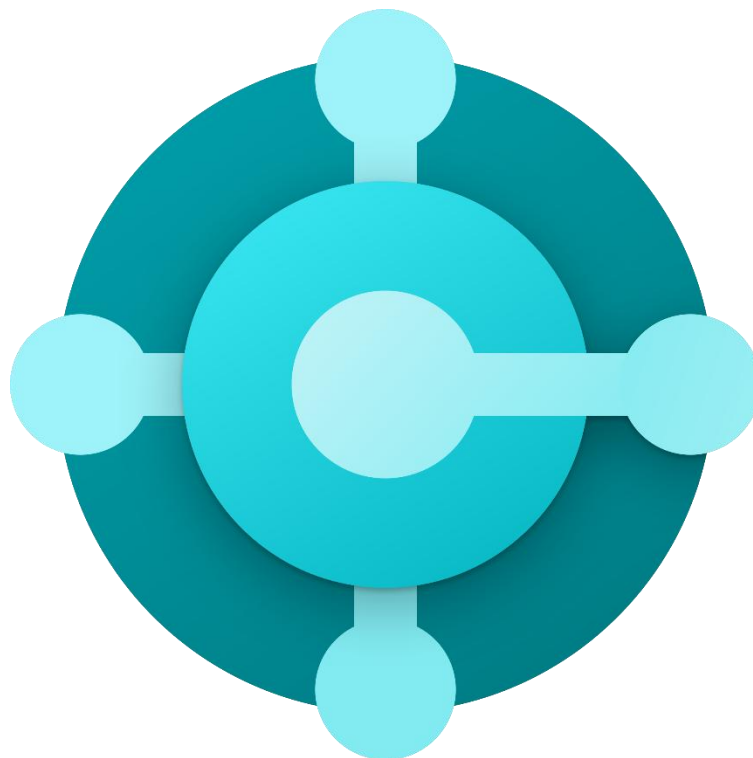


Manual:

Mandatory fields



Author: Alejandro Huerta

Date: 16/12/2025

Version: 1.0

<https://github.com/xandruv>

<https://www.linkedin.com/in/ahuertav/>

Index

1. INTRODUCTION	3
1.1 What does it include?	3
Configuration Tables	3
Codeunit	3
Pages	3
Page Extensions	3
1.2 Is it complex to add the validation to new pages?	4
Card-type pages (without lines)	4
Document-type pages (with lines)	4
1.3 Limitations	6
2. Setup	7
3. FUNCIONALITY	9

1. INTRODUCTION

The “Mandatory Fields” extension provides the ability to configure fields as mandatory on any page, so that the user is shown a warning when leaving the page, reminding them which fields should be completed.

1.1 What does it include?

The extension consists of the following components:

Configuration Tables

MFTableSetup: used to parameterize the tables to be monitored.

MFTableFieldSetup: used to parameterize the fields of the tables to be monitored.

Codeunit

MFFieldsControl: contains all the functions required to perform the validations.

Pages

MFTableSetup: page corresponding to the table of the same name.

MFTableFieldSetup: page corresponding to the table of the same name.

MFTableFieldSelector: page used to manage table fields.

Page Extensions

MFErroMessagesPart: extends the error management page to display to the user all the necessary information required to correct the error.

Extensions on commonly used pages:

MFCustomerCard: Extension of the Customer Card to add validation when closing the page.

MFIItemCard: Extension of the Item Card to add validation when closing the page.

MFVendorCard: Extension of the Vendor Card to add validation when closing the page.

Extension on pages with lines:

MFSalesOrder: extension of the Sales Order page to add validation when closing the page, both for the document header and its lines.

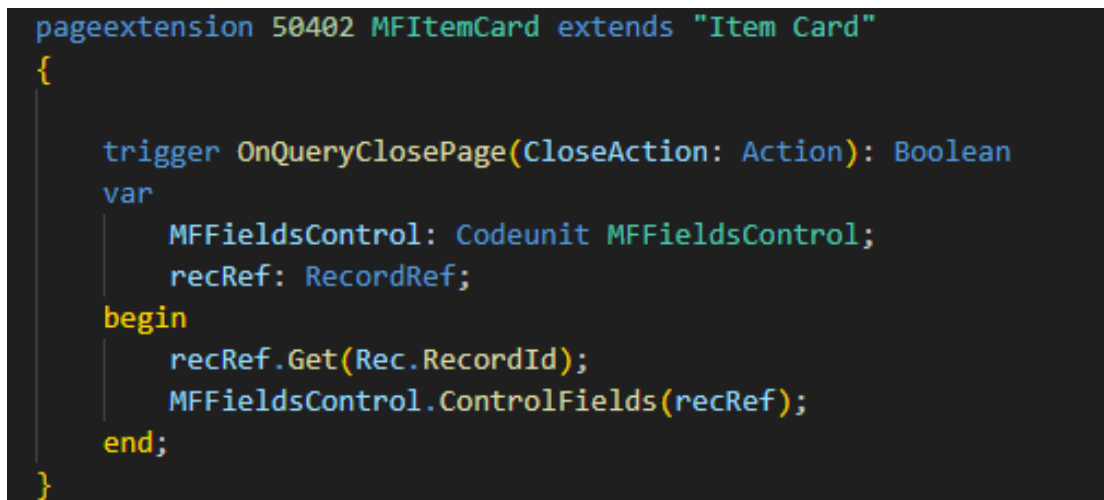
1.2 Is it complex to add the validation to new pages?

The extension has been designed to make the addition of new pages as simple as possible, enabling rapid implementation. You only need to create a new page extension and paste the code that matches the page type.

Card-type pages (without lines)

This type of page is very straightforward. Simply copy and paste the code into your page extension; the MFFieldsControl codeunit will take care of the rest.

```
trigger OnQueryClosePage(CloseAction: Action): Boolean
var
    MFFieldsControl: Codeunit MFFieldsControl;
    recRef: RecordRef;
begin
    recRef.Get(Rec.RecordId);
    MFFieldsControl.ControlFields(recRef);
end;
```



```
pageextension 50402 MFItemCard extends "Item Card"
{
    trigger OnQueryClosePage(CloseAction: Action): Boolean
    var
        MFFieldsControl: Codeunit MFFieldsControl;
        recRef: RecordRef;
    begin
        recRef.Get(Rec.RecordId);
        MFFieldsControl.ControlFields(recRef);
    end;
}
```

Document-type pages (with lines)

This type of page requires a slight adaptation. Copy and paste the code into your page extension. In this case, the salesLine variable must be adapted to the table that represents the document lines you want to validate, and the necessary filters must be applied.

```

trigger OnQueryClosePage(CloseAction: Action): Boolean
var
    MFFieldsControl: Codeunit MFFieldsControl;
    recRef: RecordRef;
    salesline: Record "Sales Line";
    lineErrors: Record "Error Message" temporary;
begin
    //Control for Header
    recRef.Get(Rec.RecordId);
    MFFieldsControl.ControlFields(recRef);
    salesline.Reset();
    salesline.SetRange("Document Type", Rec."Document Type");
    salesline.SetRange("Document No.", Rec."No.");
    if salesline.FindSet() then
        repeat
            //Control for each line
            recRef.Get(salesline.RecordId);
            MFFieldsControl.ControlFieldsLines(recRef, lineErrors);
        until salesline.Next() = 0;
    if lineErrors.Count > 0 then
        Page.RunModal(Page::"Error Messages Part", lineErrors)
    end;
end;

```

```

pageextension 50403 MFSalesOrder extends "Sales Order"
{
    trigger OnQueryClosePage(CloseAction: Action): Boolean
    var
        MFFieldsControl: Codeunit MFFieldsControl;
        recRef: RecordRef;
        salesline: Record "Sales Line";
        lineErrors: Record "Error Message" temporary;

    begin
        //Control for Header
        recRef.Get(Rec.RecordId);
        MFFieldsControl.ControlFields(recRef);
        salesline.Reset();
        salesline.SetRange("Document Type", Rec."Document Type");
        salesline.SetRange("Document No.", Rec."No.");
        if salesline.FindSet() then
            repeat
                //Control for each line
                recRef.Get(salesline.RecordId);
                MFFieldsControl.ControlFieldsLines(recRef, lineErrors);
            until salesline.Next() = 0;
        if lineErrors.Count > 0 then
            Page.RunModal(Page::"Error Messages Part", lineErrors)
    end;
}



```

1.3 Limitations

The extension allows mandatory fields to be parameterized by table, which is very useful for master data tables (items, customers, vendors, fixed assets, accounts, etc.). However, it does not support filters. As a result, for the validation of purchase and sales documents—where quote, order, and invoice tables are shared—it is not possible to apply different criteria to each document type.

2. Setup

Once the extension has been installed, access “Mandatory Fields Table Setup”.



Tell me what you want to do  

Go to Pages and Tasks

> **Mandatory Fields Table Setup** Lists

The tables for which mandatory fields need to be parameterized must be added.

Mandatory Fields Table Setup

  | [+ New](#) [Edit List](#) [Delete](#) [Fields Setup](#)





 **Fields Setup**

	Table No ↑		Table Name
→	18	⋮	Customer
	23		Vendor
	27		Item
	36		Sales Header
	37		Sales Line

Once the tables have been added, access the “Field Setup” action, where it will be possible to configure the fields of the table that are to be set as mandatory.

Mandatory Fields Table Field Setup

  | [+ New](#) [Edit List](#) [Delete](#)

	Field No ↑		Field Name
	27		Payment Terms Code
	31		Shipping Agent Code
→		 ⋮	

Drill down to record for Field No

To facilitate field management, access the field configuration through the assist edit (...) of the “Field No.” field. This will open the field management page.

Add fields

Select the fields you want to add or remove as mandatory and choose the corresponding action: “Add Fields” or “Remove Fields.”

3. FUNCTIONALITY

When leaving a page to which this functionality has been added, the system will check whether the fields configured as mandatory have been correctly filled. If not, the user will be shown a list of the fields that should be completed on the page.

The screenshot shows the 'Customer Card' for '20000 · Trey Research'. The 'General' tab is active, displaying fields for No. (20000), Name (Trey Research), and Balance (LCY). An 'Error Messages - 2' overlay is shown, listing two errors:

Message Type ↑	Additional Information	Field Name	Description
Error	Customer: 20000	Language Code	Field Language Code can not be empty
→ Error	Customer: 20000	Shipping Agent Code	Field Shipping Agent Code can not be empty

The process does not block the user; it only displays, for informational purposes, the fields that are still pending completion.

In the case of pages with lines, two error lists will be displayed: the first for the header fields, and the second for the line fields, identifying for each line which fields are still pending completion.

The screenshot shows the 'Sales Order' for '101009 · Adatum Corporation'. The 'General' tab is active, displaying the Customer No. (10000). An 'Error Messages - 1' overlay is shown, listing one error:

Messa... Type ↑	Additional Information	Field Name	Description
→ Error	Sales Header: Order,101009	Requested Delivery Date	Field Requested Delivery Date can not be empty

The screenshot shows the 'Sales Order' for '101009 · Adatum Corporation'. The 'Lines' tab is active, displaying a table with columns: Type, No., Item Reference No., and Description. An 'Error Messages - 3' overlay is shown, listing three errors:

Messa... Type ↑	Additional Information	Field Name	Description
Error	Sales Line: Order,101009,10000	Location Code	Field Location Code can not be empty
Error	Sales Line: Order,101009,30000	Location Code	Field Location Code can not be empty
→ Error	Sales Line: Order,101009,30000	Unit Price	Field Unit Price Excl. VAT can not be empty