

Extensibility for Billing Document Printout

S/4 HANA Cloud



THE BEST RUN



Table of Contents

1	<i>Provide Customer Material ID in Billing Document Output</i>	3
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
2	<i>Provide Net Weight and Unit in Billing Document Output</i>	6
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 Customer Material ID in Billing Document Output

1.1 Aim

Show Customer Material ID on Form Template Item of Billing Document Item.

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 Billing Document Item* with an appropriate Type.

The screenshot shows a 'New Field' configuration window. The 'Field Properties' section includes the following fields:

- *Business Context: Sales: Billing Document Item
- *Label: CustomerMaterialID
- *Identifier: YY1_ CustomerMaterialID
- *Tooltip: CustomerMaterialID
- *Type: Text
- *Length: 15

Business Context Capacity: 7%

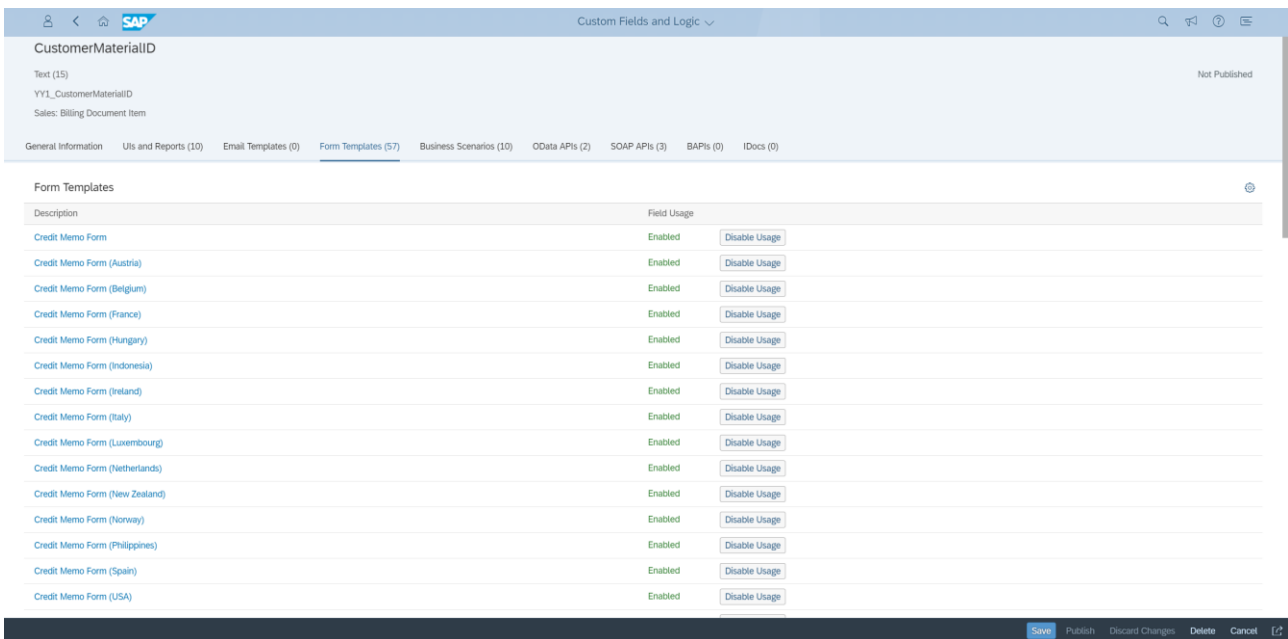
Buttons: Create, Create and Edit, Cancel

This extension field is intended to store the Customer Material ID with the Billing Document Item.

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.



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 Item* with *Definition Description Billing Standard Output Item Adaption* and an appropriate Implementation Description. In case the Enhancement Implementation already exist, this step can be skipped.

The screenshot shows the 'New Enhancement Implementation' dialog box. It contains the following fields:

- Business Context:** A dropdown menu with 'Sales: Billing Document Item' selected.
- *BAdI Description:** A dropdown menu with 'Billing Standard Output Item Adaption' selected.
- *Implementation Description:** A text input field containing 'Enhance Print Program'.
- *Implementation ID:** A text input field containing 'YY1_ ENHANCEPRINTPROGRAM'.

At the bottom of the dialog, there are two buttons: 'Create' (highlighted in blue) and 'Cancel'.

Add the following example code to the Draft:

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

* Example:

*

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

```
billingdocitem_ext_out = billingdocitem_ext_in.  
billingdocitemdescr_ext_out = billingdocitemdescr_ext_in.
```

*get SO number

```
select single salesdocument  
  from i_billingdocumentitem  
  into @data(lv_salesdocument)  
  where billingdocument = @billingdocitem-billingdocument.
```

*get Customer Material ID

```
select single MaterialByCustomer  
  from i_salesdocumentitem  
  into @billingdocitem_ext_out-yy1_customermaterialid_bdi  
  where salesdocument = @lv_salesdocument.
```

Save and publish the extension Logic.

The screenshot shows the SAP Custom Fields and Logic editor interface. The title bar indicates 'Custom Fields and Logic'. The main content area is titled 'Enhance Print Program' and shows 'Last Changed By _SAPD064591, 8/14/2019, 5:58:38 AM'. There are buttons for 'Not Published' and 'Draft Available'. Below this, there are tabs for 'Available Fields', 'Filter', and 'BAdI Documentation'. A 'Select Variant' button is also present. The main area contains four input fields for mapping: 'billingdoc:' and 'billingdocitem:' (both with 'Click to add value' buttons), and 'billingdocitem_ext_in:' and 'billingdocitemdescr_ext_in:' (both with 'Click to add value' buttons). Below the input fields, there is a 'Draft Logic' section with a code editor containing the following code:

```
1 This method can be used to set values for output to billing document item extension fields.  
2 -  
3 - Example  
4 -  
5 - First of all assign existing extension field values to output parameter:  
6 -  
7 billingdocitem_ext_out = billingdocitem_ext_in.  
8 billingdocitemdescr_ext_out = billingdocitemdescr_ext_in.  
9 -  
10 *get SO number  
11 select single salesdocument  
12   from i_billingdocumentitem  
13   into @data(lv_salesdocument)  
14   where billingdocument = @billingdocitem-billingdocument.  
15 -  
16 *get Customer Material ID  
17 select single MaterialByCustomer  
18   from i_salesdocumentitem  
19   into @billingdocitem_ext_out-yy1_customermaterialid_bdi  
20   where salesdocument = @lv_salesdocument.  
21  
22
```

At the bottom right of the editor, there are 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.

2 Provide Net Weight and Unit in Billing Document Output

2.1 Aim

Show Net Weight and Unit on Form Template Item of Billing Document Item.

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 Billing Document Item* with an appropriate Type.

The screenshot shows a 'New Field' dialog box with the following configuration:

- Business Context:** Sales: Billing Document Item
- Label:** NetWeight
- Identifier:** YY1_ NetWeight
- Tooltip:** NetWeight
- Type:** Quantity with Unit
- Business Context Capacity:** 8%

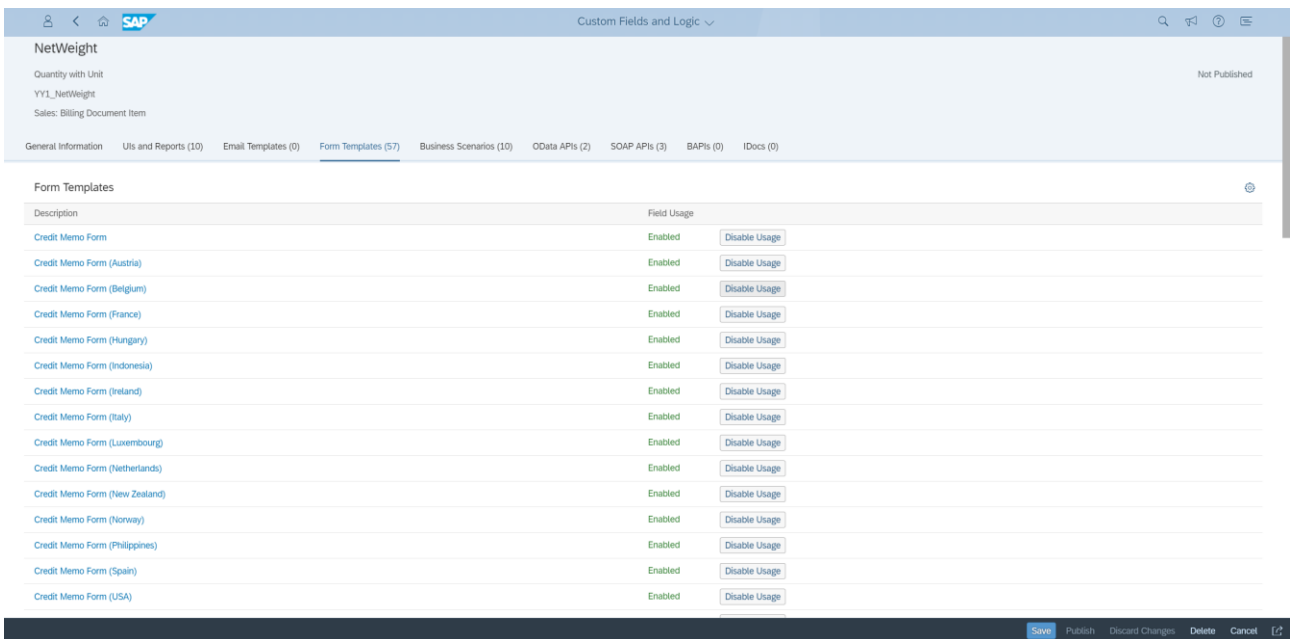
Buttons at the bottom: Create, Create and Edit, Cancel

This extension field is intended to store the Net Weight and Unit with the Billing Document Item.

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.



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 Item* with *Definition Description Billing Standard Output Item Adaption* and an appropriate Implementation Description. In case the Enhancement Implementation already exist, this step can be skipped.

New Enhancement Implementation

Business Context:

*BAdI Description:

*Implementation Description:

*Implementation ID:

Add the following example code to the Draft:

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

* Example:

*

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

```
billingdocitem_ext_out = billingdocitem_ext_in.  
billingdocitemdescr_ext_out = billingdocitemdescr_ext_in.
```

*get Net Weight

```
select single itemnetweight,itemweightunit  
from i_billingdocumentitem  
into @data(ls_weight)  
where billingdocument = @billingdocitem-billingdocument.
```

```
billingdocitem_ext_out-yy1_netweight_bdi = ls_weight-itemnetweight.  
billingdocitem_ext_out-yy1_netweight_bdiu = ls_weight-itemweightunit.
```

Save and publish the extension Logic.

The screenshot shows the SAP Custom Fields and Logic configuration interface. The 'Published Logic' section contains the following ABAP code:

```
2 *  
3 * Example:  
4 * First of all assign existing extension field values to output parameter:  
5  
6 billingdocitem_ext_out = billingdocitem_ext_in.  
7 billingdocitemdescr_ext_out = billingdocitemdescr_ext_in.  
8  
9 *get SD number  
10 select single salesdocument  
11 from i_billingdocumentitem  
12 into @data(lv_salesdocument)  
13 where billingdocument = @billingdocitem-billingdocument.  
14  
15 *get Customer Material ID  
16 select single MaterialByCustomer  
17 from i_salesdocumentitem  
18 into @data(lv_customermaterialid_bdi)  
19 where salesdocument = @lv_salesdocument.  
20  
21  
22 *get Net Weight  
23 select single itemnetweight,itemweightunit  
24 from i_billingdocumentitem  
25 into @data(ls_weight)  
26 where billingdocument = @billingdocitem-billingdocument.  
27  
28 billingdocitem_ext_out-yy1_netweight_bdi = ls_weight-itemnetweight.  
29 billingdocitem_ext_out-yy1_netweight_bdiu = ls_weight-itemweightunit.  
30  
31
```

The 'Test Results: Published Logic' table shows the following data:

Parameter	Description	Value	Type
> billingdocitem_ext_out	Extension Include persistent (VBRP)	[structure]	sdbillingdocitem_incl_eew_ps
> billingdocitemdescr_ext_out		[structure]	sdbillingdocitem_incl_eew_tr

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