



**PUBLIC**

# **How to Design GEP, BPP, and BFP File Formats in Electronic File Manager: Format Definition**

**Applicable Releases:**

**SAP Business One 8.81**

**SAP Business One 8.82**

**All Countries**

**English**

**May 2012**

## Table of Contents

<b>Introduction.....</b>	<b>3</b>
Purpose .....	4
Designing a File Format .....	4
<b>Designing a GEP File Format .....</b>	<b>6</b>
Creating a File Format to Export Business Partner Data from SAP Business One .....	6
Defining Mapping Settings.....	9
Creating Runtime Validation Rules for Target Nodes .....	12
<b>Designing a BPP File Format .....</b>	<b>21</b>
Creating a BPP File Format with TXT as the Target File .....	22
Defining Mapping Settings.....	25
<b>Designing a BFP File Format.....</b>	<b>28</b>
Creating a BFP File Format.....	28
Defining Conditions for Value Output .....	32
Defining Mapping Settings.....	32
Target SAP Business One Database Tables and Fields for Mapping .....	34
<b>Other Reference Material .....</b>	<b>36</b>
<b>Copyrights, Trademarks, and Disclaimers .....</b>	<b>37</b>

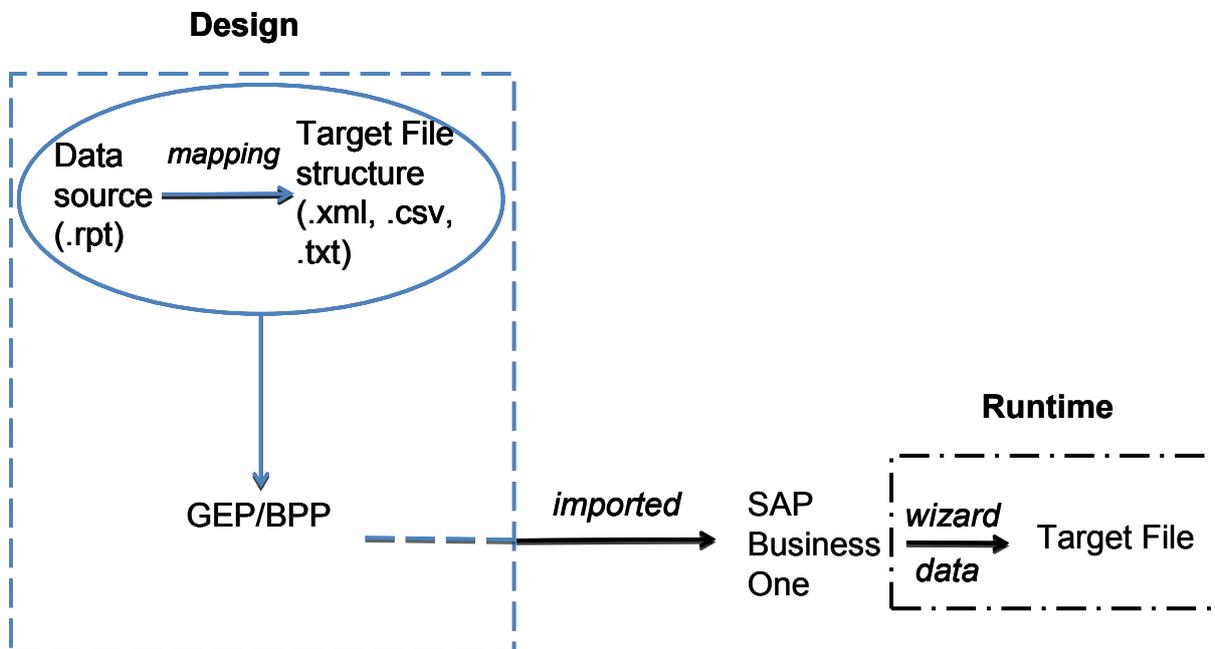
## Introduction

The Electronic File Manager: Format Definition (EFM) add-on enables you to design file formats that can be imported into SAP Business One for converting retrieved data into required formats for various uses. EFM provides the following four solutions:

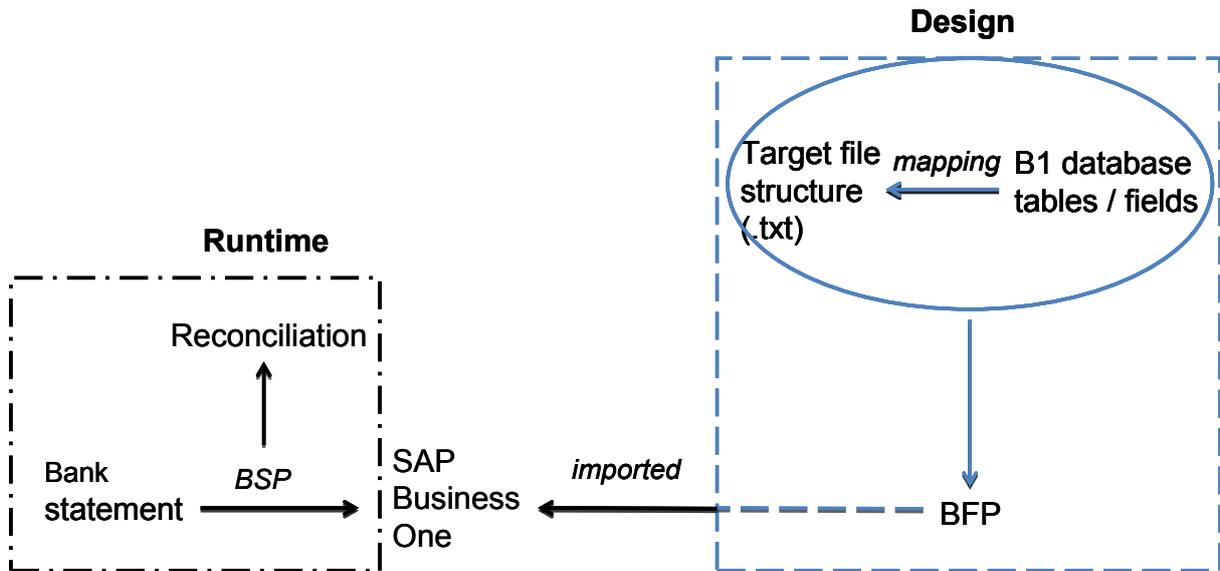
- Generic Electronic File Format Project (GEP)
- Bank Payment File Format Project (BPP)
- Bank Statement File Format Project (BFP)
- Special-Purpose File Format Project (SPP)

The following two figures illustrate the design and runtime of EFM for GEP, BPP, and BFP file formats.

### GEP and BPP File Formats



## BFP File Format



## Purpose

This how-to guide provides three examples of designing GEP, BPP, and BFP file formats. For designing SPP file formats, refer to the GEP and BPP examples.



### Note

These examples describe only the necessary steps to meet the designing purposes and omit some optional steps. Once you become more familiar with EFM, you can change the order of some of the steps.

## Designing a File Format

### Prerequisites

- You have started SAP Business One.
- You have installed and started the Electronic File Manager: Format Definition add-on.



### Note

For the BFP file format, mastery of regular expressions is mandatory.

For the GEP, BPP, and SPP file formats, mastery of the XSLT (Extensible Stylesheet Language Transformations) functions is optional; however, it is useful for more advanced needs or more complicated designs.

For more information, see the online help of the Electronic File Manager: Format Definition add-on, the online help of the Format Definition add-on, and SAP Note [1531223](#).

### Preparation

Before designing a file format, make sure to do the following:

- **Analyze and clarify your designing purpose**

You need to keep your purpose in mind:

- GEP and BPP: What information do you need to retrieve from the SAP Business One database?

- BFP: What information do you need to import into SAP Business One?

For the BPP or BFP file format, you need to study the specification defined by the bank that receives bank files from or issues bank statements to its clients. For the GEP file format, you need to prepare your own specification.

Based on your designing purpose, you decide on the following:

- The target file format structure
- The fields to add to the target file format structure
- The mapping settings, conditions, and validation rules

- **Prepare raw materials**

For the GEP file format, you need to prepare the following:

- One or more source files (.rpt)
- One target file (.xsd, .xml, or .csv) if you do not create the target format structure from scratch

For the BPP file format, you need to prepare the following :

- One or more source files (.rpt) if the pre-defined source file *Payment Wizard Run Results* is insufficient
- One target file (.xsd, .xml, or .csv) if you do not create the target format structure from scratch

For the BFP file format, you might like to acquire a sample .txt file from the bank for test running the file format. For more information, see step 12 of [Creating a BFP File Format](#).

Some sample files are attached to this guide and referred to in the examples. These files include:

- GEP: BPList.rpt (source file), BPList.xsd and BPList.xml (target files), BPList.GEP (completed GEP file format)
- BPP: OPEX.rpt (extra source file), Sample\_SAPBPFROPBT\_AFB\_DO.BPP (completed BPP file format)
- BFP: Sample\_csb43.BFP (completed BFP file format), csb43.txt (preview file)

For more information about creating .rpt source files, see the online help of the Electronic File Manager: Format Definition add-on and the guide *How to Work with SAP Crystal Reports in SAP Business One*.

To design a file format, do the following:

1. Add one or more source files and only one target file to the file format (not relevant to the BFP file format).
2. Add nodes to the target file structure, if appropriate.
3. Define properties or attributes of the file format and tree nodes.
4. Define mapping settings and conditions.
5. Add runtime validation rules (not relevant to the BFP file format).
6. Verify the file format.
7. Conduct a test run.

## Designing a GEP File Format

GEP file formats are designed to export various data from SAP Business One using wizards. The following example describes how to design a GEP file format for exporting business partner data from SAP Business One.

### Creating a File Format to Export Business Partner Data from SAP Business One

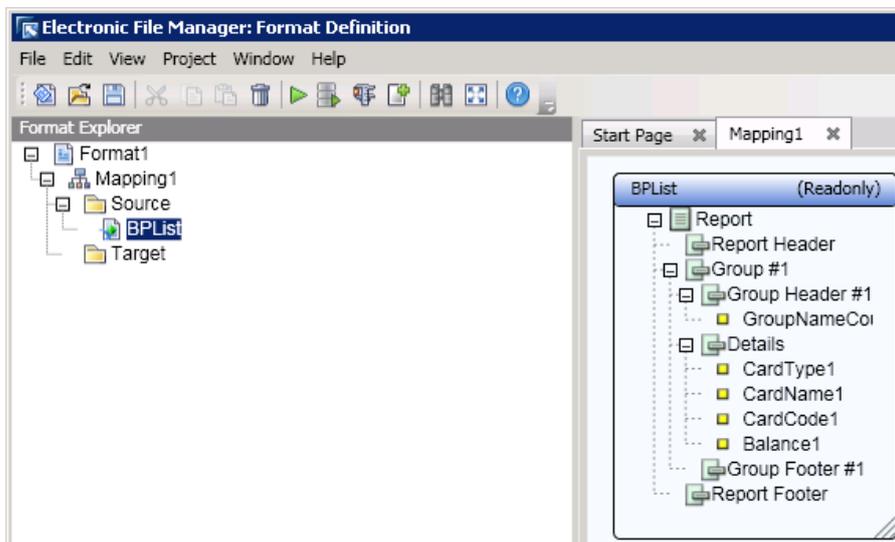
1. In the SAP Business One menu bar, choose *Tools* → *Electronic File Manager: Format Definition*.
2. Create a GEP format by doing one of the following:
  - On the *Start Page* tab, under *Create a new format project*, click *Generic Electronic File Format Project*.
  - Click  (New) or choose *File* → *New* and do the following:

- i. In the *Create a New Format* window, select  (*Generic Electronic File Format Project*).
- ii. To specify the path, choose the  (*Browse*) button and navigate to the correct location.
- iii. Specify the format name and choose the *OK* pushbutton.

A default mapping is created.

3. Add a source file as follows:
  - a. In *Format Explorer*, right-click the *Source* folder and choose *Add* → *RPT*.
  - b. In the *Open* window, select the `BPList.rpt` file at where you store it, and choose the *Open* pushbutton.

The source file is added as a data source and its format structure appears in a tree structure on the mapping tab.



4. In *Format Explorer*, select the root node of the format.
5. In the *Properties* area, in the *Deployment* section, do the following:
  - a. In the *Module* field, select *Business Partner*. After the format is imported, it appears in the Business Partner module.  
 The *Menu Path* field automatically displays the full menu path of the format in SAP Business One: \Business Partner\Business Partner Reports\Electronic Report\BP List.
  - b. In the *Menu Name* field, specify **BP List**. *BP List* appears as a menu name in the SAP Business One *Main Menu* after you import the format into SAP Business One.

Properties	
	
<b>Deployment</b>	
Version	1.0
Module	Business Partner
Menu Name	BP List
Author	Administrator
Last Changed By	Administrator
Last Changed On	11/11/2010 4:32 PM
Menu Path	\Business Partner\Business Partner Reports\Electronic R
Show Welcome Step	True
Compatible Versions	
<b>General</b>	
Name	Format 1
Description	
Type	Generic Electronic File Format Project
Localization	
File Path	C:\Documents and Settings\Administrator\Desktop\BPL

6. To add a target file for the GEP format, right-click the *Target* folder and choose *Add → XML*. Note that you can only add one target file.
7. In the displayed *Add XML Target* window, do one of the following:
  - Select *Import an XML schema*, choose the *OK* button, and do the following:
    - i. In the *Open...* window, select the *BPList.xsd* file at where you store it, and choose the *Open* pushbutton.
    - ii. In the window that appears, select *BPList* as the root node for the XML structure and choose the *OK* pushbutton.
 The imported XML schema file is added to the *Target* folder and its read-only format structure appears in a tree structure on the mapping tab.
  - Select *Import the structure from an XML document* and choose the *OK* pushbutton.  
 In the *Open...* window, select the file *BPList.xml* at where you store it, and choose the *Open* pushbutton.  
 The imported XML file is added to the *Target* folder and its format structure appears in a tree structure on the mapping tab.



#### Note

To display the target format tree in different views, do one of the following:

- Full view or compact view: Right-click the format tree head and choose the view to which you want to switch.
  - Customized compact view: do the following:
    - (i) Display the XML format tree in full view.
    - (ii) Hide or display a node by clicking the icon  (*Eye*) attached to the node. Note that this icon is only available for attribute and element nodes without mapping settings.
    - (iii) Display the XML format tree in compact view.
8. Define the mapping settings for each node. For more information, see [Defining Mapping Settings](#).
  9. Create runtime validation rules for all the target nodes except for the target node *BPList*. For more information, see [Creating Runtime Validation Rules for Target Nodes](#).
  10. To verify the format for its mapping settings and validation rules, click  (*Verify*) or choose *Project* → *Verify*.

After the verification is finished, the *Message* tab displays errors, warnings, and information messages.



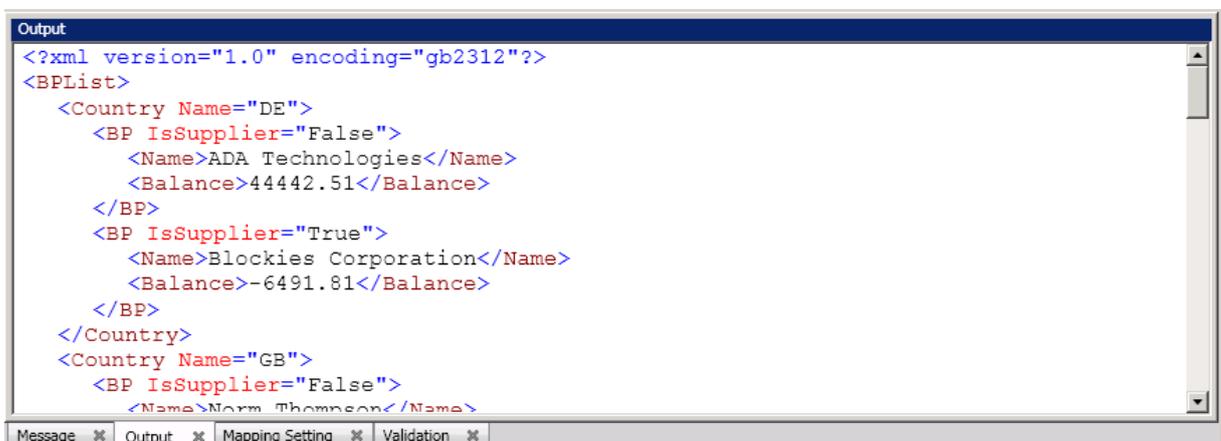
To display different types of messages, choose the *Error(s)*, *Warning(s)*, or *Information* button.

To export the messages, right-click any space on the *Message* tab and choose *Export All*. In the *Save As* window, specify the file path and name, and choose the *Save* pushbutton.

11. To start a test run, click  (*Test Run*) or choose *Project* → *Test Run*.

You test run the format with the embedded value in the source *.rpt* file.

If the test run succeeds, the *Output* tab automatically displays the simulated XML content. To save the test run results, right-click any space on the *Output* tab and choose *Export All*. In the *Save As* window, specify the file path and name, and choose the *Save* pushbutton.



If the test run fails, messages are displayed on the *Message* tab. For runtime data validation failure, the messages you specified on the *Validation* tab appear.

You can test run the format with external data. For more information, see *Test Running Formats* in the online help of the Electronic File Manager: Format Definition add-on.

12. Save the format file.

### Result

A wizard named *BP List* is created when you import the format into SAP Business One; you use this wizard to export business partner data.

## Defining Mapping Settings

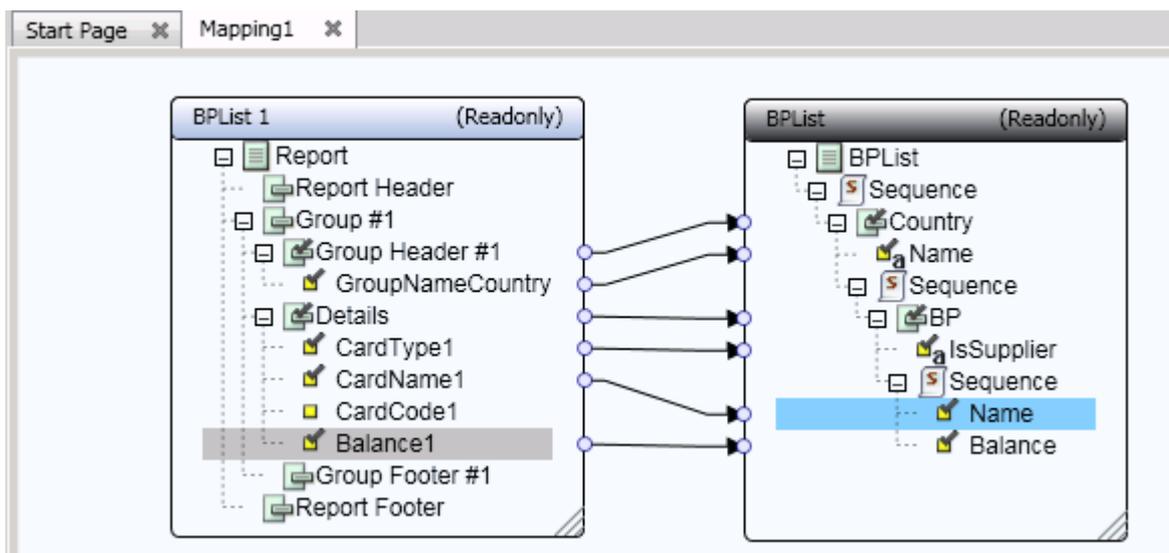
You map source nodes to target nodes by creating connections and mapping rules as required by your format solution.

### Procedure

1. Create connections between source nodes and target nodes as in the table below.

To create a connection between a source node and a target node, drag the source node from the source format tree and drop it on the target node in the target format tree. A connection line is created.

Source Node	Target Node
Group Header #1	Country
GroupNameCountry1	Name <sup>a</sup>
Details	BP
CardType1	IsSupplier <sup>a</sup>
CardName1	Name
Balance1	Balance



2. To enable mapping setting for each target node, right-click the target node and choose *Mapping Setting*.

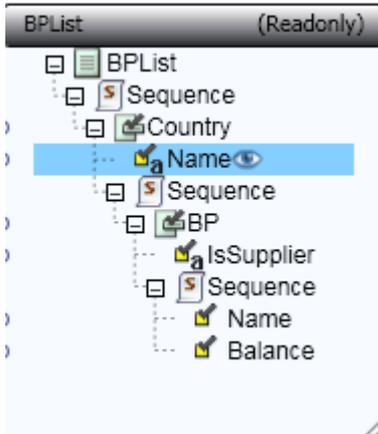
The Mapping Setting tab is enabled.

 **Note**

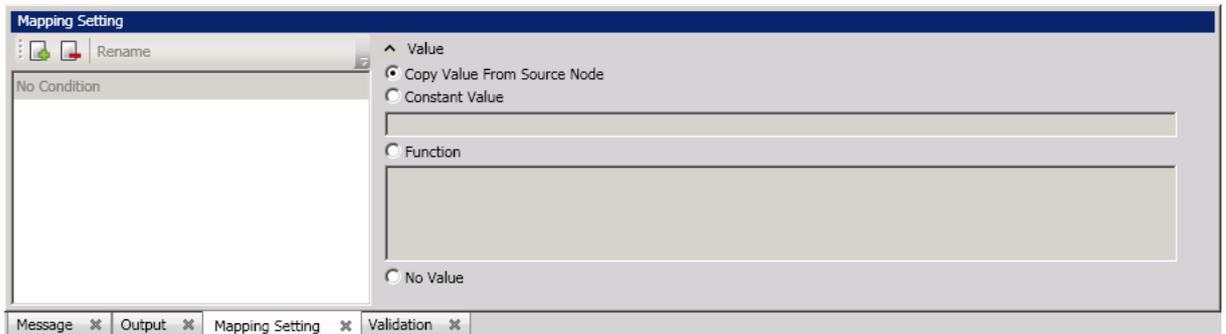
The *No Value* option is automatically selected for the target nodes that are element containers - *Country* and *BP*. The element container node cannot be filled with any value.

When the *No Value* option is selected, the padding character you specify is generated for the target node at runtime. For example, for specified character length=8, eight spaces or zeroes are generated.

3. For the target node *Name* (under the *Country* node), on the *Mapping Setting* tab, select *Copy Value From Source Node* for *Value*.



The output value for the node *Name* is its source node value.



4. For the target node *IsSupplier*, on the *Mapping Setting* tab, specify the mapping rules as follows:  
Mapping rules determine the value output at runtime.

Condition: If the condition is met at runtime, the relevant output value is used.

Else condition: If none of the conditions are met at runtime, the relevant output value is used.

- a. To add a condition, choose .
- b. Specify the following details of the expression for the condition:
  - i. For the first argument type, select *Function*.
  - ii. For the first argument, specify *substring(%CardType,1,1)*.
  - iii. For the operator, select *Equals to*.
  - iv. For the second argument type, select *Constant*.
  - v. For the second argument, specify *S*.

The condition is met when the first character of *CardType1* is "S".

**Note**

To use the reference number of a source code in a condition argument, you enter the reference number preceded by a percent sign (%), or do the following:

- i. Specify the argument type as *Reference No.*
- ii. Find the source node in the appropriate source tree and click it.
- iii. Press the **Ctrl** key and drag the source node to the blank space you have reserved in the argument, and drop.

If you have not specified a reference number for the source node, the name of the source node is automatically assigned to it as its reference number.

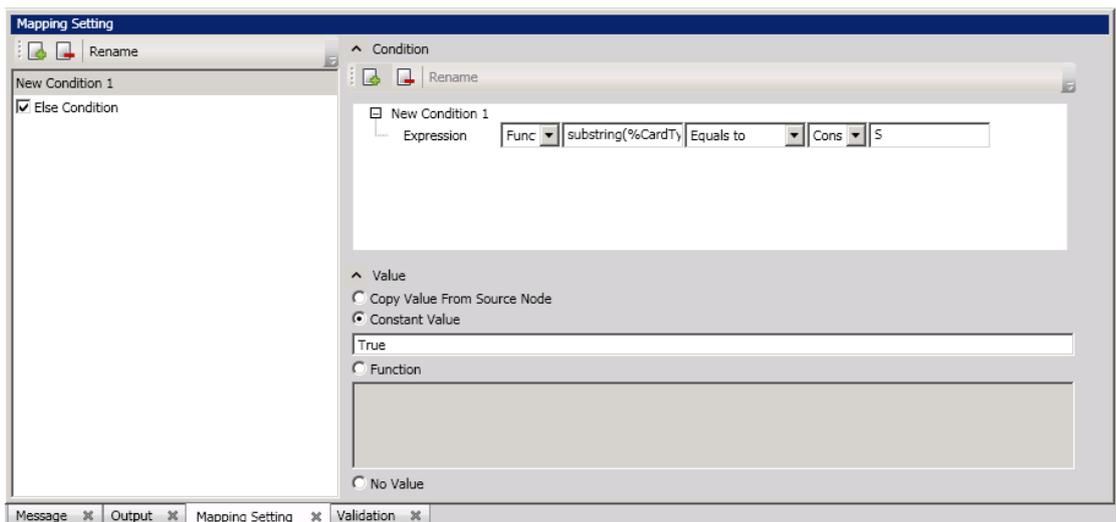
**Note**

Function: **substring(<string>, <number>, <number?>)**

Description: Returns the substring of the first argument starting at the position specified in the second argument and having the length specified in the third argument.

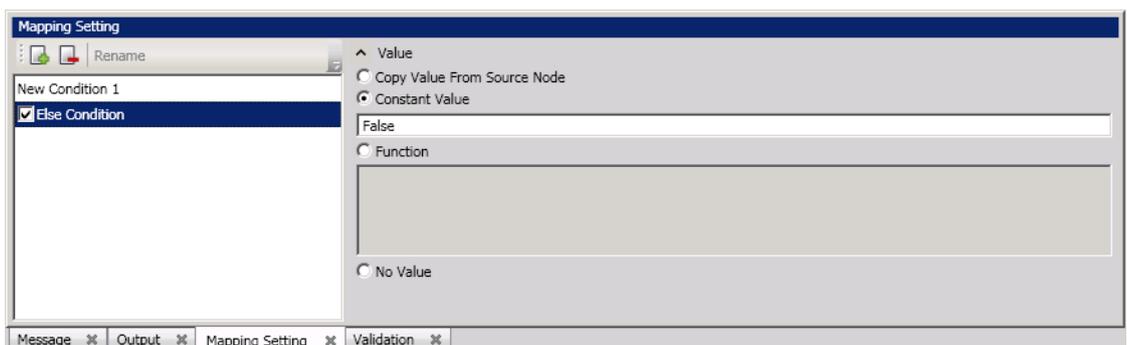
The index begins at 1.

Example: **substring("12345", 2, 3)** returns 234.

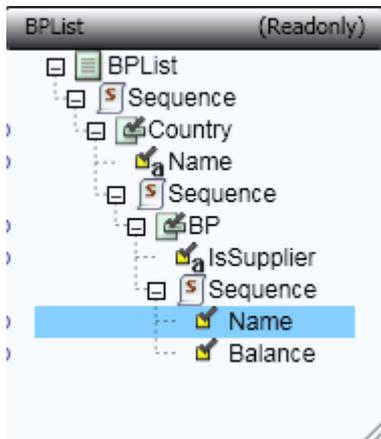


- c. Select the *Constant Value* radio button and specify **True**. If the condition is met at runtime, the output value is the constant value: **True**.
- d. Select the *Else Condition* checkbox, select the *Constant Value* radio button, and specify **False**.

If the condition is not met at runtime, the output value is the constant value: **False**.



5. For the target node *Name* (under the *Sequence* node), on the *Mapping Setting* tab, select *Copy Value From Source Node* for *Value*.



The output value for *Name* is its source node value.

6. For the target node *Balance*, on the *Mapping Setting* tab, select *Copy Value From Source Node* for *Value*.

The output value for *Balance* is its source node value.

## Creating Runtime Validation Rules for Target Nodes



### Note

If you import the XML schema file `BPList.xsd` as the target file, the validation rules are already included in the file.

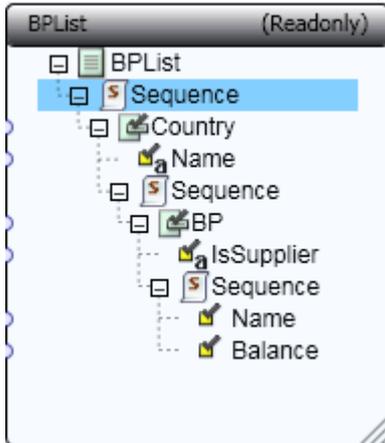
At runtime, the validation rules check the relevant values retrieved from SAP Business One. You can provide a warning or error message for any error that may occur during runtime validation:

- Error message: If any error occurs, the message appears and the validation process fails.
- Warning message: If any error occurs, the message appears but the validation process continues.

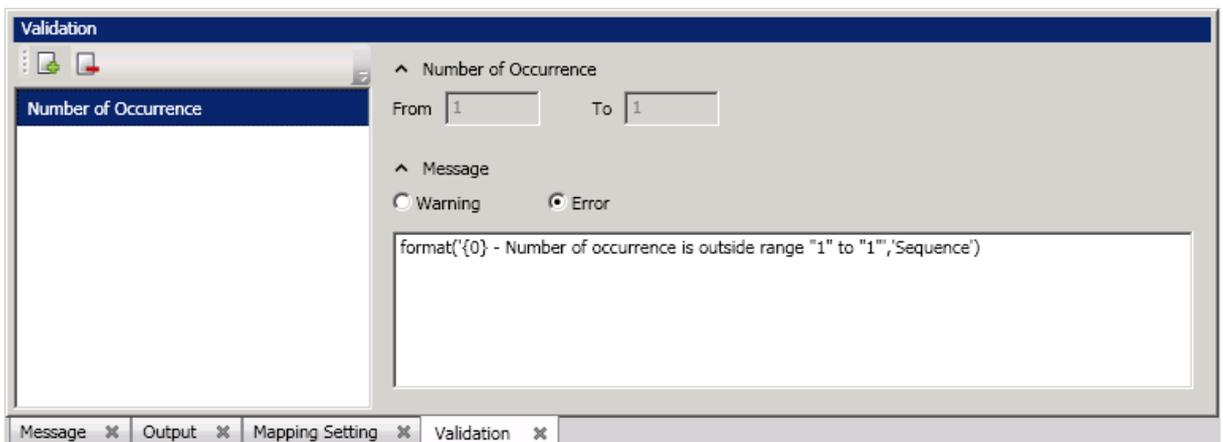
### Procedure

1. Add a validation rule for a target node as follows:
  - a. Click a node in the target tree and select the *Validation* tab.
  - b. Choose  and select one of the following validation parameters:
    - *Length*
    - *Valid Value*
    - *Data Range*
    - *Number of Occurrence*
    - *Customize*
  - c. Specify the details of the validation rule.

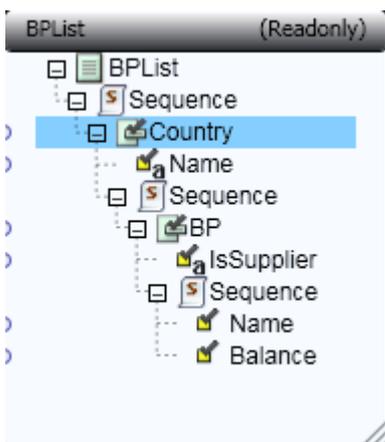
- For the target node *Sequence*, select *Number of Occurrence* and specify 1 in the *From* field and 1 in the *To* field.



The error message is automatically changed to: `format('{0} - Number of occurrence is outside range "1" to "1", 'Sequence')`.

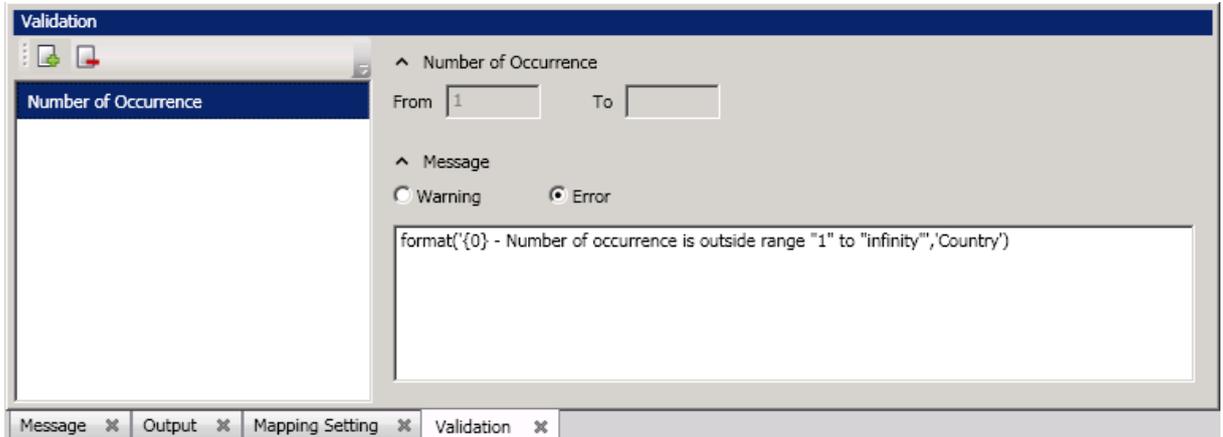


- For the target node *Country*, select *Number of Occurrence* and specify 1 in the *From* field.

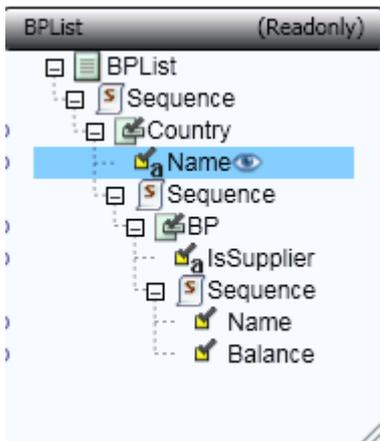


The error message is automatically changed to: `format('{0} - Number of occurrence is outside range "1" to "infinity", 'Country')`.

The country name is allowed to appear more than once at runtime. If the validation rule is broken, the following error message appears:



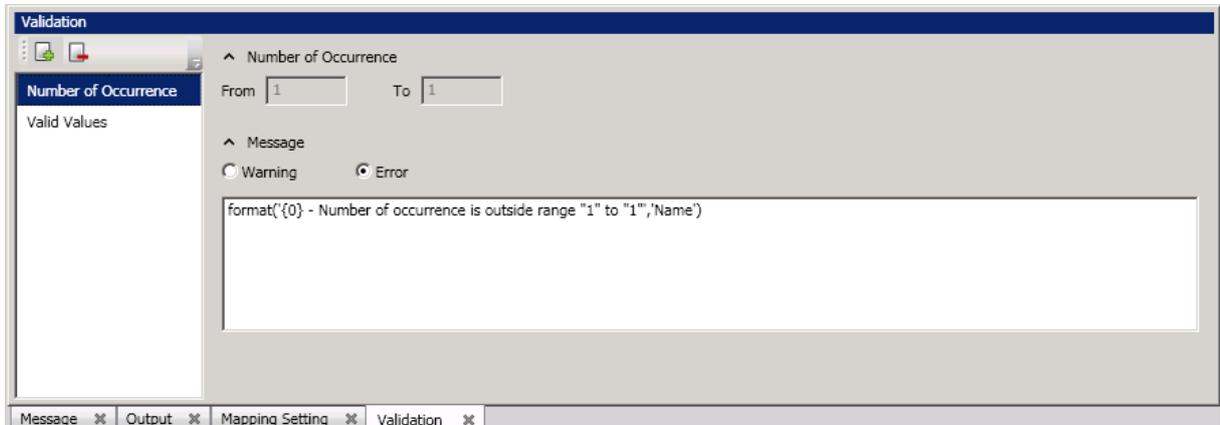
4. For the target node *Name*, configure the following:



- *Number of Occurrence*: 1 in the *From* field and 1 in the *To* field

The error message is automatically changed to: `format('{0} - Number of occurrence is outside range "1" to "1", 'Name')`.

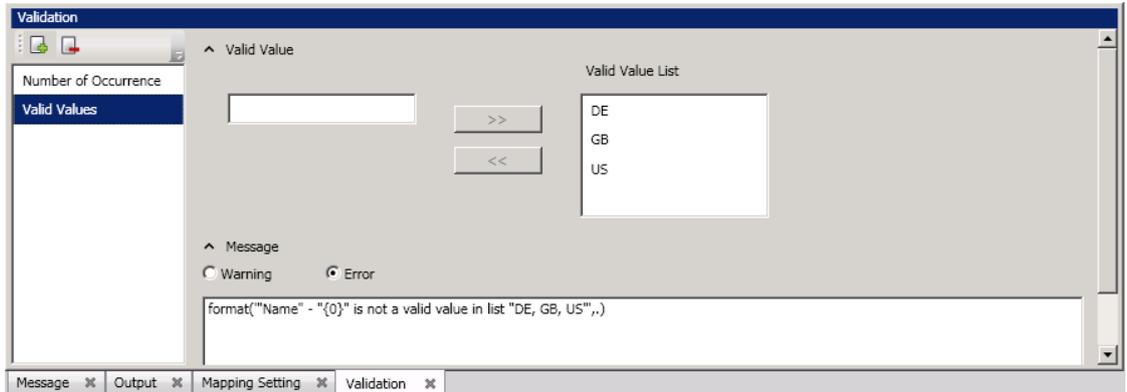
The country name is allowed to appear only once at runtime. If the validation rule is broken, the following error message appears:



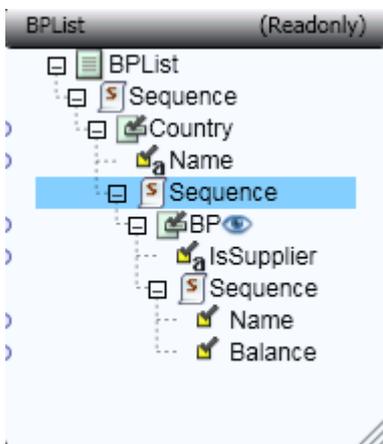
- *Valid Values*: Add **DE**, **GB**, and **US** to the valid value list.

The error message is automatically changed to: `format("Name" - "{0}" is not a valid value in list "DE, GB, US", .)`.

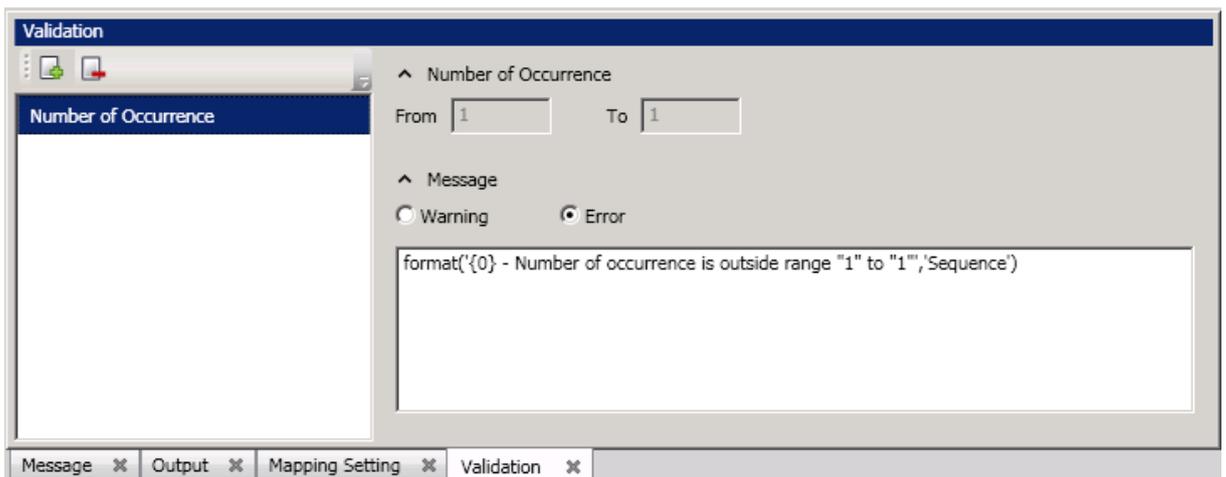
Only the values covered in the valid value list are allowed at runtime. If the validation rule is broken, the following error message appears:



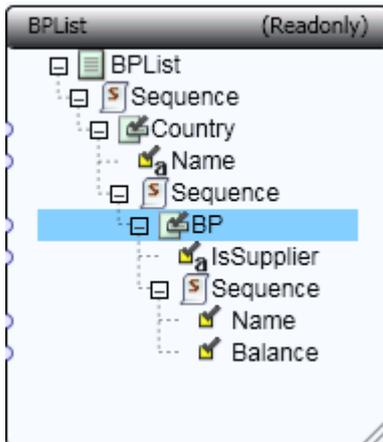
- For the target node *Sequence*, select *Number of Occurrence* and specify 1 in the *From* field and 1 in the *To* field.



The error message is automatically changed to: `format('{0} - Number of occurrence is outside range "1" to "1", 'Sequence')`.

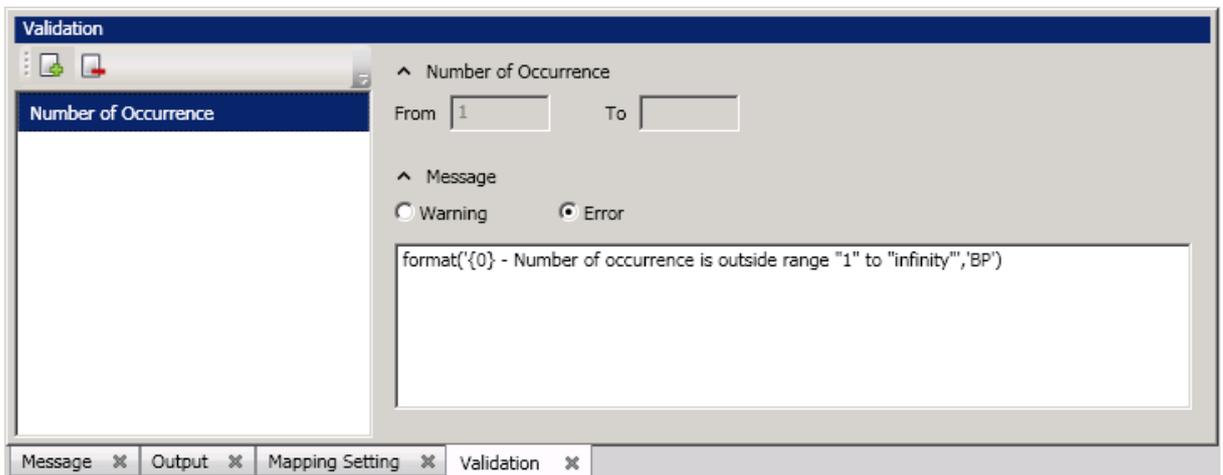


6. For the target node *BP*, select *Number of Occurrence* and specify 1 in the *From* field.

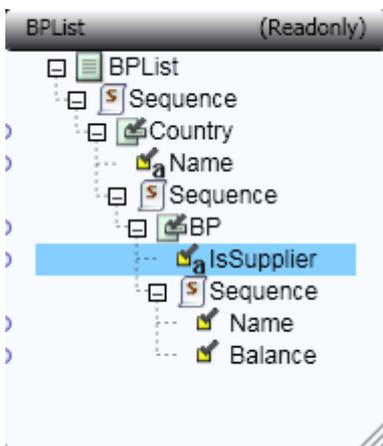


The error message is automatically changed to: `format('{0} - Number of occurrence is outside range "1" to "infinity", 'BP')`.

The BP element container is allowed to appear more than once at runtime. If the validation rule is broken, the following error message appears:



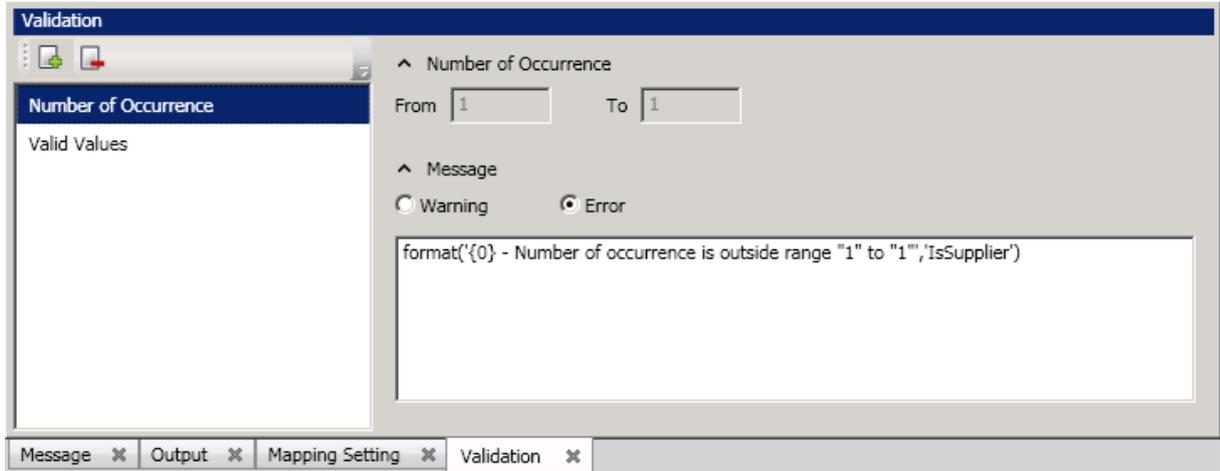
7. For the target node *IsSupplier*, configure the following:



- o *Number of Occurrence*: 1 in the *From* field and 1 in the *To* field

The error message is automatically changed to: `format('{0} - Number of occurrence is outside range "1" to "1", 'IsSupplier')`.

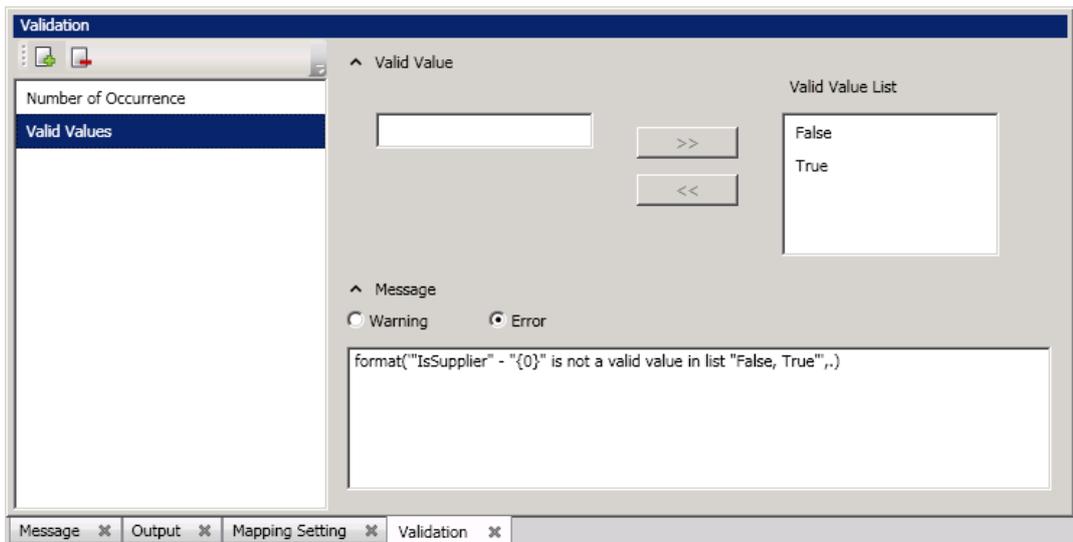
The IsSupplier attribute is allowed to appear only once at runtime. If the validation rule is broken, the following error message appears:



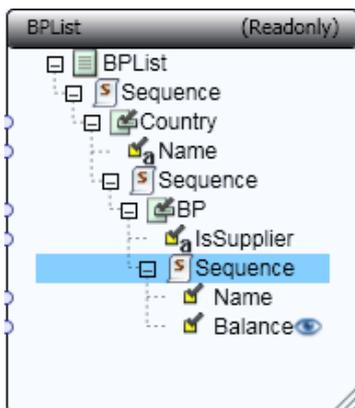
- **Valid Values:** Add **False** and **True** to the valid values list.

The error message is automatically changed to: `format( "IsSupplier" - "{0}" is not a valid value in list "False, True",.)`.

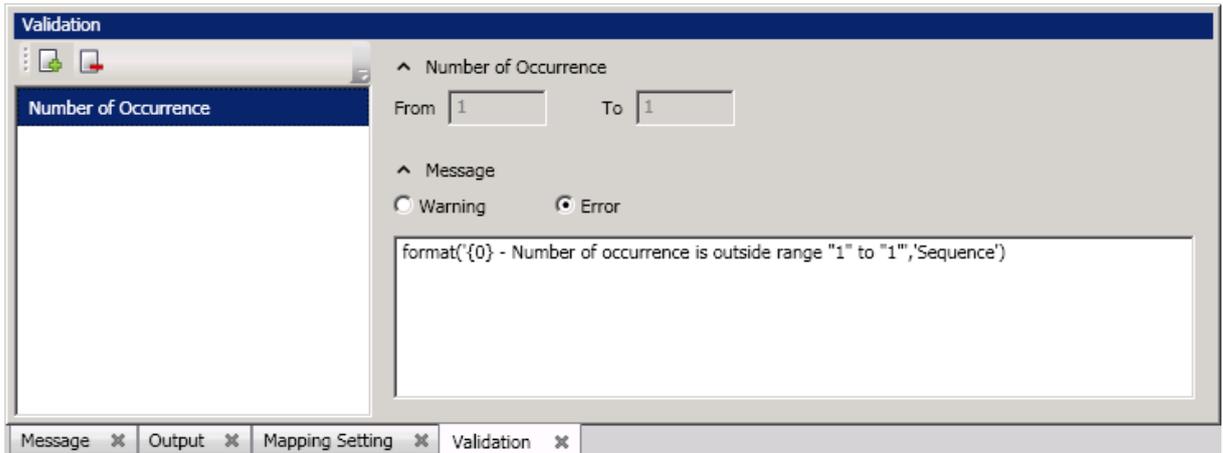
Only the values covered in the valid values list are allowed. If the validation rule is broken, the following error message appears:



- For the target node *Sequence*, select *Number of Occurrence* and specify 1 in the *From* field and 1 in the *To* field.

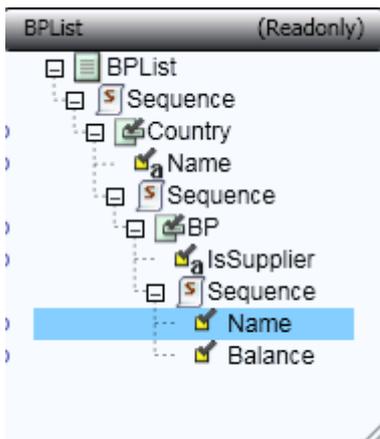


The error message is automatically changed to: `format('{0} - Number of occurrence is outside range "1" to "1", 'Sequence')`.

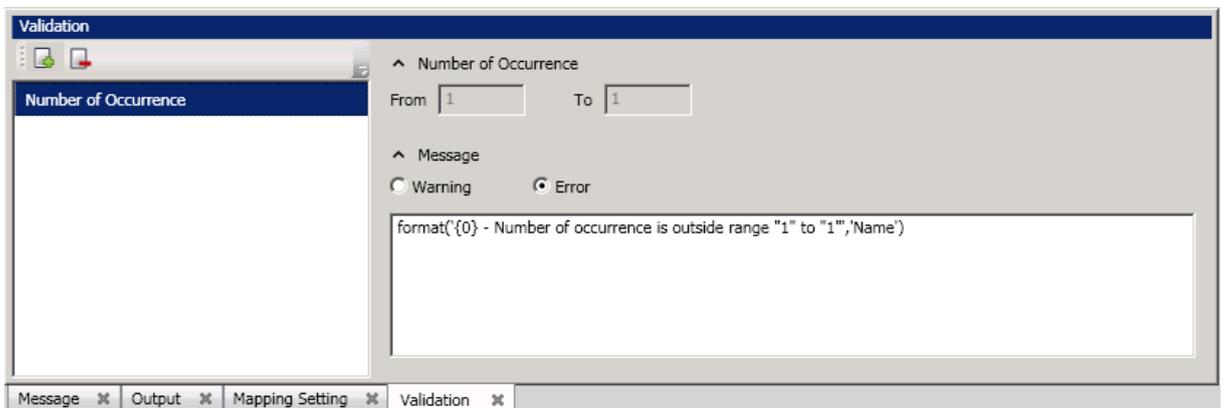


- For the target node *Name*, select *Number of Occurrence* and specify **1** in the *From* field and **1** in the *To* field.

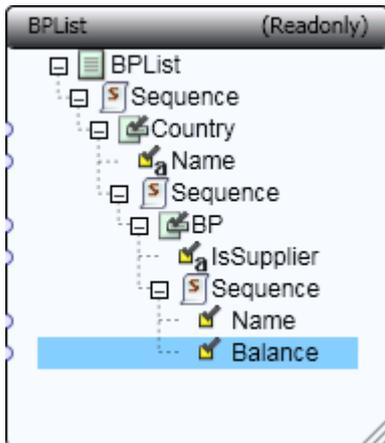
The error message is automatically changed to: `format('{0} - Number of occurrence is outside range "1" to "1", 'Name')`.



The name is allowed to appear only once at runtime. If the validation rule is broken, the following error message appears:



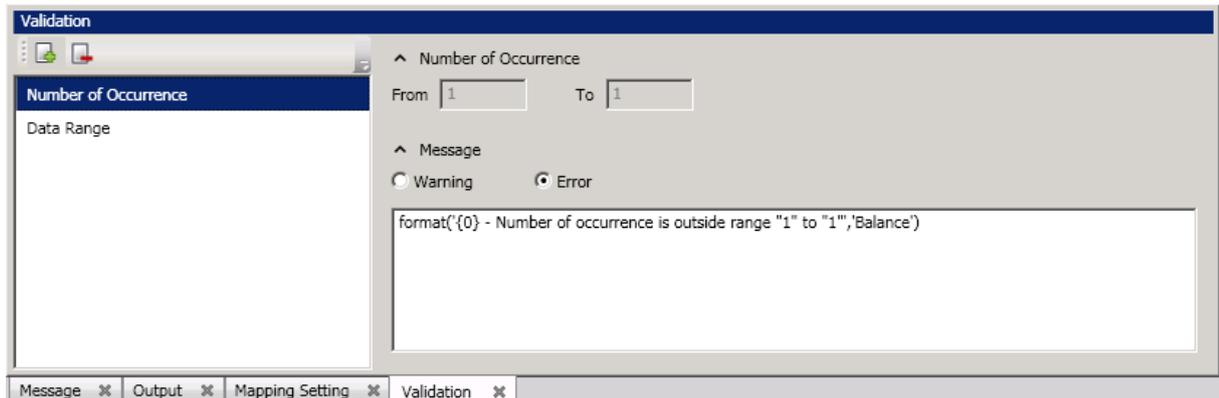
10. For the target node *Balance*, configure the following:



- *Number of Occurrence*: 1 in the *From* field and 1 in the *To* field

The error message is automatically changed to: `format('{0} - Number of occurrence is outside range "1" to "1", 'Balance')`.

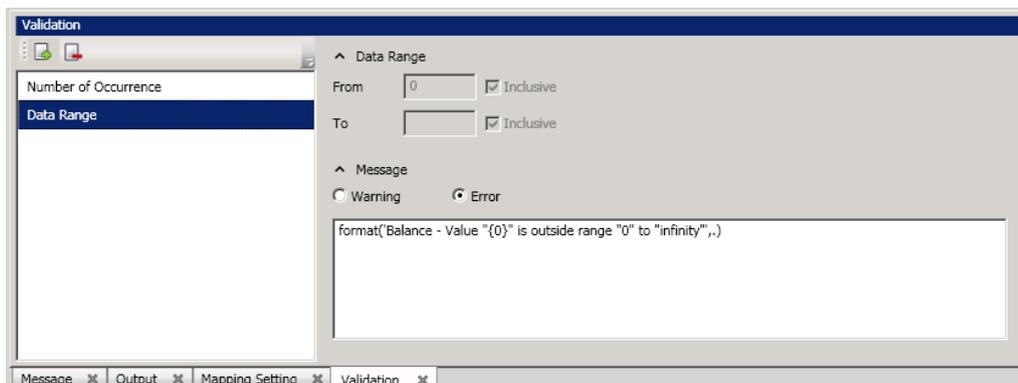
The *Balance* attribute is allowed to appear only once at runtime. If the validation rule is broken, the following error message appears:.



- *Data Range*: Specify 0 in the *From* field and select the *Inclusive* checkbox. Leave the *To* field empty and select the *Inclusive* checkbox.

The error message is automatically changed to: `format('Balance - Value "{0}" is outside range "0" to "infinity",.')`.

The allowed balance range is from zero to infinity. If the validation rule is broken, the following error message appears:





**Note**

You can create validation rules with the data range parameter only for target fields of the number or date type. These two field types are determined in the imported XML schema. If you manually create a target field, you cannot change its type to number or date in the Electronic File Manager: Format Definition add-on.

## Designing a BPP File Format

BPP file formats are designed for exporting payment data from SAP Business One into standardized bank (payment) files using the payment wizard.

The following example is based on an excerpt of the attached specification *BPP\_Spec.xls* and describes some representative steps for creating a BPP file format. To view the complete file format, open the sample file `Sample_SAPBPFROPBT_AFB_DO.BPP` in EFM.

Note that the column *Value* is not included in the original specification provided by the bank. It is the analysis result of the original specification.

### Simplified BPP specification: SAPBPFROPBT\_AFB\_DO

Section	Field	Length (Bytes)	Rules	Value	Condition
Record 06	Code enreg.	C2	"03"	"03"	/
	No sequ	C8	SAPCE	BLANK(4)	/
	Montant	C16	REGUH- RWBTR (unpacked)	PymDocAmnt	/
	Libelle/Code motif	C29	<>	<>	PymNumOfPa == 1
Trailer	Montant total 06	C16	<>	Sum(PymDocAmnt)	/



#### Note

When creating a BPP file format, you decide the position of each field in the target tree structure according to the offset defined in the specification. In turn, when you create a field, the offset is displayed automatically as read-only and you cannot change the position of the field.

However, the offset of each node is ignored in this example because it does not affect the illustration purpose.

The selected nodes of this example are related to different types of output and mapping methods:

Node Type	Node Name	Mapping Method	Output Type
Segment group	Transactions	Connection to target node	Repetitive
Standard field	Code enreg.	/	Constant value
Standard field	No sequ	/	Spaces
Standard field	Montant	Connection to target node	Numeric value
Standard field	Libelle/Code motif	Connection to target node with conditions + Function	String + constant value
Standard field	Montant total 06	Function	Sum value

## Creating a BPP File Format with TXT as the Target File

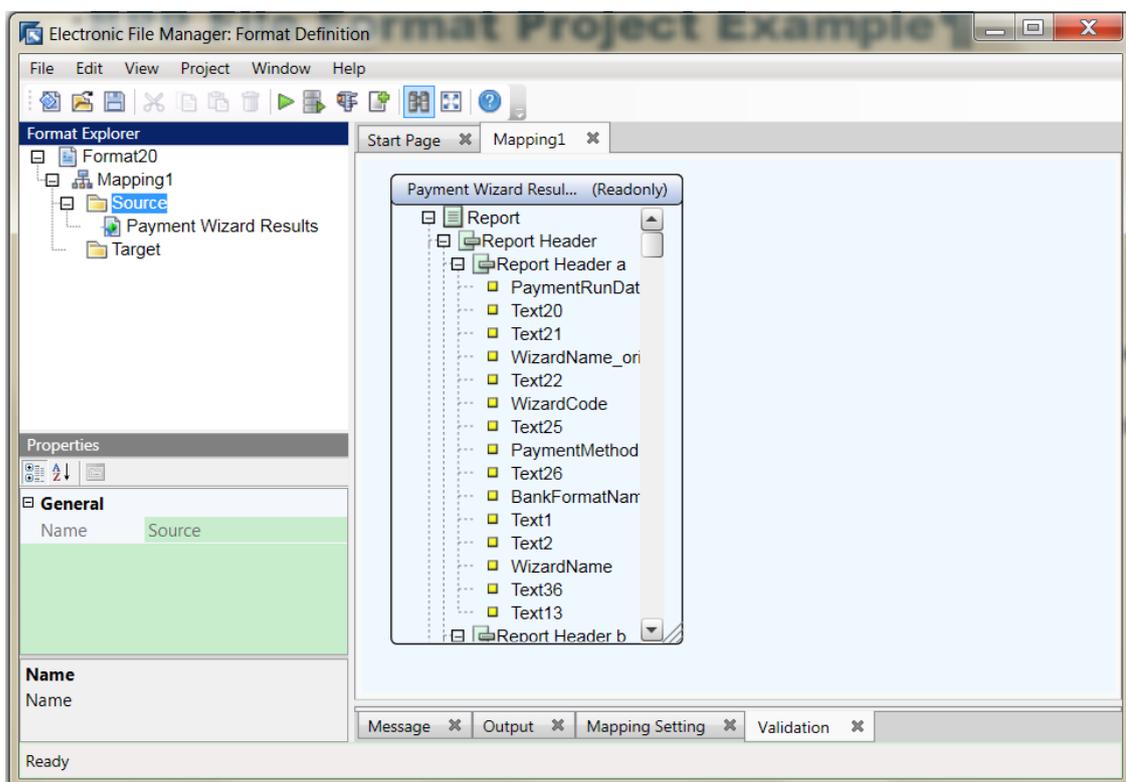


### Note

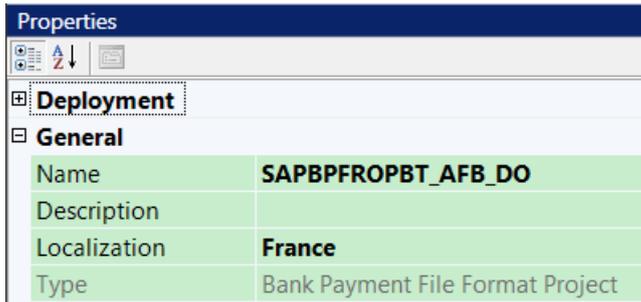
This example does not describe how to create runtime validation rules. For reference, see the GEP example.

1. In the SAP Business One menu bar, choose *Tools* → *Electronic File Manager: Format Definition*.
2. On the start page of the Electronic File Manager: Format Definition add-on, under *Create a new format project*, choose *Bank Payment File Format Project*. Alternatively, choose  (New) or *File* → *New*.

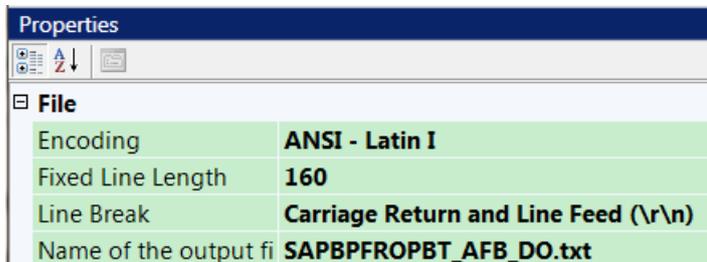
A pre-defined .rpt file *Payment Wizard Results* is automatically added as a data source and its read-only format tree structure appears on a default mapping tab of the workspace.



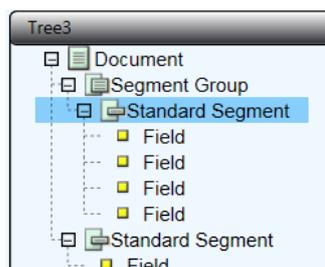
3. In *Format Explorer*, select the root node, and in the *General* section of the *Properties* area, do the following:
  - o Rename the file format as **SAPBPPFROPBT\_AFB\_DO**. The name is displayed as the default name when the file format is imported into SAP Business One.
  - o Specify the localizations at which the format is targeted by proceeding as follows:
    - i. Click the *Localization* field and choose the  (*Browse*) button that appears.
    - ii. In the *Select Localizations* window, select the *France* checkbox and choose the *OK* pushbutton.



4. Add the extra source file `OPEX.rpt` by proceeding as follows:
  - a. In *Format Explorer*, right-click the *Source* folder and choose *Add → RPT*.
  - b. In the *Open* window, select the `OPEX.rpt` file at where you store it and choose the *Open* pushbutton.
5. To add the target file, in *Format Explorer*, right-click the *Target* folder and choose *Add → TXT*. Note that you can only add one target file.
6. In *Format Explorer*, select the target file and specify its properties as follows:
  - a. In the *Encoding* field, from the dropdown list, select *ANSI – Latin I*.
  - b. In the *Fixed Line Length* field, specify `160`.
  - c. In the *Line Break* field, from the dropdown list, select *Carriage Return and Line Feed (\r\n)*.
  - d. In the *Name of the output file* field, specify `SAPBPFROPBT_AFB_DO.txt`.

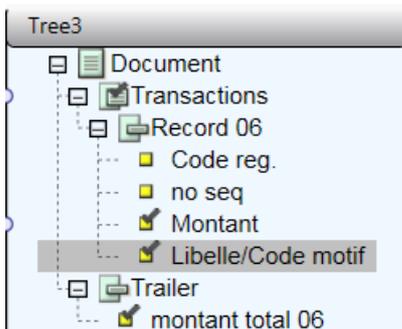


7. To define the structure of the target file, on the *Mapping 1* tab of the workspace, do the following:
  - a. Right-click the root node *Document* and add a segment group.
  - b. Right-click the root node *Document* and add a standard segment.
  - c. Right-click the newly-added segment group and add a standard segment.
  - d. Right-click the standard segment under the segment group and add four standard fields one at a time.
  - e. Right-click the standard segment under the root node *Document* and add a standard field.



8. Select each node individually and, in the *Properties* area, specify its settings as follows:

Type	Data			General
	Length	Text Alignment	Default Value	Name
Root node	/	/	/	/
Segment group	/	/	/	<b>Transactions</b>
Standard segment	/	/	/	<b>Rocord 06</b>
Standard field	2	<b>Left</b>	06	<b>Code enreg.</b>
Standard field	8	<b>Left</b>	/	<b>No sequ</b>
Standard field	16	<b>Left</b>	/	<b>Montant</b>
Standard field	29	<b>Left</b>	/	<b>Libelle/Code motif</b>
Standard segment	/	/	/	<b>Trailer</b>
Standard field	16	<b>Left</b>	/	<b>Montant total 06</b>



9. Define the mapping settings for each node. For more information, see [Defining Mapping Settings](#).
10. To verify the format, click (Verify) or choose *Project* → *Verify*.

After the verification is finished, the *Message* tab displays errors, warnings, and information messages.



To display different types of messages, choose the *Error(s)*, *Warning(s)*, or *Information* button.

By default, all types of messages are displayed on the *Message* tab.

To export messages, right-click any space on the *Message* tab and choose *Export All*. In the *Save As* window, specify the file path and name, and choose the *Save* pushbutton.

11. To start a test run, click (*Test Run*) or choose *Project* → *Test Run*.

You test run the format with the embedded value in the source `.rpt` files.

If the test run succeeds, the *Output* tab automatically displays the simulated data content. To save the test run result, right-click any space on the *Output* tab and choose *Export All*. In the *Save As* window, specify the file path and name, and choose the *Save* pushbutton.

Output					
0611.90	119.00	65.45	1188.81	17.85	VIREMENT
35.88					

If the test run fails, messages are displayed on the *Message* tab. For runtime data validation failure, the messages you specified on the *Validation* tab appear.

You can test run the format with external data. For more information, see Test Running Formats in the online help of the Electronic File Manager: Format Definition add-on.

- To save this format, click  (Save), specify the file path and name, and choose the Save pushbutton.

Alternatively, choose *File* → *Save* or *File* → *Save As*.

## Defining Mapping Settings

You map source nodes to target nodes by creating connections and mapping rules as required by your format solution.

### Recommendation

To find a node more easily, select the appropriate format tree, click  (Find), specify the searching criteria, and choose the *Find* pushbutton.

### Procedure

- Create connections between source nodes and target nodes as in the table below.

To create connection between a source node and a target node, drag the source node from the source format tree and drag and drop it on the target node in the target format tree. A connection line is created.

Target Node	Source Data File	Source Node
<i>Transactions</i> (segment)	<i>Payment Wizard Results</i>	<i>Details</i>
<i>Montant</i> (field)	<i>OPEX</i>	<i>PymDocAmnt1</i>

- To enable mapping setting for each target node, right-click the target node and choose *Mapping Setting*.

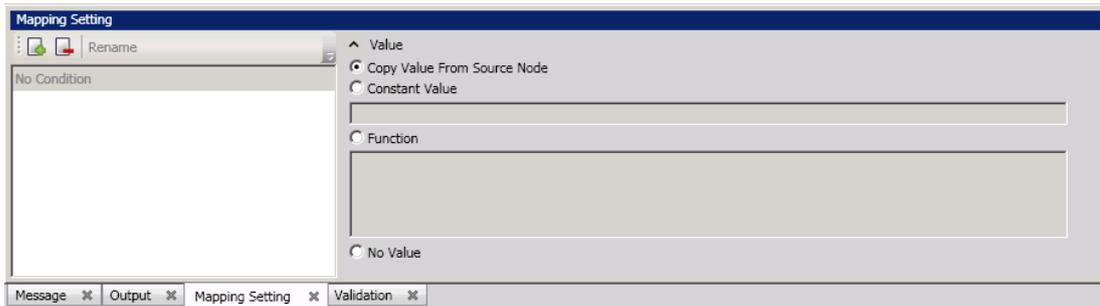
The Mapping Setting tab is enabled.



The *No Value* option is automatically selected for the target segment group node *Transactions*. The segment group node cannot be filled with any value.

When the *No Value* option is selected, the padding character you specify is generated for the target node at runtime. For example, for specified character length=8, eight spaces or zeroes are generated.

- For the target node *Montant*, on the *Mapping Setting* tab, select *Copy Value From Source Node* for *Value*.



The output value for *Montant* is its source node value.

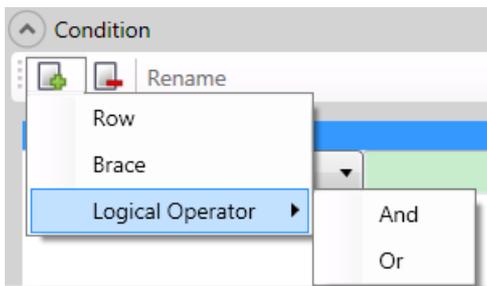
4. For the target node *Libelle*, define its output conditions and values by proceeding as follows:

- a. In the left pane of the *Mapping Settings* tab, click

A new condition is added.



- b. In the right pane of the *Mapping Settings* tab, select *New Condition 1*, click , and add the logical operator *And*.

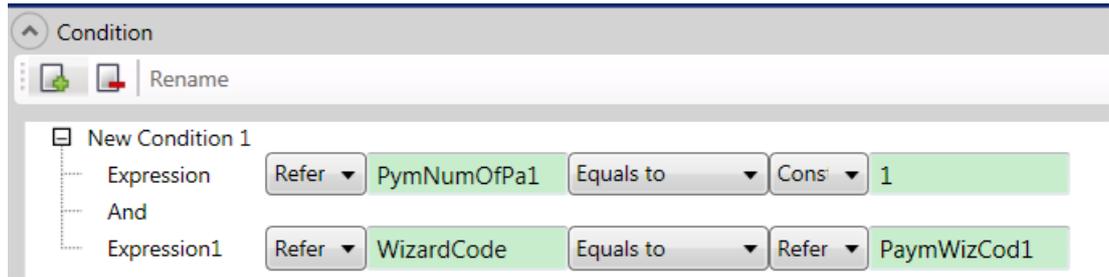


- c. Click and choose *Row*.

A new expression is added.

- d. Specify the two expressions of the condition as follows:

Name	1 <sup>st</sup> Argument Type	1 <sup>st</sup> Argument	Operator	2 <sup>nd</sup> Argument Type	2 <sup>nd</sup> Argument
<i>Expression</i>	<b>Reference No.</b>	<b>PymNumOfPa1</b>	<b>Equals to</b>	<b>Constant Value</b>	<b>1</b>
<i>Expression 1</i>	<b>Reference No.</b>	<b>WizardCode</b>	<b>Equals to</b>	<b>Reference No.</b>	<b>PaymWizCod1</b>



The second expression ensures that the data retrieved from the *OPEX* and the *Payment Wizard Results* source tables are for the same payment wizard run.



### Note

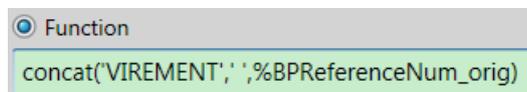
To use the reference number of a source code in a condition argument, enter the reference number preceded by a percent sign (%), or do the following:

- i. Specify the argument type as *Reference No.*
- ii. Find the source node in the appropriate source tree and click it.
- iii. Press the **Ctrl** key and drag the source node to the blank space you have reserved in the argument, and drop.

If you have not specified a reference number for the source node, the name of the source node is automatically assigned to it as its reference number.

- e. Select *Function* for *Value* and specify the following expression:

```
concat('VIREMENT', ' ', %BReferenceNum_orig)
```



This function concatenates a string and a number with a space in between.

**BReferenceNum\_orig** is the reference number of the source node *BReferenceNum\_orig* in the source tree *Payment Wizard Results*.



### Note

To use the reference number of a source code in a function, enter the reference number preceded by a percent sign (%), or do the following:

- i. Find the source node in the appropriate source tree and click it.
- ii. Press the **Ctrl** key and drag the source node to the blank space you have reserved in the function expression, and drop.

If you have not specified a reference number for the source node, the name of the source node is automatically assigned to it as its reference number.

- f. On the left pane of the *Mapping Setting* tab, select the *Else Condition* checkbox.
  - g. On the right pane of the *Mapping Setting* tab, select *Function* for *Value* and enter `string('VIREMENT')`.
5. For the target node *Montant total*, on the *Mapping Setting* tab, select *Function* for *Value* and enter `sum(%PaymentAmountLC)`.

**PaymentAmountLC** is the reference number of the source node *PaymentAmountLC* in the source tree *Payment Wizard Results*.

## Designing a BFP File Format

BFP file formats are designed for parsing and importing bank statement data into SAP Business One.



### Note

BFP is the same file format you edit in the Format Definition add-on of SAP Business One 2007 and 8.8.

The following example is based on an excerpt of the attached specification `Spec_csb43.doc` and describes some representative steps for creating a BFP file format. To view the complete file format, open the sample file `Sample_csb43.BFP` in EFM.

Note that the column *B1 Field* in the specification is not included in the original specification provided by the bank. It is an analysis result of the original specification.

### Simplified BFP Specification: csb43

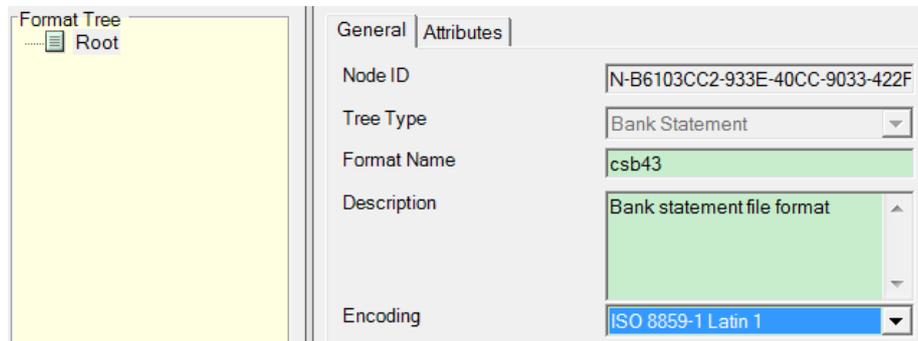
Description	B1 Field	Offset	Length (Bytes)	Content	More Info.
Record code	/	0	C2	"11"	/
Bank code	/	2	C4	/	Code of the bank that sends the file. If smaller than 4 characters, zeros on the left
Bank account number	AccountNumber	10	C10	/	Bank account number. If smaller than 10 characters, zeros on the left
Beginning date	/	20	C6	/	YYMMDD
Key 1 or 2	/	32	C1	/	1. Debit 2. Credit
Beginning balance amount	StartingBalanceF	33	C14	/	Last two characters reflect the decimal amount. Zeros on the left
Currency	Currency	47	C3	/	ISO code of the currency Euro
Free field	/	77	C3	/	Blanks

## Creating a BFP File Format

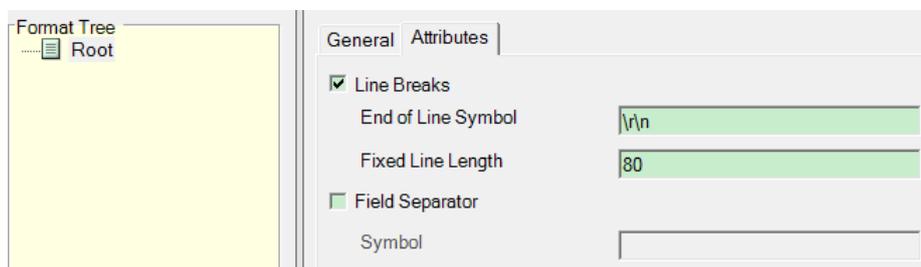
1. In the SAP Business One menu bar, choose *Tools* → *Electronic File Manager: Format Definition*.
2. On the start page of the Electronic File Manager: Format Definition add-on, under *Create a new format project*, choose *Bank Statement File Format Project*.

Alternatively, choose  (New) or *File* → *New*.

3. Select the root node and do the following:
  - Specify the general settings of the format as below:



- Specify the format attributes as below:



EFM supports four symbols: carriage return (`\r`), line feed (`\n`), backslash (`\`), and tab (`\t`). To define a symbol, you either type in the symbol, or choose *Insert* → *Symbols* and specify the symbol.

The format attributes are automatically copied to each standard segment or field.

4. Right-click the root node and insert a segment group. Alternatively, select the root node and choose *Insert* → *Segment Group*.
5. Select the segment group and do the following:
  - On the *General* tab, rename the group as **Statements**.
  - Specify the group attributes as shown below:



6. Right-click the segment group *Statements* and insert a standard segment. Alternatively, select the segment group and choose *Insert* → *Standard Segment*.
7. Select the standard segment and do the following:
  - On the *General* tab, rename the segment as **Header** and specify the segment as mandatory.
  - On the *Attributes* tab, select to locate the segment by keyword and specify the starting keyword as 11.

**Note**

After creating a standard segment, you cannot proceed to create any new node before you specify by which attribute to locate the segment.

8. Right-click the segment *Header* and insert the following fields:

Field Type	Field Name	General Settings		Attributes
		Mandatory	Reference No.	
Standard	<b>BankCode</b>	Yes	/	Locate by: Position Position: 3 Length: 4 Data Type: String Format String: right (Leading 0)
Standard	<b>AccountNumber</b>	Yes	/	Locate by: Position Position: 11 Length: 10 Data Type: String Format String: right (Leading 0)
Standard	<b>BeginDate</b>	Yes	/	Locate by: Position Position: 21 Length: 6 Data Type: Date Format String: YYMMDD
Standard	<b>BalanceSign</b>	Yes	Ref1	Locate by: Position Position: 33 Length: 1
Conditional	<b>BeginBalance</b>	Yes	/	Locate by: Position Position: 34 Length: 14 Data Type: Numeric Format String: 00123456 (.) 12
Standard	<b>Currency</b>	Yes	/	Locate by: Position Position: 78 Length: 3

Field Type	Field Name	General Settings		Attributes
		Mandatory	Reference No.	
Standard	Free	Yes	/	Locate by: Position Position: 34 Length: 14 Data Type: Numeric Format String: 00123456 (.) 12

9. Right-click the conditional field *BeginBalance*, insert two result fields, and name them as *DebitAmount* and *CreditAmount*.
10. On the *Conditions* tab, define the conditions for the mapping settings to take effect and generate the required value. For more information, see [Defining Conditions for Value Output](#).
11. On the *Mapping* tab, define the mapping settings for each node. For more information, see [Defining Mapping Settings](#).
12. To test run the file format, do the following:
  - a. Choose *File* → *Data Preview*.
  - b. Select the sample bank statement file `csb43.txt` or any other file that matches the file format.
  - c. View the details of the statements and statement rows.



**Note**

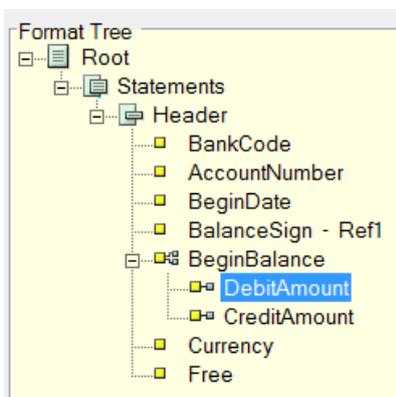
Before performing a test run of the file format, the system automatically runs a verification on the conditions and mapping settings. If the verification fails, you need to correct the errors you made before test running the file format.

For more information, see the online help of the Format Definition add-on.

13. To save this format, choose *File* → *Save* or *File* → *Save As*, specify the file path and name, and choose the *Save* pushbutton.

**Result**

A BFP file format is created with the following format tree:



## Defining Conditions for Value Output

1. Select the result field *DebitAmount* and define the conditions as the following:

- *Type1: Reference No.*
- *Arg1-2: Ref1*
- *Operator: Equal to*
- *Type2: Constant*
- *Arg2-1: 1*
- *And/Or: And*

Type1	Arg1-1	Arg1-2	Operator	Type2	Arg2-1	Arg2-2	And/Or
Reference No.		Ref1	Equal to	Constant		1	And

A reference number is the exclusive identifier of a node. You must define the reference number before you use it to define a condition or mapping method.



### Note

Argument 1-1 or 2-1 is only active when its type is *Database Field*.

2. Select the result field *CreditAmount* and define the conditions as the following:

- *Type1: Reference No.*
- *Arg1-2: Ref1*
- *Operator: Equal to*
- *Type2: Constant*
- *Arg2-1: 2*
- *And/Or: And*

Type1	Arg1-1	Arg1-2	Operator	Type2	Arg2-1	Arg2-2	And/Or
Reference No.		Ref1	Equal to	Constant		2	And

## Result

At runtime, the result fields *DebitAmount* and *CreditAmount* receive values from their parent conditional field *BeginBalance* according to the defined conditions.

## Defining Mapping Settings

You map an SAP Business One database table or field to a target node if needed. The settings of all mapping are defined as in the table below:

Target Node	Mapping Settings			
	B1 Table	B1 Field	Mapping Method	Constant
Statements	<b>BankStatement</b>	/	/	/
AccountNumber	<b>BankStatementRow</b>	<b>AccountNumber</b>	/	/
DebitAmount	<b>BankStatement</b>	<b>StartingBalanceF</b>	<b>concat('-',.)</b>	/
CreditAmount	<b>BankStatement</b>	<b>StartingBalanceF(1)</b>	/	/

Target Node	Mapping Settings			
	B1 Table	B1 Field	Mapping Method	Constant
Currency	BankStatement	Currency	/	EUR

**Note**

- The result fields *DebitAmount* and *CreditAmount*, and accordingly, their parent conditional field *BeginBalance*, are mapped to the same target field *StartingBalanceF*.
- You concatenate the value of the field *DebitAmount* with a minus sign (-) because debited amounts in the bank statements are recorded like this.
- For the value of the field *Currency*, you use a currency code as defined in SAP Business One instead of the source value. You decide which currency code to use according to the actual situation of your company.

## Target SAP Business One Database Tables and Fields for Mapping

EFM supports three SAP Business One database tables and some of their fields for mapping. They are listed in the table below:

Target Table	Target Field
BankStatement	StatementNumber
	StatementDate
	StartingBalanceF
	EndingBalanceF
	Currency
	StartingBalanceL
	EndingBalanceL
BankStatementRow	AccountNumber
	AccountName
	Reference
	DueDate
	Details
	DebitAmountFC
	CreditAmountFC
	CreditCurrency
	Balance
	ExternalCode
	BPName
	Details
	BankStmtLineDate
	BankStmtDueDate
	BPBankCode
	BPBankAccount
	BPBICSwiftCode
	DebitAmountLC
	CreditAmountLC
	ExchangeRate
IBANofBPBankAccount	
FeeOnTheLine	
MultiplePayment	DocumentIdentifier

Target Table	Target Field
	AmountFC
	AmountLC
	IsDebit   <b>Note</b> Use the constant <code>⌘YES</code> or <code>⌘NO</code> .

For more information about these tables and fields, see the SDK online help.

## Other Reference Material

The documents listed in the table below are referred to in this document.

Document	Location
SDK online help file <i>SDK_EN.CHM</i>	<a href="http://service.sap.com/smb/sbocustomer/documentation">http://service.sap.com/smb/sbocustomer/documentation</a> and choose <i>Release Family 8.8</i> → <i>SDK and Custom Development</i> → <i>SAP Business One 8.81/SAP Business One 8.82</i> → <i>SDK 8.82 Help Center</i>  Available also on the SAP Business One product DVD and in the download package from SAP Service Marketplace
Electronic File Manager: Format Definition online help	From within the Electronic File Manger: Format Definition add-on
Format Definition online help	From within the Electronic File Manager: Format Definition add-on when you are creating or editing a BFP file format
<i>How to Work with SAP Crystal Reports in SAP Business One</i>	At <a href="http://service.sap.com/smb/sbocustomer/documentation">http://service.sap.com/smb/sbocustomer/documentation</a> , choose <i>Release Family 8.8</i> → <i>Modules and Features</i> → <i>How-To Guides 8.8 Release Family</i> → <i>Title</i> .

For more information about the SPP file format, see SAP Note [1531223](#).

## Copyrights, Trademarks, and Disclaimers

© Copyright 2012 SAP AG. All rights reserved.

The current version of the copyrights, trademarks, and disclaimers at <http://service.sap.com/smb/sbocustomer/documentation> is valid for this document.