

Hinzufügen eines Feldes in der Validierung (ab Version 2.10)

Im Folgenden wird anhand einer Beispielextension dargestellt, wie man als Entwickler Felder in der Validierung hinzufügen kann (hier: Dokumentenklasse Rechnung/Gutschrift).

Optionale Integration Events inkl. beispielhafter Prozeduren:

```
codeunit 50100 EventSubs
{
    //
    // [If you want to perform plausibility checks on the newly added header field, this is
the place]
    //
    [EventSubscriber(ObjectType::Codeunit, Codeunit::"DXP SQZ P. Inv/Crdt Memo Impl.",
'OnAfterDoHeaderPlausibilityChecks', '', false, false)]
    local procedure SQZPInvCrdtMemoImplOnBeforeDoHeaderPlausibilityChecks(DocHeader: Record
"DXP SQZ Document Header"; var PlausibilityCheck: Codeunit "DXP Plausiblity Check Mgt.")
    begin
        CheckCustomer(PlausibilityCheck, DocHeader."Customer No.");
    end;

    //
    // [If you want to perform plausibility checks on the newly added line field, this is the
place]
    //
    [EventSubscriber(ObjectType::Codeunit, Codeunit::"DXP SQZ P. Inv/Crdt Memo Impl.",
'OnAfterDoLinePlausibilityChecks', '', false, false)]
    local procedure SQZPInvCrdtMemoImplOnAfterDoLinePlausibilityChecks(DocHeader: Record "DXP
SQZ Document Header"; DocLine: Record "DXP SQZ Document Line"; var PlausibilityCheck: Codeunit
"DXP Plausiblity Check Mgt.")
    begin
        CheckCustomer(PlausibilityCheck, DocLine."Customer No.");
    end;

    [TryFunction]
```

```

local procedure CustomerExists(CustomerNo: Code[20])
var
    Customer: Record Customer;
begin
    Customer.Get(CustomerNo);
end;

local procedure CheckCustomer(var PlausibilityCheck: Codeunit "DXP Plausibility Check
Mgt."; CustomerNo: Code[20]): Boolean
begin
    if CustomerNo = '' then
        exit(false);

    if not CustomerExists(CustomerNo) then begin
        PlausibilityCheck.AddPlausibilityCheckEntry(GetLastErrorText(), Page::"Customer
List");
        exit(false);
    end;

    exit(true);
end;
}

```

Table Extensions:

SQUEEZE 4 BC

```

tableextension 50100 "SQZ Doc. Header Ext." extends "DXP SQZ Document Header"
{
    fields
    {
        field(50100; "Customer No."; Code[20])
        {
            TableRelation = Customer;
            ValidateTableRelation = false;
            Caption = 'Customer No.';
        }
    }
}
}

```

```

tableextension 50101 "SQZ Doc. Line Ext." extends "DXP SQZ Document Line"
{
    fields
    {
        field(50100; "Customer No."; Code[20])
        {
            TableRelation = Customer;
            ValidateTableRelation = false;
            Caption = 'Customer No.';
        }
    }
}

```

Zielbeleg (hier: Einkaufsbeleg)

```

tableextension 50102 "Purchase Header Ext." extends "Purchase Header"
{
    fields
    {
        field(50100; "DXP Customer No."; Code[20])
        {
            DataClassification = CustomerContent;
            TableRelation = Customer;
            Caption = 'DEXPRO Customer No.';
        }
    }
}

```

Page Extensions:

SQUEEZE 4 BC

```

pageextension 50100 "SQZ Document Ext." extends "DXP SQZ Document v2"
{
    layout
    {
        addafter(BuyFromVendorInternal)
        {
            field("Customer No. Internal"; Rec."Customer No.")

```

```

    {
        ApplicationArea = All;
        Tooltip = 'Specifies the value of the Customer No. field.';
        //The following code is optional. It is used to highlight a recognized value in
the Squeeze Viewer
        trigger OnAssistEdit()
        begin
            MarkField(Rec.FieldNo("Customer No."), Rec."Customer No.", Rec);
        end;

        trigger OnValidate()
        begin
            CheckPlausibility();
        end;
    }
}
addafter(BuyFromVendorExternal)
{
    field("Customer No."; Rec."Customer No.")
    {
        ApplicationArea = All;
        Tooltip = 'Specifies the value of the Customer No. field.';
        //The following code is optional. It is used to highlight a recognized value
in the Squeeze Viewer
        trigger OnAssistEdit()
        var
            ApiMgt: Codeunit "DXP SQZ API Mgt.";
            ViewerResident: Codeunit "DXP SQUEEZE Viewer Resident";
        begin
            MarkField(Rec.FieldNo("Customer No."), Rec."Customer No.", Rec);
        end;

        trigger OnValidate()
        begin
            CheckPlausibility();
        end;
    }
}
}

```

```

local procedure MarkField(AppFldNo: Integer; FieldVal: Variant; DocHeader: Record "DXP SQZ
Document Header")
var
    ApiMgt: Codeunit "DXP SQZ API Mgt.";
    ViewerResident: Codeunit "DXP SQUEEZE Viewer Resident";
begin
    ViewerResident := GetViewerResident();
    ApiMgt.MarkField(AppFldNo, FieldVal, ViewerResident, DocHeader);
end;
}

```

```

pageextension 50101 "SQZ Document Sf. Ext." extends "DXP SQZ Document Subform"
{
    layout
    {
        addafter("No.")
        {
            field("Customer No."; Rec."Customer No.")
            {
                ApplicationArea = All;
                Tooltip = 'Specifies the value of the Customer No. field.';
            }
        }
    }
}

```

Zielbeleg (hier: Einkaufsbeleg)

```

pageextension 50102 "Purchase Invoice Ext." extends "Purchase Invoice"
{
    layout
    {
        addafter("Buy-from Vendor No.")
        {
            field("DXP Customer No."; Rec."DXP Customer No.")
            {
                Tooltip = 'Specifies the value of the DEXPRO Customer No. field.';
                ApplicationArea = all;
            }
        }
    }
}

```

```
}  
}
```

```
pageextension 50103 "Purchase Inv. Sf Ext." extends "Purch. Invoice Subform"  
{  
  layout  
  {  
    addafter("No.")  
    {  
      field("DXP Customer No."; Rec."DXP Customer No.")  
      {  
        ToolTip = 'Specifies the value of the DEXPRO Customer No. field.';  
        ApplicationArea = all;  
      }  
    }  
  }  
}
```

Revision #4

Created 2024-02-06 17:40:36 UTC by Bernd Feddersen

Updated 2026-01-29 09:10:04 UTC by Bernd Feddersen