

# Extensibility for Billing Document Printout

S/4 HANA Cloud



**THE BEST RUN**



# Table of Contents

<b>1</b>	<b><i>Provide Sales Order Reason in Billing Document Output</i></b>	<b>3</b>
1.1	Aim	3
1.2	Steps to do	3
1.2.1	Creation of an extension field at Billing Document	3
1.2.2	Extend the Form Template with the Extension Field	3
1.2.3	Creation of an extension logic	4
<b>2</b>	<b><i>Provide Sales Order Reason Text in Billing Document Output</i></b>	<b>6</b>
2.1	Aim	6
2.2	Steps to do	6
2.2.1	Creation of an extension field at Billing Document	6
2.2.2	Extend the Form Template with the Extension Field	6
2.2.3	Creation of an extension logic	7

## 1 Provide Sales Order Reason in Billing Document Output

### 1.1 Aim

Show Sales Order Reason on Form Template Item of Billing Document.

### 1.2 Steps to do

#### 1.2.1 Creation of an extension field at Billing Document

Use app [Custom Fields and Logic](#), tab [Custom Fields](#), in the Extensibility business catalog. Create a new extension field in [Business Context Sales: Billing Document](#) with an appropriate Type.

New Field

Field Properties

\*Business Context: Sales: Billing Document

\*Label: salesorderreason

\*Identifier: YY1\_ salesorderreason

\*Tooltip: salesorderreason

\*Type: Text

\*Length: 3

Business Context Capacity: 10%

Create Create and Edit Cancel

This extension field is intended to store the Sales Order Reason with the Billing Document.

*Remark: The extension field will only be filled when triggering the output request.*

#### 1.2.2 Extend the Form Template with the Extension Field

Within the same App [Custom Fields and Logic](#) and tab [Custom Field](#) enable the Form Templates for the Billing Document with the extension field.

The screenshot shows the SAP 'Custom Fields and Logic' interface. The main header is 'salesorderreason'. Below it, there's a list of items: 'Text (3)', 'YY1\_salesorderreason', and 'Sales: Billing Document'. The 'Form Templates' tab is selected, showing a table of templates. The table has columns for 'Description' and 'Field Usage'. The 'Field Usage' column shows 'Enabled' for all templates, with a 'Disable Usage' button next to each. The templates listed are 'Credit Memo Form' and 'Credit Memo Form' for various countries (Austria, Belgium, France, Hungary, Indonesia, Ireland, Italy, Luxembourg, Netherlands, New Zealand). At the bottom right, there are buttons: 'Save', 'Publish', 'Discard Changes', 'Delete', and 'Cancel'.

Description	Field Usage
Credit Memo Form	Enabled <button>Disable Usage</button>
Credit Memo Form (Austria)	Enabled <button>Disable Usage</button>
Credit Memo Form (Belgium)	Enabled <button>Disable Usage</button>
Credit Memo Form (France)	Enabled <button>Disable Usage</button>
Credit Memo Form (Hungary)	Enabled <button>Disable Usage</button>
Credit Memo Form (Indonesia)	Enabled <button>Disable Usage</button>
Credit Memo Form (Ireland)	Enabled <button>Disable Usage</button>
Credit Memo Form (Italy)	Enabled <button>Disable Usage</button>
Credit Memo Form (Luxembourg)	Enabled <button>Disable Usage</button>
Credit Memo Form (Netherlands)	Enabled <button>Disable Usage</button>
Credit Memo Form (New Zealand)	Enabled <button>Disable Usage</button>

Save and publish the extension field.

### 1.2.3 Creation of an extension logic

Use app *Custom Fields and Logic*, tab *Custom Logic*, in the Extensibility business catalog. Create a new Enhancement Implementation in *Business Context Sales: Billing Document* with *Definition Description Billing Standard Output Header Adaption* and an appropriate Implementation Description. This step can be skipped in case the Enhancement Implementation already exist.

The screenshot shows the 'New Enhancement Implementation' dialog box. It has four main sections: 'Business Context:', '\*BApI Description:', '\*Implementation Description:', and '\*Implementation ID:'. The 'Business Context' dropdown is set to 'Sales: Billing Document'. The '\*BApI Description' dropdown is set to 'Billing Standard Output Header Adaption'. The '\*Implementation Description' text field contains 'Enhance Print Program'. The '\*Implementation ID' section has a prefix 'YY1\_' and a text field containing 'ENHANCEPRINTPROGRAM'. At the bottom right, there are 'Create' and 'Cancel' buttons.

Add the following example code to the Draft:

\* This method can be used to set values for output to billing document header extension fields.

\* First of all assign existing extension field values to output parameter:

```
billingdoc_extension_out = billingdoc_extension_in.  
billingdocdescr_extension_out = billingdocdescr_extension_in.  
  
if billingdoc-sddocumentcategory CA 'OP'.  
  
    select single sddocumentreason  
    from i_billingdocumentitem  
    into @billingdoc_extension_out-yy1_salesorderreason_bdh  
    where billingdocument = @billingdoc-billingdocument.  
endif.
```

Save and publish the extension Logic.

The screenshot shows the SAP Custom Fields and Logic editor interface. The top navigation bar includes the SAP logo, navigation icons, and the title 'Custom Fields and Logic'. On the right, there are search and list icons. Below the navigation bar, there are tabs for 'Create Draft', 'Published' (which is active), and 'Example'. The main area displays the 'Published Logic' code, which is a list of 26 lines. Lines 1-3 are comments. Lines 4-6 are another set of comments. Lines 7-14 contain the code snippet provided in the previous block, which is highlighted with a yellow box. Lines 15-26 contain additional code for handling the 'if sy-subrc = 0' condition. At the bottom, there is a 'Test Results: Published Logic' section, and a footer bar with 'Save Draft', 'Delete', 'Test', and 'Publish' buttons.

*Remark: To see the new created extension field in the Billing Document Printout, it is necessary to create a new Form Template.*

## 2 Provide Sales Order Reason Text in Billing Document Output

### 2.1 Aim

Show Sales Order Reason Text on Form Template Item of Billing Document.

### 2.2 Steps to do

#### 2.2.1 Creation of an extension field at Billing Document

Use app [Custom Fields and Logic](#), tab [Custom Fields](#), in the Extensibility business catalog. Create a new extension field in [Business Context Sales: Billing Document](#) with an appropriate Type.

New Field

Field Properties

\*Business Context: Sales: Billing Document

\*Label: salesorderreasontext

\*Identifier: YY1\_ salesordreasontxt

\*Tooltip: salesorderreasontext

\*Type: Text

\*Length: 15

Business Context Capacity: 11%

Create Create and Edit Cancel

This extension field is intended to store the Sales Order Reason Text with the Billing Document.

*Remark: The extension field will only be filled when triggering the output request.*

#### 2.2.2 Extend the Form Template with the Extension Field

Within the same App [Custom Fields and Logic](#) and tab [Custom Field](#) enable the Form Templates for the Billing Document with the extension field.

The screenshot shows the SAP 'Custom Fields and Logic' interface. The breadcrumb trail is 'salesorderreasoncontext'. Below it, the text 'Text (15)' and 'YY1\_salesorderreasontxt' are visible, along with 'Sales: Billing Document'. The top navigation bar includes 'General Information', 'UIs and Reports (8)', 'Email Templates (4)', 'Form Templates (54)', 'Business Scenarios (6)', 'OData APIs (2)', 'SOAP APIs (2)', 'BAPIs (0)', and 'IDocs (0)'. The 'Form Templates' tab is active, displaying a table with columns 'Description' and 'Field Usage'. The table lists various 'Credit Memo Form' templates for different countries, all with 'Enabled' status and a 'Disable Usage' button. At the bottom right, there are buttons for 'Save', 'Publish', 'Discard Changes', 'Delete', and 'Cancel'.

Description	Field Usage
Credit Memo Form	Enabled <button>Disable Usage</button>
Credit Memo Form (Austria)	Enabled <button>Disable Usage</button>
Credit Memo Form (Belgium)	Enabled <button>Disable Usage</button>
Credit Memo Form (France)	Enabled <button>Disable Usage</button>
Credit Memo Form (Hungary)	Enabled <button>Disable Usage</button>
Credit Memo Form (Indonesia)	Enabled <button>Disable Usage</button>
Credit Memo Form (Ireland)	Enabled <button>Disable Usage</button>
Credit Memo Form (Italy)	Enabled <button>Disable Usage</button>
Credit Memo Form (Luxembourg)	Enabled <button>Disable Usage</button>
Credit Memo Form (Netherlands)	Enabled <button>Disable Usage</button>
Credit Memo Form (New Zealand)	Enabled <button>Disable Usage</button>

Save and publish the extension field.

### 2.2.3 Creation of an extension logic

Use app *Custom Fields and Logic*, tab *Custom Logic*, in the Extensibility business catalog. Create a new Enhancement Implementation in *Business Context* **Sales: Billing Document** with *Definition Description* **Billing Standard Output Header Adaption** and an appropriate Implementation Description. This step can be skipped in case the Enhancement Implementation already exist.

The screenshot shows the 'New Enhancement Implementation' dialog box. It contains the following fields: 'Business Context' with a dropdown menu showing 'Sales: Billing Document'; '\*BApI Description:' with a dropdown menu showing 'Billing Standard Output Header Adaption'; '\*Implementation Description:' with a text input field containing 'Enhance Print Program'; and '\*Implementation ID:' with a text input field containing 'YY1\_ ENHANCEPRINTPROGRAM'. At the bottom right, there are 'Create' and 'Cancel' buttons.

Add the following example code to the Draft:

```
* This method can be used to set values for output to billing document header extension fields.

* First of all assign existing extension field values to output parameter:

billingdoc_extension_out = billingdoc_extension_in.
billingdocdescr_extension_out = billingdocdescr_extension_in.

if billingdoc-sddocumentcategory CA 'OP'.

    select single sddocumentreason
    from i_billingdocumentitem
    into @billingdoc_extension_out-yy1_salesorderreason_bdh
    where billingdocument = @billingdoc-billingdocument.

    if sy-subrc = 0.

        select single sddocumentreasontext
        from i_sddocumentreasontext
        into @billingdoc_extension_out-yy1_salesordreasontxt_bdh
        where sddocumentreason = @billingdoc_extension_out-yy1_salesorderreason_bdh
        and language = @sy-langu.

    endif.
endif.
```

Save and publish the extension Logic.

The screenshot shows the SAP Custom Fields and Logic editor interface. At the top, there is a navigation bar with the SAP logo and a dropdown menu for 'Custom Fields and Logic'. Below this, there is a tabbed interface with 'Create Draft', 'Published', and 'Example' tabs. The 'Published' tab is active, displaying a list of 'Published Logic' items. The first item is selected, showing a preview of the logic code. The code is a SQL SELECT statement that assigns values to output parameters based on the input parameters and the billing document category. The code is as follows:

```
1 * This method can be used to set values for output to billing document header extension fields.
2 * Example:
3 *
4 * First of all assign existing extension field values to output parameter:
5 billingdoc_extension_out = billingdoc_extension_in.
6 billingdocdescr_extension_out = billingdocdescr_extension_in.
7
8 * if billingdoc-sddocumentcategory CA 'OP'.
9
10     select single sddocumentreason
11     from i_billingdocumentitem
12     into @billingdoc_extension_out-yy1_salesorderreason_bdh
13     where billingdocument = @billingdoc-billingdocument.
14
15
16     if sy-subrc = 0.
17
18         select single sddocumentreasontext
19         from i_sddocumentreasontext
20         into @billingdoc_extension_out-yy1_salesordreasontxt_bdh
21         where sddocumentreason = @billingdoc_extension_out-yy1_salesorderreason_bdh
22         and language = @sy-langu.
23
24     endif.
25 endif.
```

Below the code preview, there is a section for 'Test Results: Published Logic'. At the bottom of the editor, there is a dark blue bar with buttons for 'Save Draft', 'Delete', 'Test', and 'Publish'.

**Remark:** To see the new created extension field in the Billing Document Printout, it is necessary to create a new Form Template.