Adding a field in the validation (from version 2.10)

The following example extension shows how developers can add fields in the validation (here: document class invoice/credit note).

Optional integration events incl. sample procedures:

```
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 Plausiblity 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.';
        }
    }
}
```

Target document (here: Purchase Document)

```
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.';
            }
        }
    }
}
```

Target document (here: Purchase document)

```
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 #1
Created 2024-02-13 13:38:15 UTC by Christoph Koepke-von Seth
Updated 2026-01-29 09:11:15 UTC by Christoph Koepke-von Seth