

How to Guide

SAP Business One 8.82, 9.0

Document Version: 1.0 – 2013-11-27

PUBLIC

How to Create a Text-Based Bank File Format with the Electronic File Manager

All Countries



Typographic Conventions

Type Style	Description
<i>Example</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Textual cross-references to other documents.
Example	Emphasized words or expressions.
EXAMPLE	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
Example	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
Example	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE	Keys on the keyboard, for example, F2 or ENTER.

Document History

Version	Date	Change
1.0	2013-11-27	First version

Table of Contents

Introduction.....	5
Prerequisite.....	5
Example of Designing a BPP File Format	6
Step 1: Starting the Electronic File Manager	6
Step 2a: Creating a Target File	9
Step 2b: Defining the File Structure.....	12
Step 3: Mapping Settings.....	19
Mapping Settings for DTAUS_EN.BPP	25
Step 4: Assigning File Formats.....	31
Step 5: Generating Payment Files.....	33
Appendix 1: DTAUS File Format Description.....	34
Record Levels.....	35
Record Level A (Data Header).....	35
Record Level C (Single Payment Order)	37
Record E (Data Trailer)	43
Checking Records C	45
Appendix 2: Payment Wizard Results Source.....	46
Appendix 3: Functions.....	51

Introduction

The Payment Engine add-on of SAP Business One allows you to export your payment data to a payment file once you have defined the appropriate bank file format.

This guide explains how to use the Electronic File Manager to design BPP (Bank Payment File Format Project) file formats. The example specification used in this guide is for the German DTAUS file format (version: DTAUS0, 2010); the particular SAP Business One version is Release 9.0, but the instructions apply to Release 8.82 as well.

Prerequisite

Before starting to create a bank payment file format with the Electronic File Manager (EFM) you should familiarize yourself with the structure of the bank file. The file format description is freely available on the Internet.

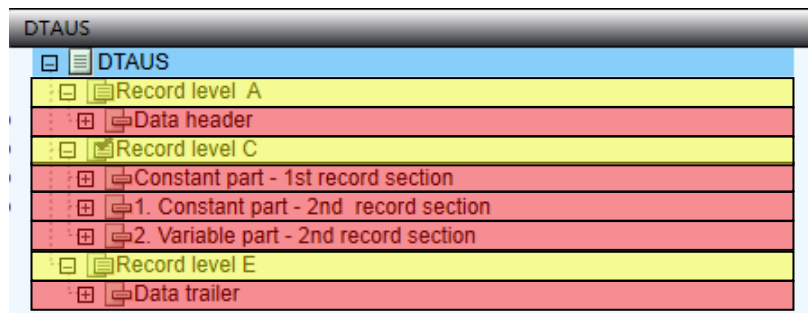
Alternatively, contact the local bank and ask for the file format description of the file format you want to create.

For a detailed DTAUS format description, see [Appendix 1: DTAUS File Format Description](#).

Example of Designing a BPP File Format

Once you have become familiar with the structure of the payment file, you can start to create a payment file format with EFM.

The following step-by-step instructions use the German DTAUS file format as the example. Below is the preview of the DTAUS file format structure that we use as an example in this guide:



Legend:

Segment group:



Standard segment:



Step 1: Starting the Electronic File Manager

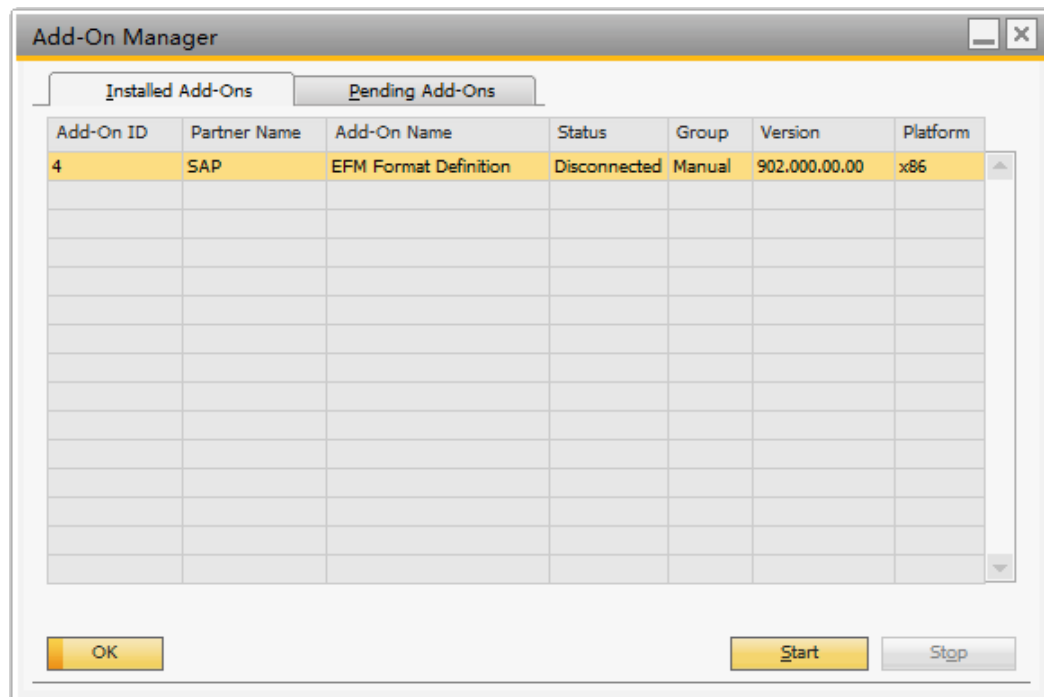
Prerequisite

You have registered and installed the Electronic File Manager: Format Definition add-on.

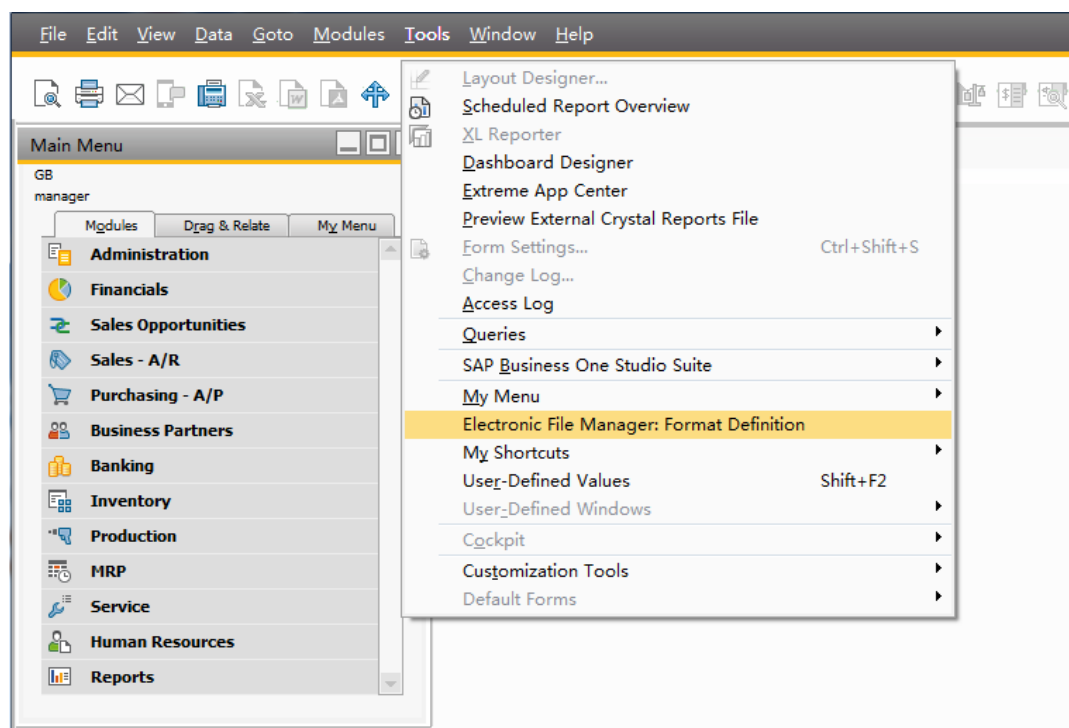
Procedure

1. From the SAP Business One [Main Menu](#), choose [Administration](#) → [Add-Ons](#) → [Add-On Manager](#).

2. In the *Add-On Manager* window, on the *Installed Add-Ons* tab, select the EFM add-on and choose the *Start* button.

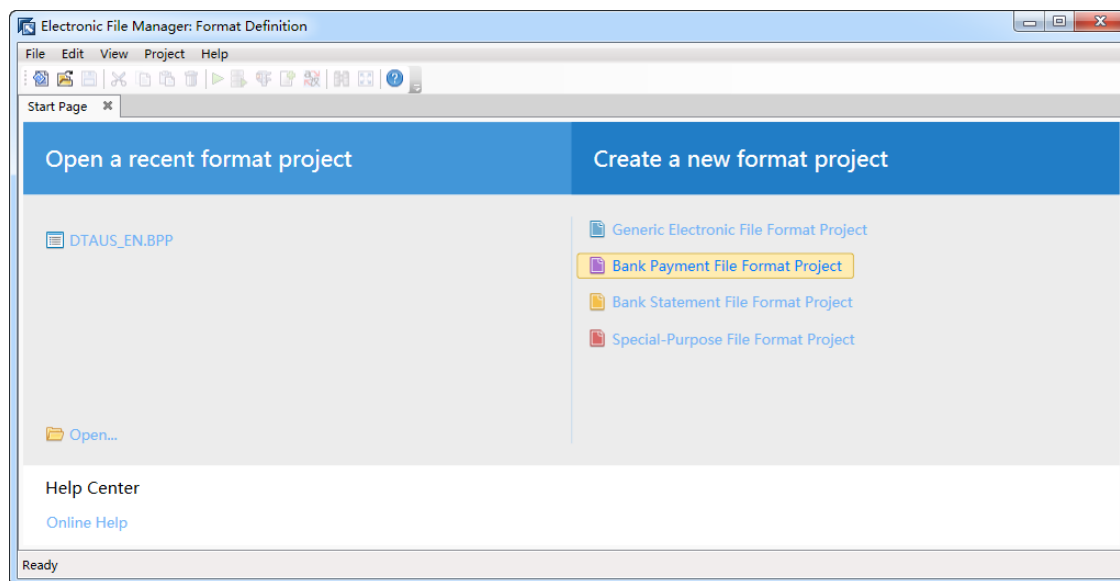


- From the **Tools** menu, choose *Electronic File Manager: Format Definition*.



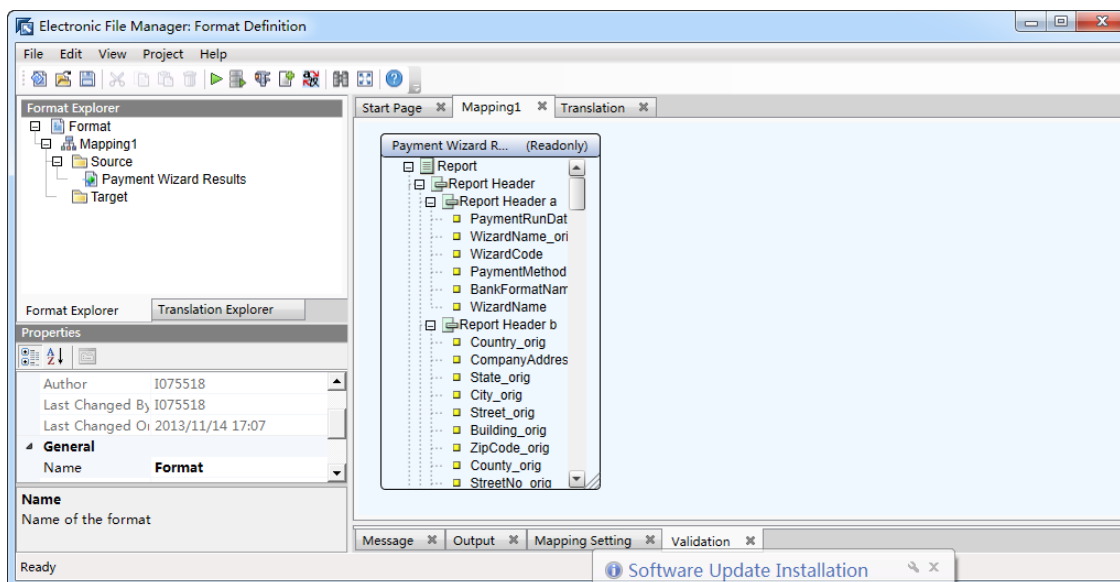
The Electronic File Manager opens.

4. In the *Create a new format project* pane, choose *Bank Payment File Format Project*.



Result

The source *Payment Wizard Results* is loaded automatically.



The source file is a Crystal report (.rpt). For more information, see [Appendix 2: Payment Wizard Results Source](#).

Step 2a: Creating a Target File

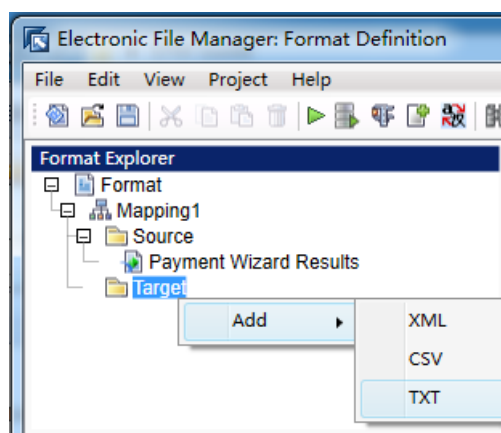
Generally, the embedded source *Payment Wizard Results* is enough for you to create a file format. Therefore, our example skips the step of adding additional source files. You can directly add a target file.

Note

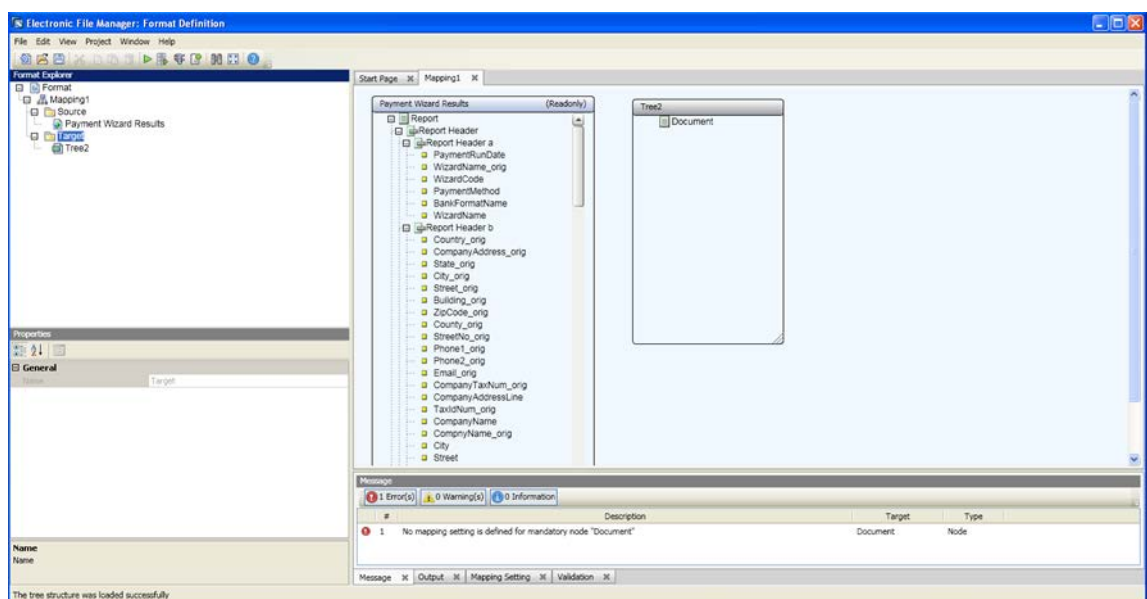
You can add more than one source file but only one target file is allowed.

Procedure

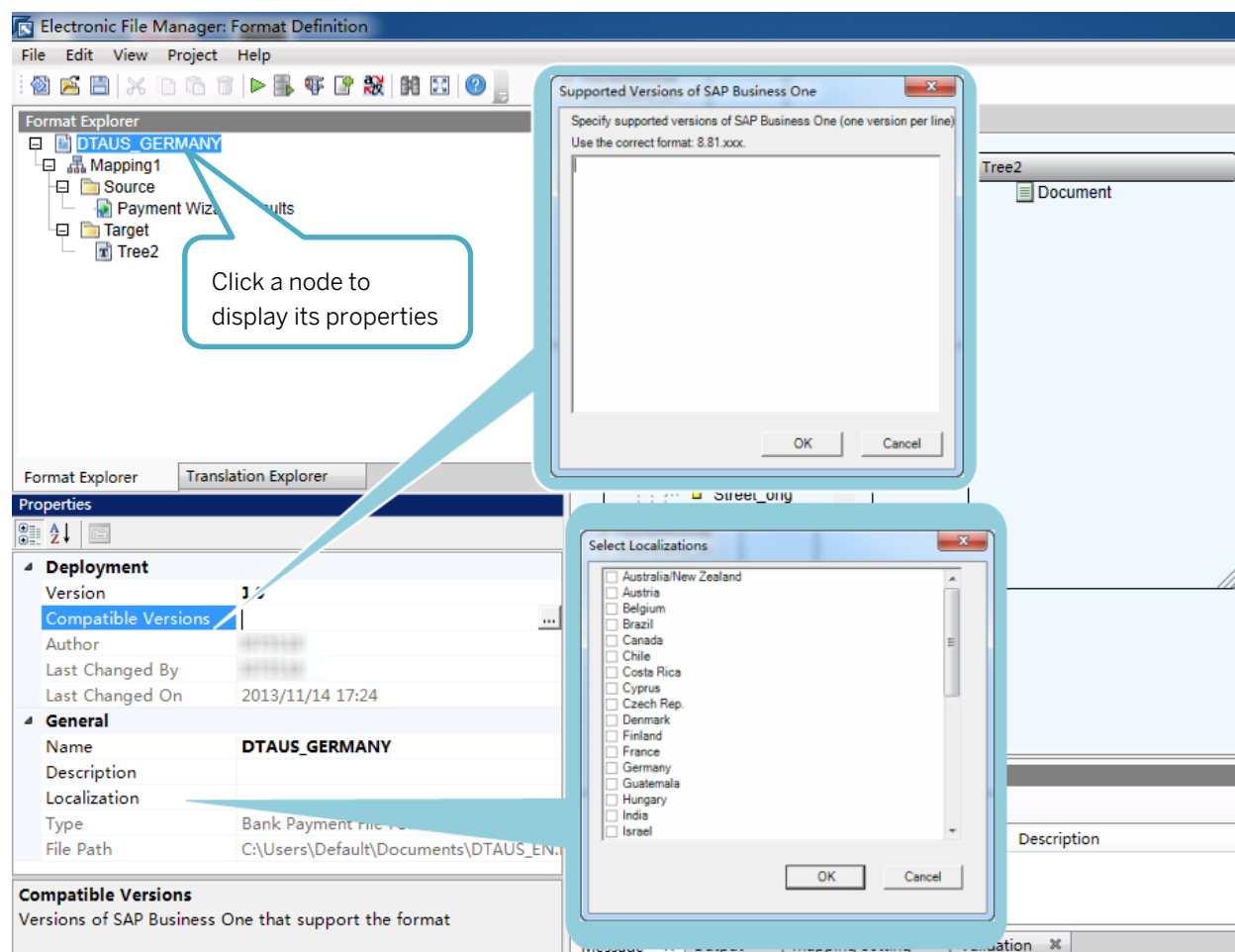
1. Right-click the *Target* node and choose *Add* -> *TXT*, as this is the format of the payment file we want to create.



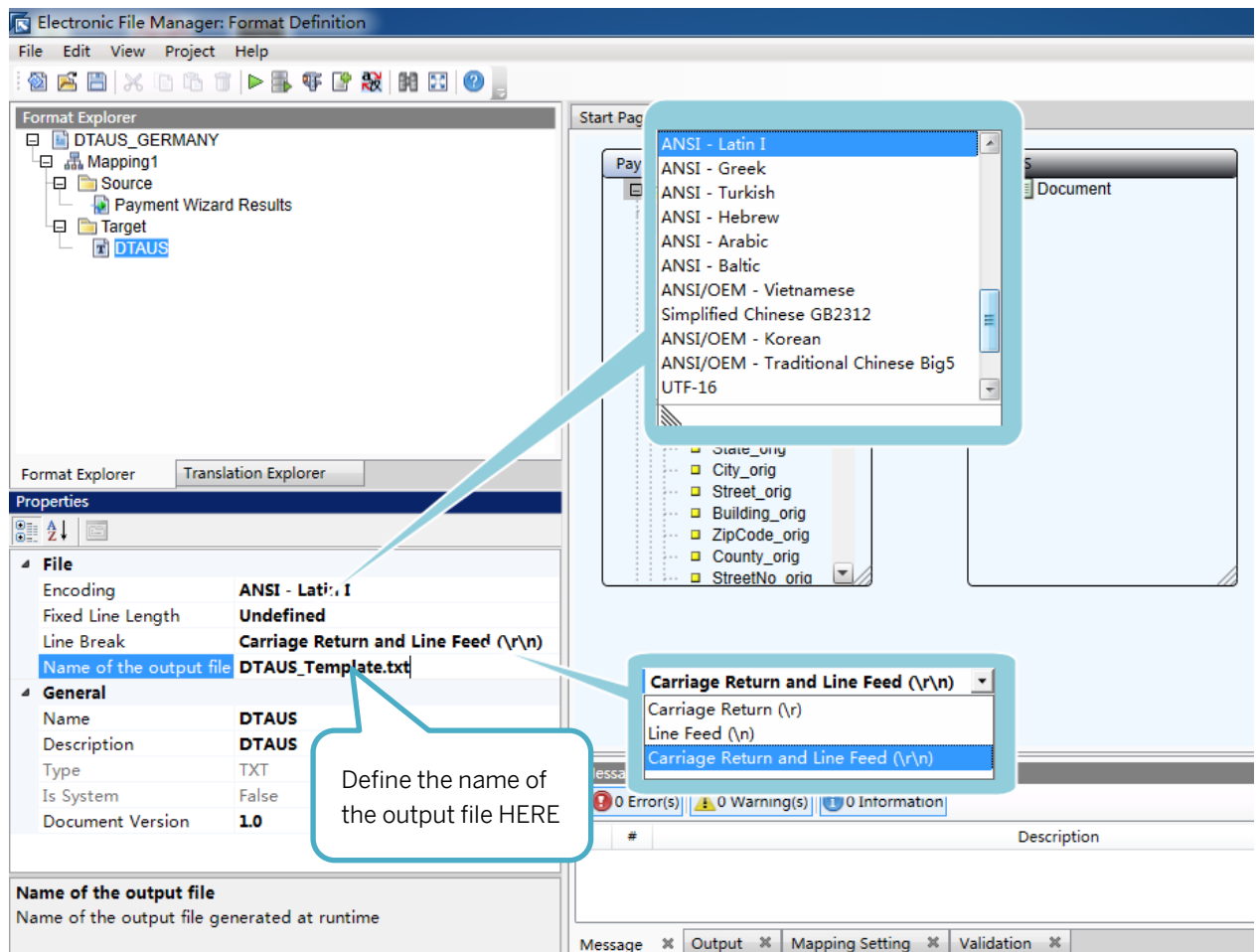
A new target node is available:



2. Define properties for the main node (BPP file). Specify the settings according to your needs.



3. Define properties for the target file (output file). Specify the settings according to your needs.



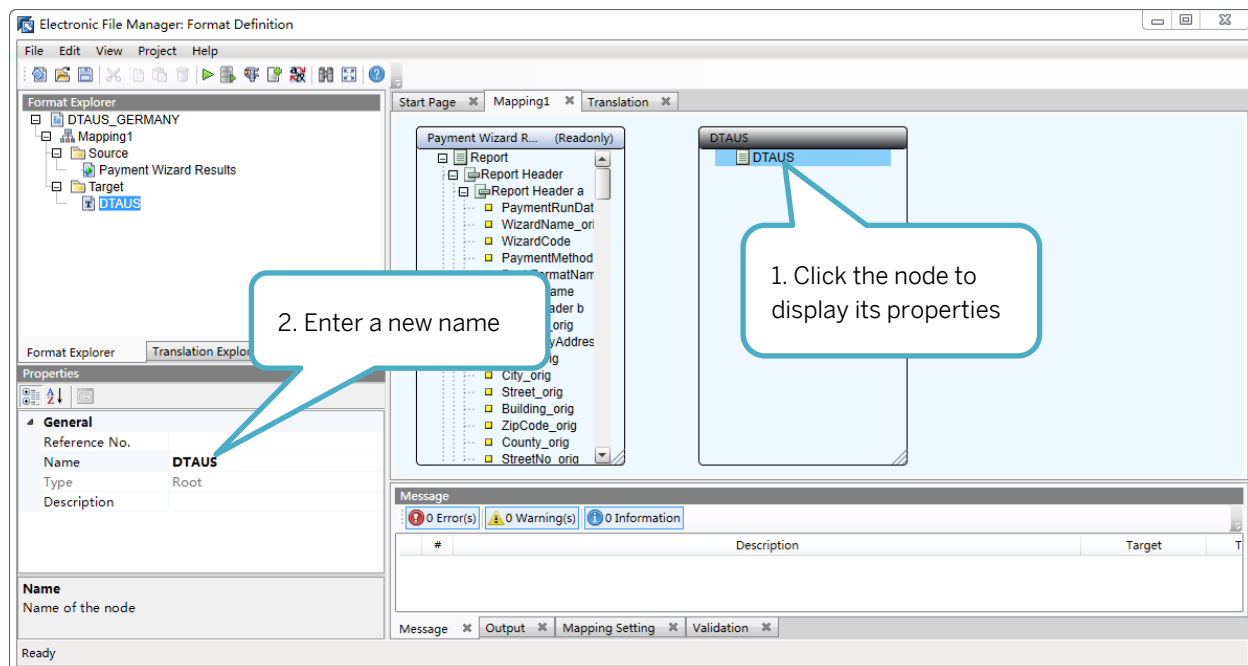
Step 2b: Defining the File Structure

According to the DTAUS format description, the format structure is as follows:

1. Record level A = data header
 1. Constant part - 1st record section
 2. Constant part - 2nd record section
 3. Variable part - 2nd record section
2. Record level C = single payment order
3. Record level E = data trailer

Procedure

1. Rename the root target node.

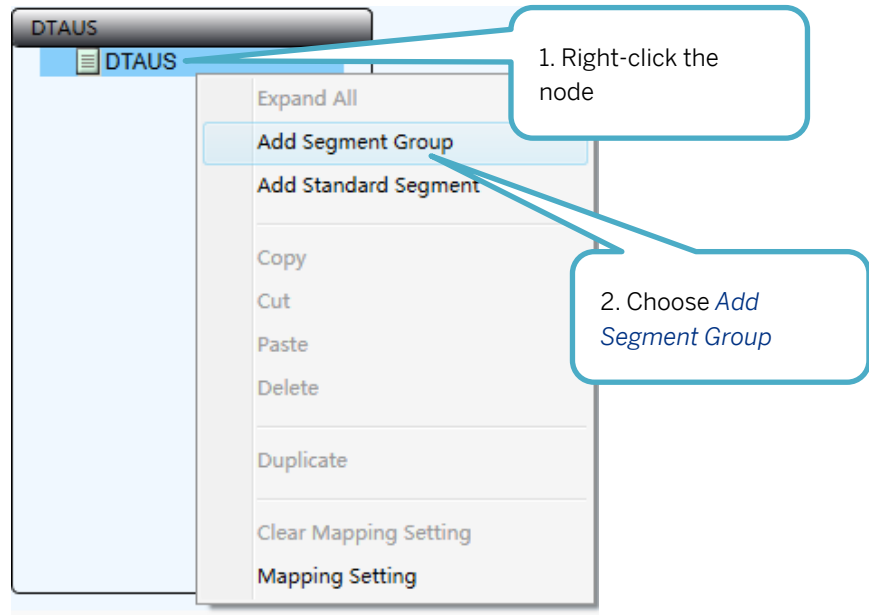


2. Define segment groups for record levels A, C, and E.

A segment group can contain standard segments and other segment groups. The segment group is used only to split target output into logical parts; you cannot add fields to segment groups.

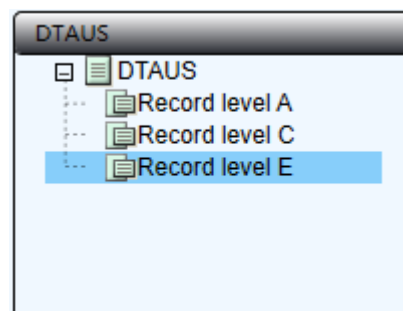
Use of segment groups and segments depends on the target file structure.

1. Add a segment group of record level A.



2. Rename the segment group as **Record level A**.
3. Define another two segment groups for record levels C and E, as above.

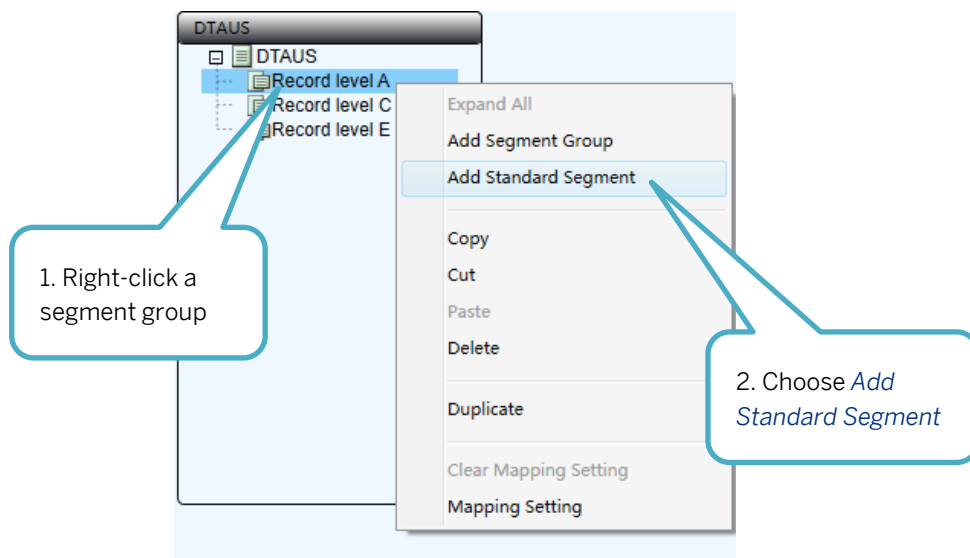
You have now defined the following segment groups:



3. Define standard segments for record levels A, C, and E.

A standard segment can contain only fields and can affect output by inserting line breaks. You need to define the following standard segments (for more information, see [Record Levels](#) in the appendix):

- o Data header (Record Level A segment group)
- o Constant part – 1st record section (Record Level C segment group)
- o Constant part – 2nd record section (Record Level C segment group)
- o Variable Part – 2nd record section (Record Level C segment group)
- o Data trailer (Record Level E segment group)

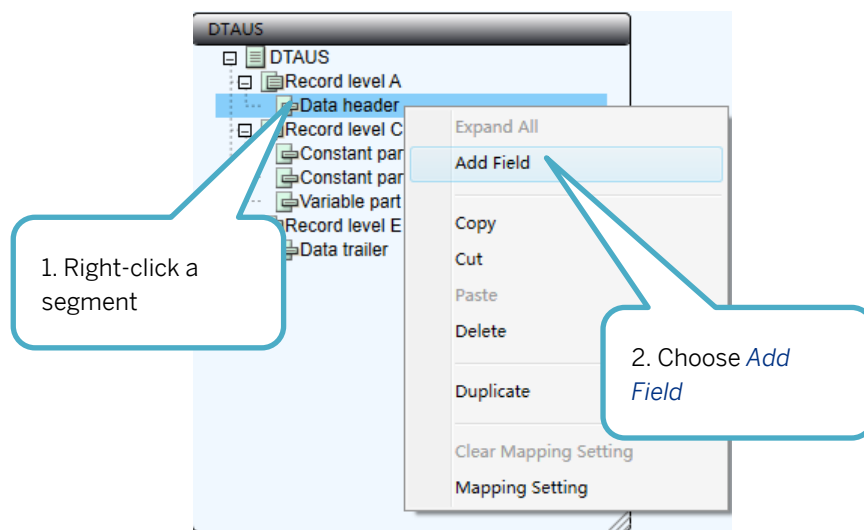


You have now defined the structure of the target file.

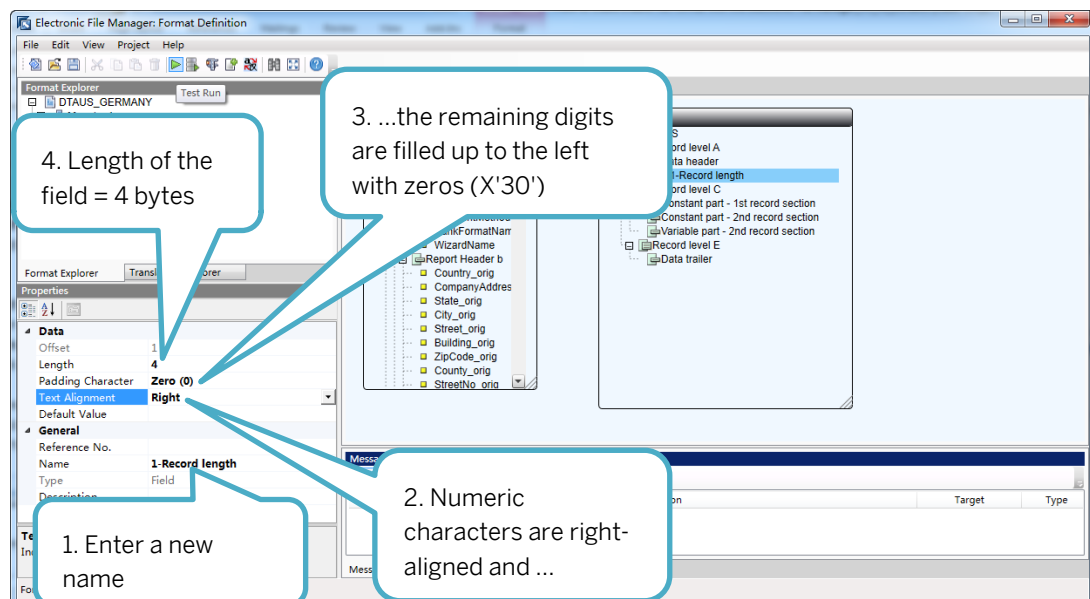
In the next step you create the fields in the target file.

4. Define fields and set properties.

1. Add a new field in the Data header segment:

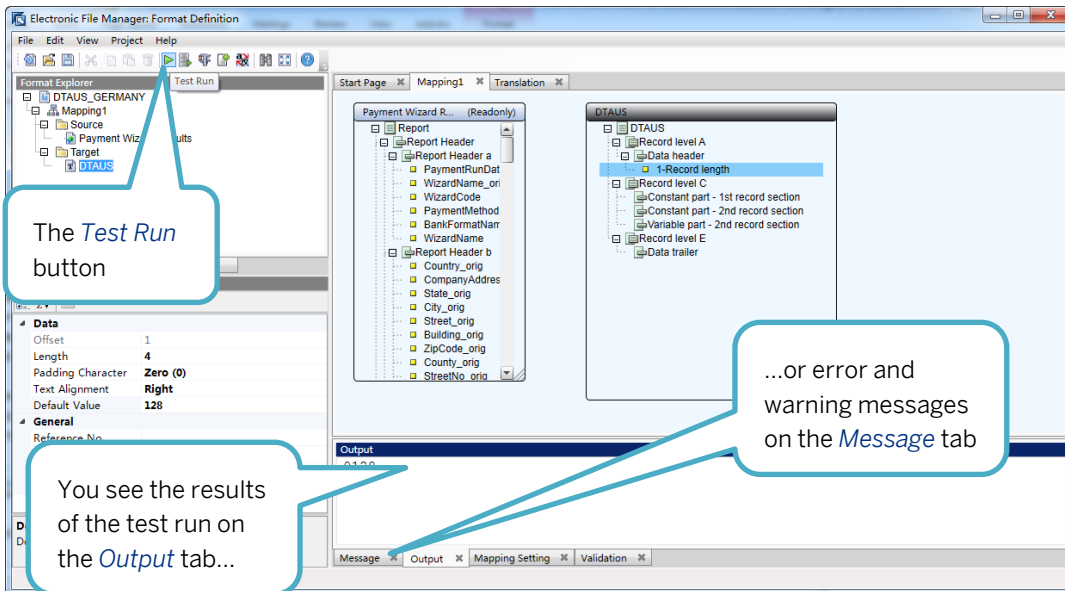


2. Define the properties of the new field. You can take all the necessary information from the file format description.



Tip:

Use the *Test Run*-button often to discover errors early.



3. Create all necessary fields in the same way. Define the properties of each field by using the values in the tables below:

Fields in standard segment "Data header"

Offset (not editable)	Length	Padding Character	Text Alignment	Name
1	4	Zero (0)	Right	1-Record length
5	1	Space	Left	2-Record type
6	2	Space	Left	3-Identifier
8	8	Zero (0)	Right	4-German bank code (receiver)
16	8	Zero (0)	Right	5-X'30'
24	27	Space	Left	6-Name of bank customer (sender)
51	6	Zero (0)	Right	7-Date file created (DDMMYY)
57	4	Space	Left	8-Blank (X'20')
61	10	Zero (0)	Right	9-Account number payer
71	10	Zero (0)	Right	10-Reference number of submitting customer
81	15	Space	Left	11a-Blank (X'20')
96	8	Space	Left	11b-Execution date (DDMMYYYY)

Offset (not editable)	Length	Padding Character	Text Alignment	Name
104	24	Space	Left	11c-Blank (X'20')
128	1	Space	Left	12-Currency attribute

Fields in standard segment "Constant part - 1st record section"

Offset (not editable)	Length	Padding Character	Text Alignment	Name
1	27	Space	Left	15-Name Payer/Payee
28	27	Space	Left	16-Remittance information
55	1	Space	Left	17a-Currency attribute
56	2	Space	Left	17b-Blank (X'20')
58	2	Zero (0)	Right	18-Number of extensions

Fields in standard segment "Variable Part - 2nd record section"

This part is not mandatory. In this specific case you want to use this part to include more detail remittance information in the payment file.

Offset (not editable)	Length	Padding Character	Text Alignment	Name
1	2	Zero (0)	Right	19-Identifier of extension
3	27	Space	Left	20-Remittance information
30	2	Zero (0)	Right	21-Identifier of the extension
32	27	Space	Left	22-Data of extension
59	11	Space	Left	23-Blank (X'20') – record section delimiter

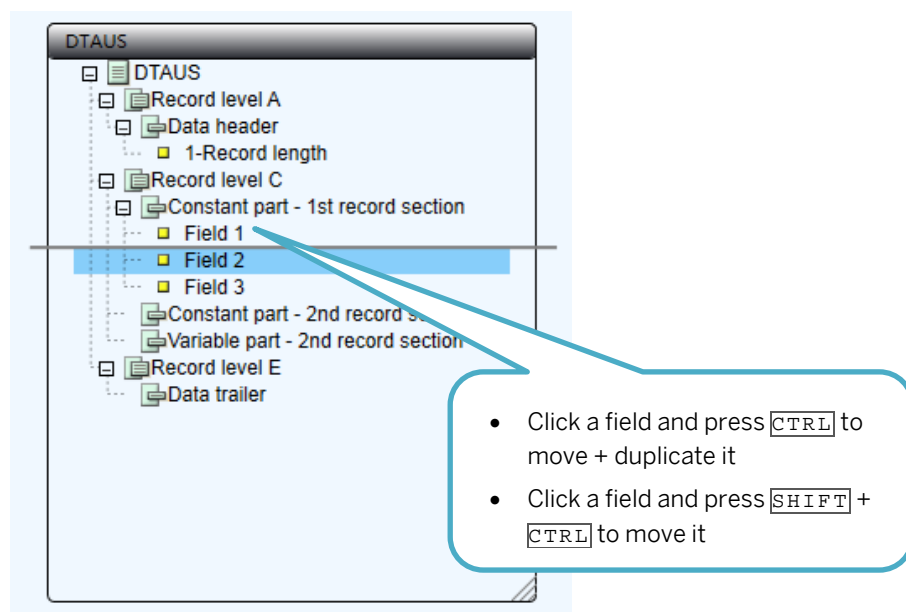
Fields in standard segment "Data trailer"

Offset (not editable)	Length	Padding Character	Text Alignment	Name
1	4	Zero (0)	Right	1-Record length
5	1	Space	Left	2-Record type (constant E)
6	5	Space	Left	3-Blank (X'20')
11	7	Zero (0)	Right	4-Number of C records
18	13	Zero (0)	Right	5-Zero
31	17	Zero (0)	Right	6-Arithmetic sum of account numbers of field 5 of C records

Offset (not editable)	Length	Padding Character	Text Alignment	Name
48	17	Zero (0)	Right	7-Arithmetic sum of the bank codes of field 4 of the C records
65	13	Zero (0)	Right	8-Arithmetic sum of euro amounts of field 12 of C records
78	51	Space	Left	9-Blank (X'20' - record section delimiter)

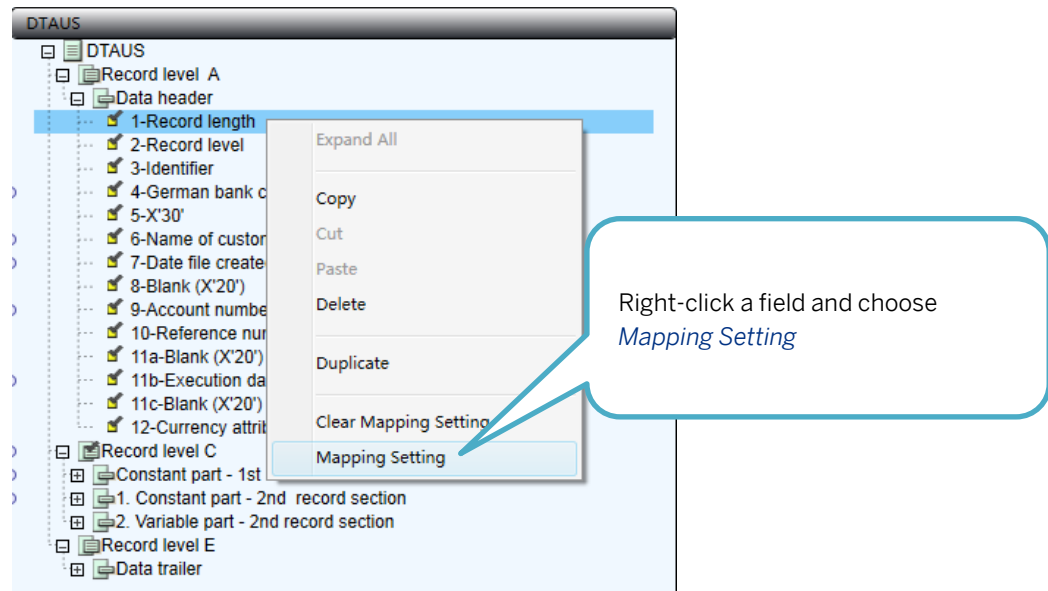
Tip:

You can change the position of a field as long you have not done the mapping settings for that field:

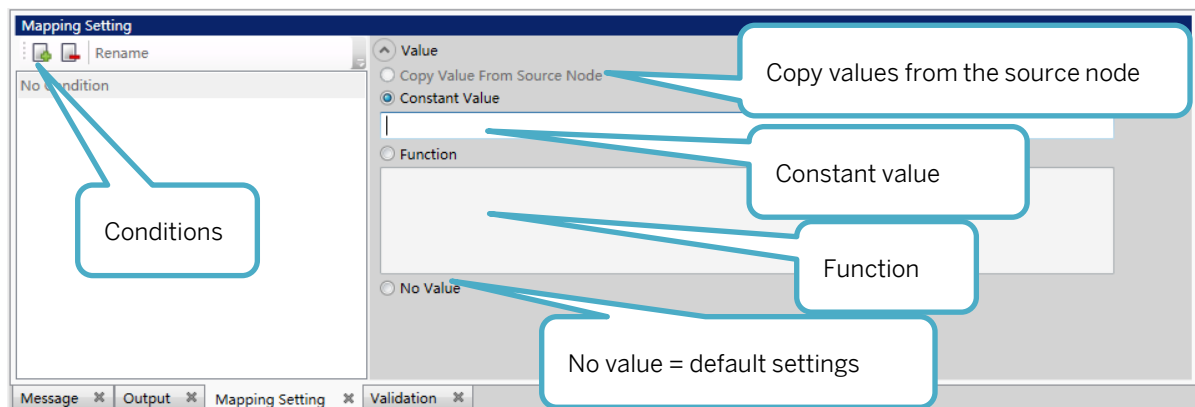


Step 3: Mapping Settings

After creating all required fields, you can proceed to map the source to the target nodes.



Basically, 5 different types of mapping settings are available:



All the mapping setting types are explained in the following sections, except for the *No Value* option.

Example: Copying Values from Source Node

1. Select the *BankCode* field in the source file tree.

2. Keeping the left mouse button pressed, connect *BankCode* with field *4-German bank code*

3. In the *Mapping Setting* window, the *Copy Value From Source Node* radio button is selected automatically.

The screenshot shows the 'Electronic File Manager: Format Definition' interface. The 'Format Explorer' on the left shows the source file tree with 'BankCode' selected. The 'Payment Wizard Results' window in the center shows the target file tree with '4-German bank code' selected. The 'Mapping Setting' window at the bottom shows the 'Copy Value From Source Node' radio button selected. The status bar at the bottom indicates 'Operation completed successfully'.

Result

The *4-German bank code (receiver)* field contains the value from the *BankCode* field from the *Payment Wizard Results*-file (source file).

Example: Defining Constant Value

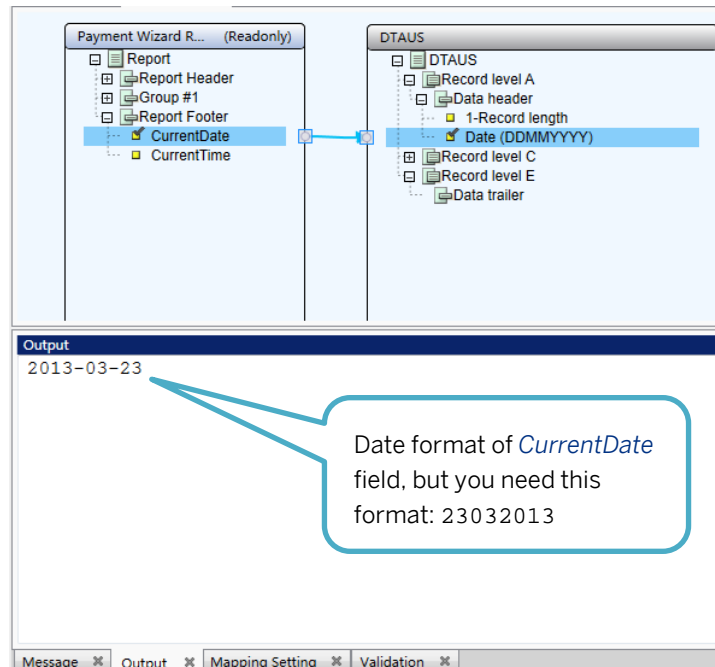
1. Select the *1-Record Length* field in the source file tree

2. Select the *Constant Value* radio button and then enter *128*.

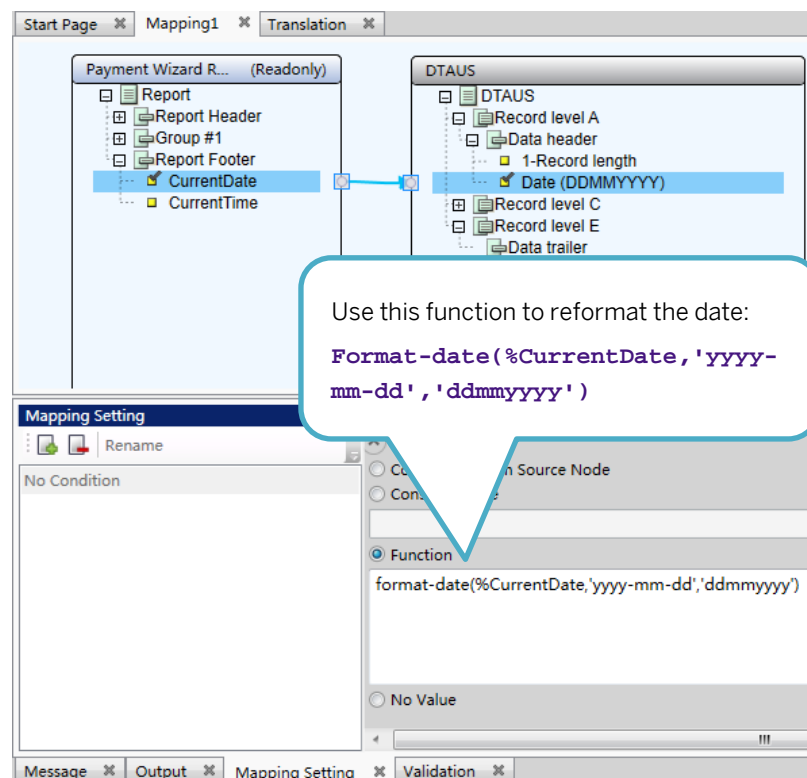
The screenshot shows the 'Electronic File Manager: Format Definition' interface. The 'Format Explorer' on the left shows the source file tree with '1-Record Length' selected. The 'Payment Wizard Results' window in the center shows the target file tree with '1-Record Length' selected. The 'Mapping Setting' window at the bottom shows the 'Constant Value' radio button selected and the value '128' entered. The status bar at the bottom indicates 'Operation completed successfully'.

Example: Defining Functions

In this example, the format of the *CurrentDate* field in the source file is not as expected by the payment file format. A reformatting is necessary.

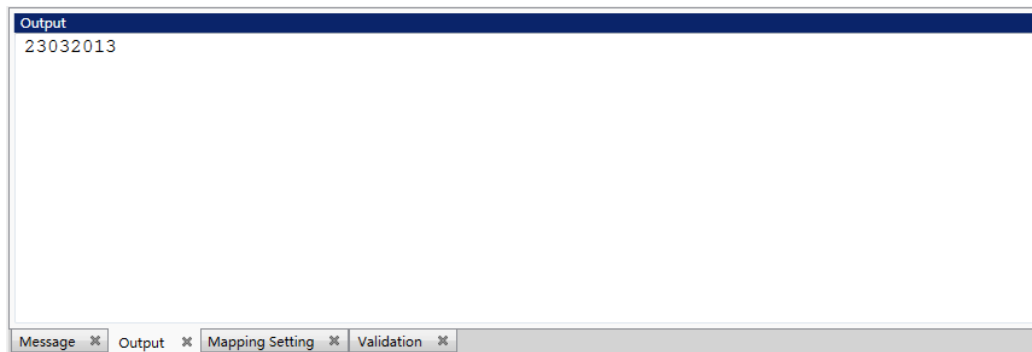


Therefore you use the format-date function:



Tip:

Instead of entering `%CurrentDate` in the function, we recommend that you pull the `CurrentDate` field from the *Payment Wizard Results* window directly into the format date function. To do this, press `CTRL`, and drag and drop the field into the function. If you enter the field name directly in the function window, EFM may not recognize the variable and may generate an error-message.

New date format:

To understand the meaning of the syntax of this and all other functions currently available in EFM, see the online help (available by pressing `F1` in the EFM). For more information, see [Appendix 3: Functions](#).

Example: Defining Conditional Functions

In this example, you need to define conditional functions for field "8-Arithmetic sum of euro amounts of field 12 of C records" in record level E. You need as the target format a maximum of 13 characters, including 2 decimal places, but without a delimiter. For example, 1.360,78 must be converted to 136078.

Problem

If you use the sum function in EFM, the decimal places that are zero will be cut off:



Example

100,50 > 100,5

100,00 > 100

Solution

Decimal places that were cut off need to be filled in again. You can use conditions to achieve this purpose.

In this example you need to define these conditions:

- Condition 1: If number of decimal places = 0 > fill in with '00' (100 > 10000)
- Condition 2: If number of decimal places = 1 > fill in with '0' and delete delimiter (100,0 > 10000)
- Condition 3: If number of decimal places = 2 > delete delimiter (100,00 > 10000)

Condition 1:

Expression:

Function

```
contains(sum(%PaymentAmountLC),'.')
```

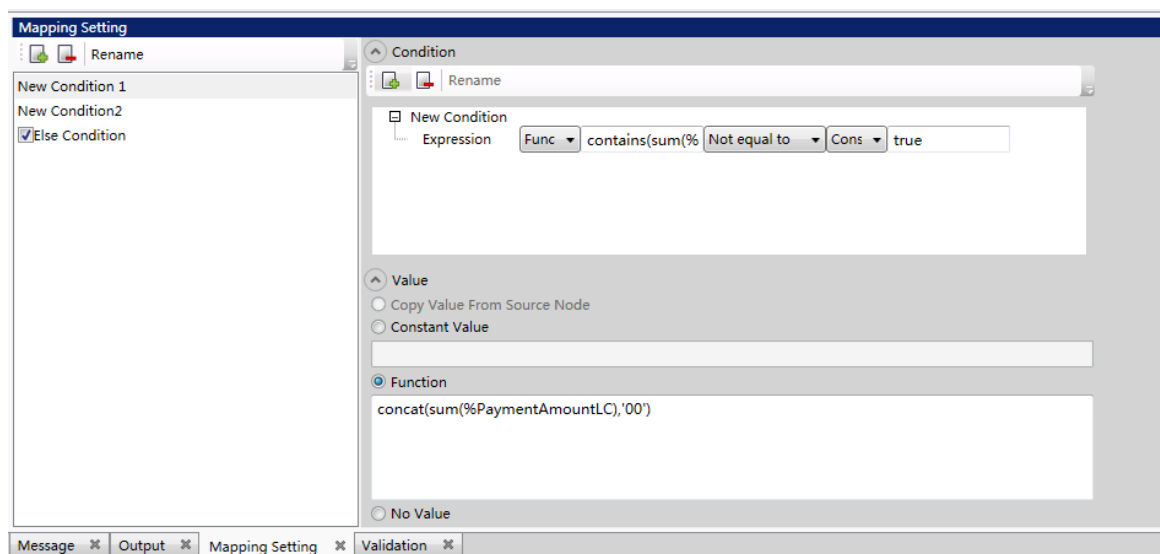
Not equal to

Constant

true

Function:

```
concat(sum(%PaymentAmountLC),'00')
```



Condition 2:

Expression:

Function

```
string-length(substring-after(sum(%PaymentAmountLC),'.'))
```

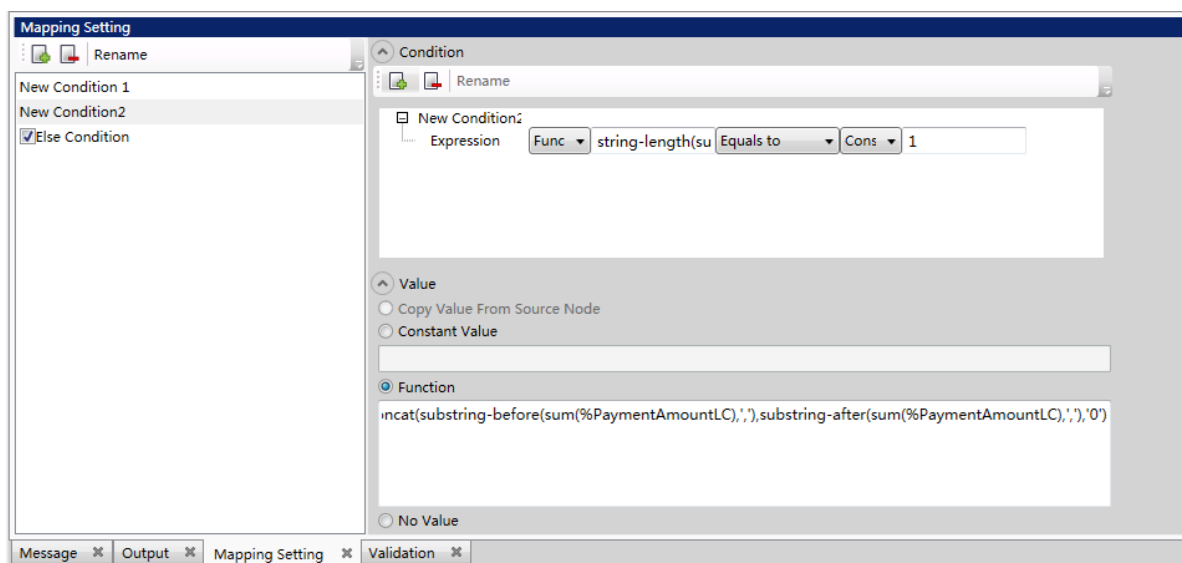
Equal to

Constant

1

Function:

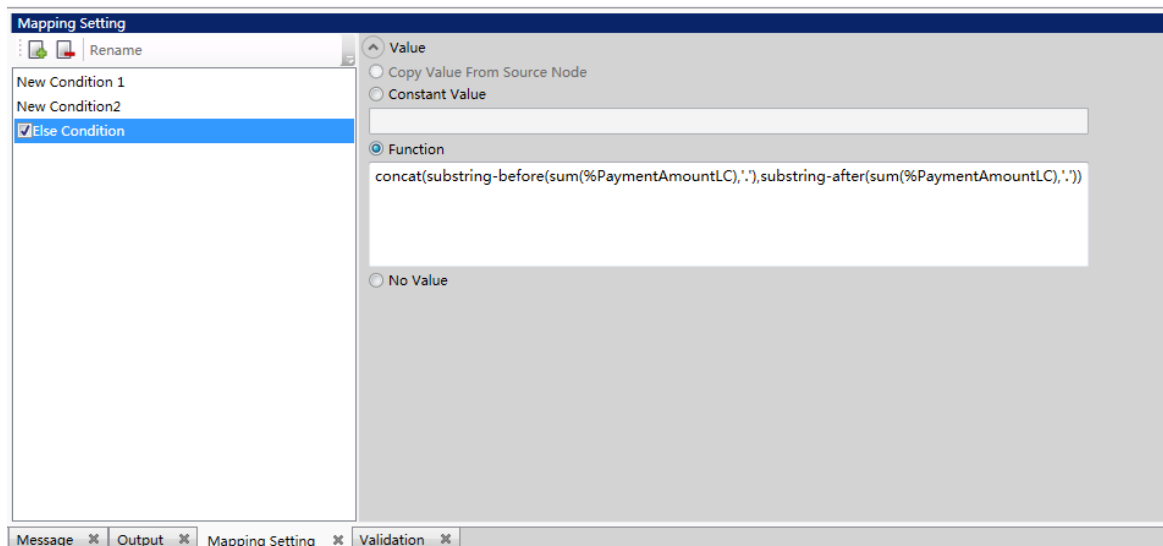
```
concat(substring-before(sum(%PaymentAmountLC),'.'),substring-  
after(sum(%PaymentAmountLC),'.'),'0')
```



Else Condition (condition 3):

Function:

```
concat(substring-before(sum(%PaymentAmountLC), '.'), substring-  
after(sum(%PaymentAmountLC), '.'))
```



Mapping Settings for DTAUS_EN.BPP

You should be basically familiar now with all types of mapping settings. Define the mapping settings for each field by using the settings from the tables below. If you want to learn/understand the meaning of the functions that are used, refer to [Appendix 3: Functions](#).

Important:

You must connect the node *Group Header 4* (contains payment information) in the source to the segment group *Record level C* in the target; otherwise, the data-sequence in the target file is not correct.

Example

You want to pay four documents from two different business partners (BP):

- o BP1: Doc1 + Doc2
- o BP2: Doc3 + Doc4

In the payment file you need this data sequence:

```
BankCodeBP1, AccountNoBP1, ..., SumBP1(Doc1+Doc2)...BankCodeBP2,  
AccountNoBP2, ..., SumBP2(Doc3+Doc4)...
```

If you do not connect *Group Header 4* to *Record level C*, you get this data sequence:

```
BankCodeBP1, BankCodeBP2, AccountNoBP1, AccountNoBP2, ..., SumBP1(Doc1+Doc2),  
SumBP2(Doc3+Doc4)...
```

The following tables list the mapping settings for the attached DTAUS_EN.BPP file format:

Mapping-Settings for Data Header

Target Field	Source-Field	Mapping Setting	Comment
1-Record length	/	Constant Value: 128	/
2-Record type	/	Constant Value: A	/
3-Identifier	/	Constant Value: GK	/
4-German bank code	Group #1 / Group header #1 / BankCode	Copy Value From Source Node	/
5-X'30'	/	No Value	/
6-Name of bank customer (sender)	Reportheader b/CompanyName	Function: upper (%CompanyName)	CompanyName in capital letters
7-Date file created (DDMMYY)	Reportfooter/CurrentDate	Function: format-date(%CurrentDate, 'yyyy-MM-dd', 'ddmmyy')	
8-Blank (X'20')		No Value	
9-Account number payer	Group #1 / Group header #1/BankAccount	Copy Value from Source Node	
10-Reference number of submitting customer		No Value	
11a-Blank (X'20')		No Value	
11b-Execution date (DDMMYYYY)	Reportfooter/CurrentDate	Function: format-date(%CurrentDate, 'yyyy-MM-dd', 'ddMMyyyy')	
11c-Blank (X'20')		No Value	
12-Currency attribute		Constant Value: 1	

Mapping-Settings for Record Level C / Constant Part – 1st Record Section

Target Field	Source Field	Mapping Setting	Comment
1-Record length		Constant Value: 245	
2-Record type		Constant Value: C	
3-Bank Code (first institution)		No Value	

Target Field	Source Field	Mapping Setting	Comment
4-Bank Code (destination institution)	Group #3 / Group header #3 / BPBankCode	Copy Value From Source Node	
5-Account number	Group #3 / Group header #3 / BPBankAccount_orig	Copy Value From Source Node	
6-Internal	Reportheader	No Value	
7a-Text key		Constant Value: 51	
7b-Text key extension		Constant Value: 000	
8-Bank internal field (blank – X'20')		No Value	
9-Zero		No Value	
10-Bank code first institution instructed	Group #1 / Group header #1/BankCode	Copy Value from Source Node	
11-Account number	Group #1 / Group header #1/BankAccount	Copy Value from Source Node	
12-Amount in EUR	Group #4 / Group header #4/PaymentAmountLC	Function: <code>number (concat (substring - before (%PaymentAmountLC , '.'), substring- after (%PaymentAmountLC, , '.')))</code>	Remove delimiter character
13-Blank (X'20')		No Value	
14a-Name Payee / Payer	Group #3 / Group header #3/BPName	Function: <code>upper (%BPName)</code>	BPName in capital letters
14b-Blank (X'20')		No Value	

Mapping-Settings for Record Level C / Constant Part – 2nd Record Section

Target Field	Source Field	Mapping Setting	Comment
15-Name Payer/Payee	Reportheader b/CompanyName	Function: <code>upper (%CompanyName)</code>	CompanyName in capital letters
16-Remittance information		Function: <code>upper (node- concat (%BPReferenceNum_ orig, '&'))</code>	Explanation please see page 43
17a-Currency attribute		Constant Value: 1	
17b-Blank (X'20')		No Value	

Target Field	Source Field	Mapping Setting	Comment
18-Extension character		Constant Value: 02	Please see page 5: 02 means that this extension contains remittance information

Mapping-Settings for Record Level C / Variable Part – 2nd Record Section

Target Field	Source Field	Mapping Setting	Comment
19-Identifier of extension		Constant Value: 02	
20-Remittance information		Function: upper (node-concat (%DocumentTotalLC, '&'))	Explanation see page 43
21-Identifier of the extension		Constant Value: 03	Please see page 5: 02 means that this extension contains payer/payee information
22-Data of extension		upper(%CompanyName)	
23-Blank (X'20')		No Value	

Mapping-Settings for Record Level E / Data Trailer

Target Field	Source Field	Mapping Setting	Comment
1-Record length		Constant Value: 128	
2-Record type		Constant Value: E	
3-Blank		No Value	
4-Number of C records		Function: count (%PaidDocumentCount)	
5-Zero		No Value	
6-Arithmetic sum of account numbers of field 5 of the C records		Function: sum (%BPBankAccount_orig)	
7-Arithmetic sum of bank codes of field 4		Arithmetic sum of account numbers of field 5	

Target Field	Source Field	Mapping Setting	Comment
of the C records			
8-Arithmetic sum of euro amounts of field 12 of the C records		<p>Condition 1: Expression: <i>Function</i> <code>contains(sum(%PaymentAmountLC),'.')</code> <i>Not equal to</i> <i>Constant: true</i> Function: <code>concat (sum (%PaymentAmountLC), '00')</code></p> <p>Condition 2: Expression: <i>Function</i> <code>string-length(substring-after (sum (%PaymentAmountLC), ' . '))</code> <i>Equal to</i> <i>Constant: 1</i> Function: <code>concat (substring-before (sum (%PaymentAmountLC), ' . '), substring-after (sum (%PaymentAmountLC), ' . '), '0')</code></p> <p>Else Condition: Function: <code>concat (substring-before (sum (%PaymentAmountLC), ' . '), substring-after (sum (%PaymentAmountLC), ' . '))</code></p>	Explanation see p. 35
9-Blank (X'20')		No Value	

Explanation of Function Node-Concat

In the *Payment Wizard Results* report you have, for example, three invoices for Business Partner No. 10000 that you want to pay. Assume that it is document No. 1 (100 EUR), No. 2 (50 EUR) and No. 3 (20 EUR).

In the bank file you create only one dataset for this Business Partner, with a total payment amount of 170 EUR. However, for information purposes, in this dataset you want to add the document number of each document in a reference field (see below).

Source Datasets

Doc. No.	BP No.	Amount
1	1000	100,00
2	1000	50,00
3	1000	20,00

Target Dataset

BP	PaymentAmount	PaymentReference
1000	170,00	DocNo 1&2&3

Therefore, you can use the function `node-concat(node-set, string spaceMark?)`.

Step 4: Assigning File Formats

The previous steps have helped you create a BPP file format. Now you need to import it into SAP Business One and assign it to appropriate payment methods. If not assigned, the format will not be available for data export in the payment run.

Procedure

1. From the SAP Business One *Main Menu*, choose *Administration* → *Setup* → *Banking* → *Payment Methods* → *Payment Methods-Setup* window.
2. Find the payment method you want to assign to your EFM file.
3. Open the *File Formats - Setup* window.

Payment Methods - Setup

Payment Method Code: Outgoing BT
Description: DTAUS payment file

Payment Type: Outgoing
Payment Means: Bank Transfer

House Bank:
Country: United Kingdom
Bank: Barclays Bank(House Bar
Account: 12345678
Branch: 201030
Control No.:
IBAN:
BIC/SWIFT Code:

Bank File Generation:
Key Code:
Transaction Type:
File Format: SAPBPIEOPBT_EFTS
☐ Debit Memo

Bank Charges:
Bank Charge Rate (%): 0.00

Payment Process:
☐ Group Invoices by Pay-To Address
☐ Group Invoices by Pay-To Bank
☐ Group Invoices by Currency
☐ Group Invoices by Due Date
☐ Post to G/L Interim Account

Payment Validation:
☐ Check Address
☐ Check Bank Details
☐ Check Collection Authorization
☐ Foreign Payment Block
☐ Foreign Bank Block
☐ Currency Selection
☐ Post Office Bank

Report: [Field]

OK Cancel

List of File Formats

Find: [Field]

#	File Format Description	Encoding Type
1	SAPBPIEOPBT_EFTS	
2	SAPBPIEOPBT_EMTS	
3	SAPBPIEOPBT_UBAMTS	
4	SAPBPUKOPBT_BACS	
5	SAPBPUKOPBT_BACS_BARCLAY	
6	SAPBPUKOPBT_BACS_TELEBANK	
7	SAPBPXOPBOE_OBOE	
8	SAPBPXOPDD_OPEX	
9	SAPBPXOPBT_OPEX	
10	SAPSEPASTANDARD_CT	

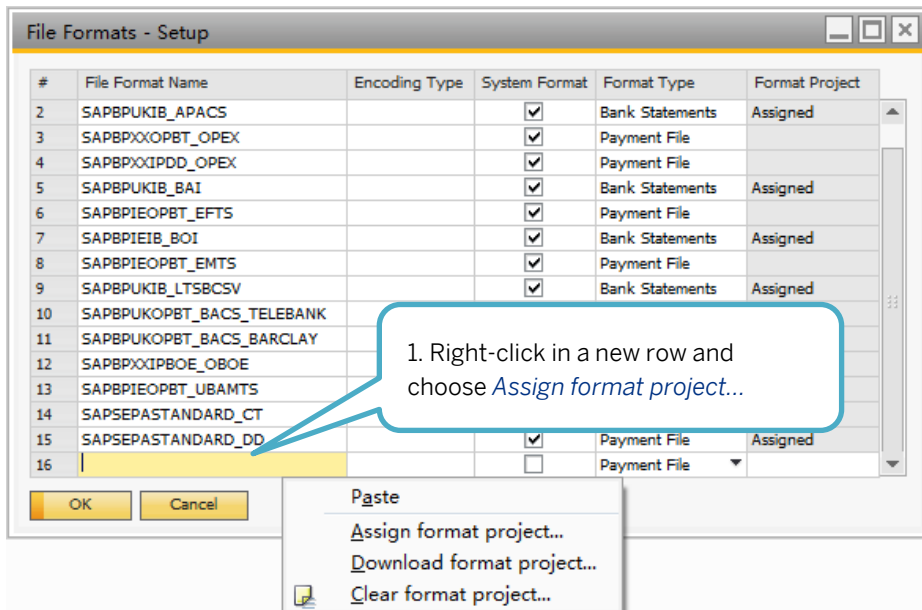
Choose Cancel New

File Formats - Setup

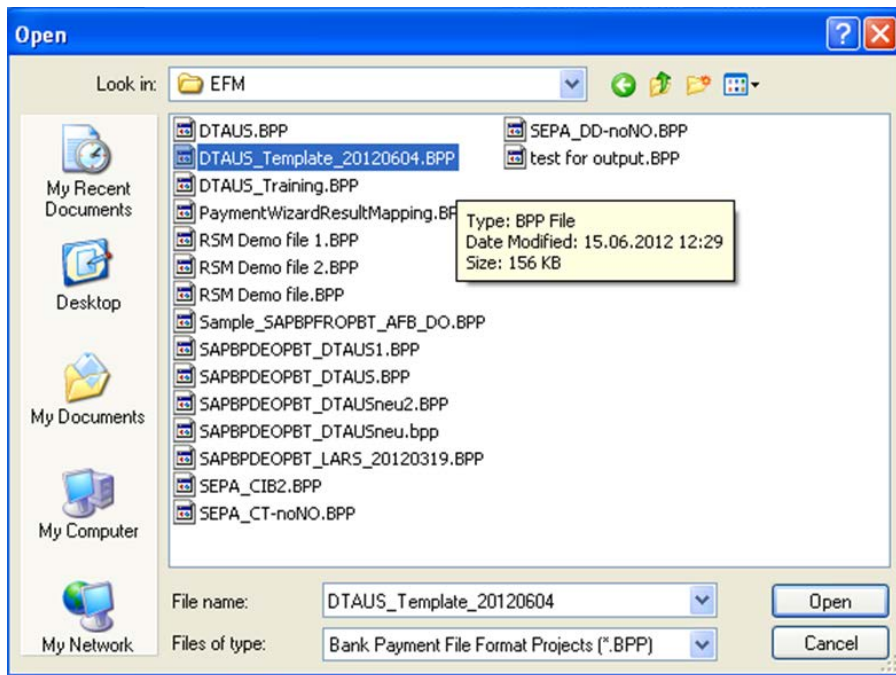
#	File Format Name	Encoding Type	System Format	Format Type	Format Project
2	SAPBPUKIB_APACS		<input checked="" type="checkbox"/>	Bank Statements	Assigned
3	SAPBPXOPBT_OPEX		<input checked="" type="checkbox"/>	Payment File	
4	SAPBPXOPDD_OPEX		<input checked="" type="checkbox"/>	Payment File	
5	SAPBPUKIB_BAI		<input checked="" type="checkbox"/>	Bank Statements	Assigned
6	SAPBPIEOPBT_EFTS		<input checked="" type="checkbox"/>	Payment File	
7	SAPBPIEIB_BOI		<input checked="" type="checkbox"/>	Bank Statements	Assigned
8	SAPBPIEOPBT_EMTS		<input checked="" type="checkbox"/>	Payment File	
9	SAPBPUKIB_LTSBCSV		<input checked="" type="checkbox"/>	Bank Statements	Assigned
10	SAPBPUKOPBT_BACS_TELEBANK		<input checked="" type="checkbox"/>	Payment File	
11	SAPBPUKOPBT_BACS_BARCLAY		<input checked="" type="checkbox"/>	Payment File	
12	SAPBPXOPBOE_OBOE		<input checked="" type="checkbox"/>	Payment File	
13	SAPBPIEOPBT_UBAMTS		<input checked="" type="checkbox"/>	Payment File	
14	SAPSEPASTANDARD_CT		<input checked="" type="checkbox"/>	Payment File	Assigned
15	SAPSEPASTANDARD_DD		<input checked="" type="checkbox"/>	Payment File	Assigned
16			<input type="checkbox"/>	Payment File	

OK Cancel

- Assign the format project.



- Browse to and select your BPP file.



- Save your changes.

Note

You must make sure the format type is **Payment File** for the BPP file.

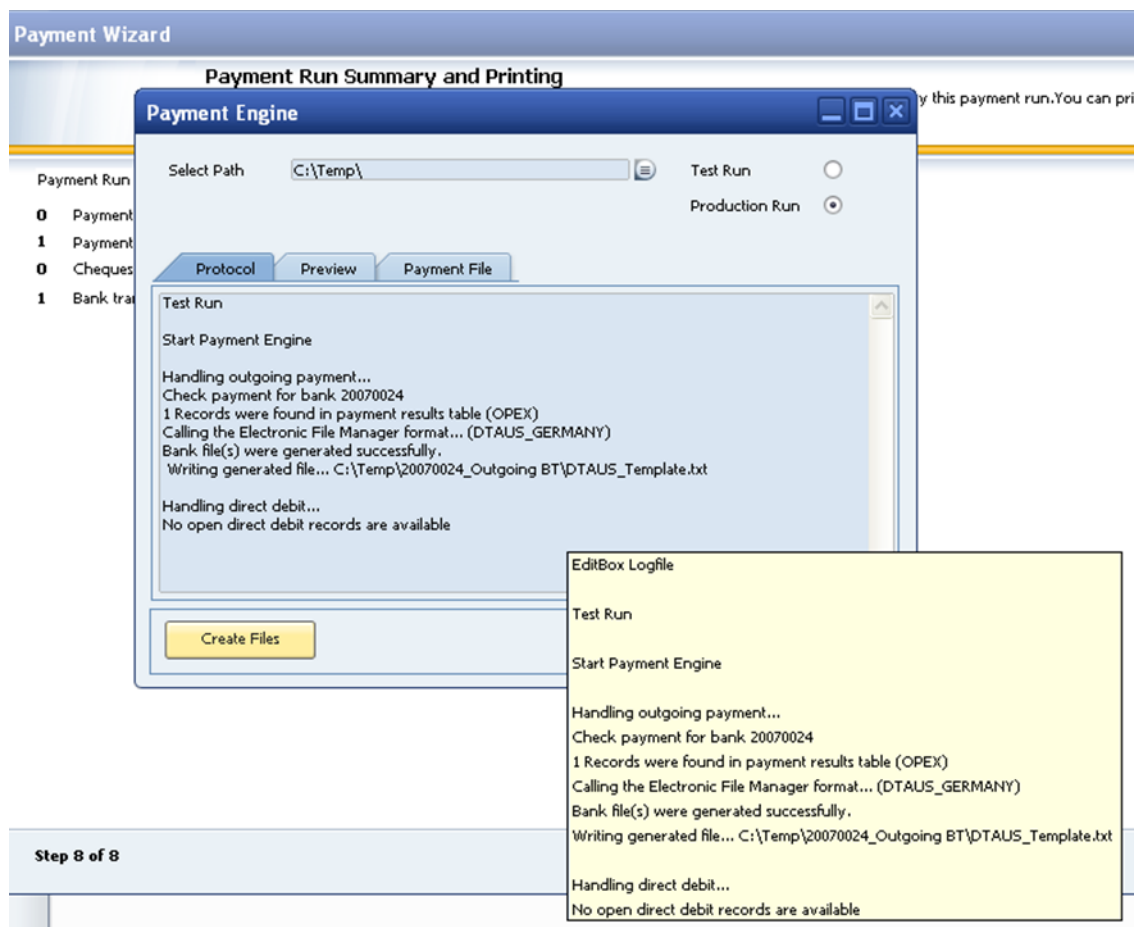
Step 5: Generating Payment Files

After you install and start the Payment Engine add-on, an additional *Bank File* button appears in step 8 of the payment wizard. You can choose this button to generate bank files for the payments created during a particular payment run.



Caution

Make sure that account numbers do not contain special characters; otherwise, you will get a plug-in error message caused by the sum function in record level E (trailer).



Your payment file is generated successfully and you can transfer it to the bank.

Appendix 1: DTAUS File Format Description

To design a DTAUS file format, you must first understand the structure of the DTAUS file.

The physical record length of each data record is 128 bytes. Record levels A and E consist of one physical record each with 128 bytes. Every data record C comprises at least two record sections (physical records) with 128 bytes each.

The following table lists the permitted character sets and their corresponding hexadecimal codes.

Permitted Character Set	Characters	Hexadecimal Code
Numeric characters	0 to 9	X '30' - X '39'
Upper-case letters	A to Z	X '41' - X '5A'
Special characters:		/
Blank	" "	
Full stop	"."	
Comma	" , "	
Ampersand	"&"	
Hyphen	"_"	
Slash	" / "	
Plus sign	" + "	
Asterisk	" * "	
Dollar sign	" \$ "	
Percent sign	" % "	
Special German characters are coded as follows:	"Ä"	X '5B'
	"Ö"	X '5C'
	"Ü"	X '5D'
	"ß"	X '7E'

Record Levels

The logical file structure of a DTAUS file must be as follows:

- [Record level A](#) = data header
- [Record level C](#) = single payment order
- [Record level E](#) = data trailer

Column Descriptions of File Format Description

The [Content](#) column of the file format description may display hexadecimal codes. For example, in field 5 of Record level A, X'30' means that you must fill this field with "0" (zero), as the hexadecimal code represents the character "0" (see table above). In this particular case, you must fill field 5 of record level A with 8 zeros, as the field is 8 bytes long (see column [Length in Bytes](#)).

The [Format](#) column indicates whether this field is numeric (n) or alphanumeric (an). **Alphanumeric characters in ASCII code are left-aligned and filled up to the right with blanks (X'20'). Numeric characters are right-aligned and the remaining digits are filled up to the left with zeros (X'30').**

Consider the comments in the [Explanation](#) column; for example, field 7 of record level A must be the date format.

Record Level A (Data Header)

Record level A contains the information of the sender (SAP Business One company that sends the payment file) and receiver (bank) of the file, and exists only once in each logical file. It is 128 bytes long.

Field	Length in Bytes	Format	Content	Explanation
1	4	n	Record length	Constant: '0128'
2	1	an	Record type	Constant: "A"
3	2	an	Identifier "GK" or "LK" "GB" or "LB"	Reference to Credit transfer (=G) or direct debit (=L), C2B (=K), B2B (=B)
4	8	n	German bank code	German bank code of the receiver
5	8	n	X'30'	B2B (bank to bank) only; zero otherwise
6	27	an	Name of bank customer	Initiating party (sender)
7	6	n	Date	Creation date of file (DDMMYY; D=day, M=month, Y=year)

Field	Length in Bytes	Format	Content	Explanation
8	4	an	X '20'	Blanks (bank internal field)
9	10	n	Account number	German account number of sender (payee of a direct debit/payer of a credit transfer), max 10 digits (right-aligned, empty digits set to zero). The bank customer makes or receives payments via this account (house bank account).
10	10	n	Reference number of the sender	Optional
11a	15	an	(X '20')	Reserve
11b	8	an	Execution date (DDMMYYYY)	Optional. The earliest execution date may be the file creation day (field A7) or, at most, up to 15 calendar days later than the day specified in field A7. If a particular date is provided in this data field, the period stipulated in paragraph 3, No. 4 of the Special Condition for Remote Data Transfer of at least 14 calendar days, is to be calculated from the scheduled execution date.
11c	24	an	Blanks (X '20')	Reserve
12	1	an	Currency attribute	"1" = Euro
	128			

Record Level C (Single Payment Order)

Record level C contains details of the orders to be executed (credit transfers or direct debits). It comprises a constant and a variable part.

Constant part, 1st record section

Field	Length in Bytes	Format	Content	Explanation
1	4	n	Record length	Logical record length (constant part with 187 bytes + extension(s) of 29 bytes), max. '0622)
2	1	an	Record type	Constant "C"
3	8	n	Bank code	German bank code: first financial institution involved, discretionary
4	8	n	Bank code	German bank code: destination financial institution/place of payment
5	10	n	Account number	German account number: payee (for credit transfer) / payer (for direct debit)
6	13	n	If not used: zeros	Field C6 can be filled as follows: 1st byte= 0 or 1 2nd - 12th bytes: internal customer number or zeros 13th byte=0
7a	2	n	Text key	Identifier for payment type and text key additions according to the section <i>Explanations of Fields 7a and 7b of Record C.</i>
7b	3	n	Text key extension	

Field	Length in Bytes	Format	Content	Explanation
8	1	an	X '20'	Bank internal field
9	11	n	Zero	Right aligned; reserve
10	8	n	Bank code	German bank code: first financial institution instructed/ first place of collection
11	10	n	Account number	German account number: payer (for credit transfer) / payee (for direct debit); right aligned
12	11	n	Amount in Euros, including decimal places	Right aligned
13	3	an	X '20'	Reserve
14a	27	an	Name	Payee (for credit transfer) / payer (for direct debit), left aligned
14b	8	an	X '20'	To be used as record section delimiter (must not contain any data)
	128			

Explanations of fields 7a and 7b of record C

Banks have defined standard text keys to identify the type of payment. Any special text keys specified for individual types of payment must always be used. This applies especially for wage, salary and pension payments (text key "53") and for employment savings benefits (text key "54"). Public institutions can use text key "56" to identify the wages and salaries they pay. The following entries are possible for data fields 7a and 7b:

Text Key (Field 7a)	Text Key Addition (Field 7b)	Content of Field 7	Explanation
04	000	'04000'	Direct debit (Pre-authorized payment order procedure)
05	000	'05000'	Direct debit (Direct debit authority procedure)
05	005	'05005'	Direct debit from POS transaction- electronic cash

Text Key (Field 7a)	Text Key Addition (Field 7b)	Content of Field 7	Explanation
05	006	'05006'	Direct debit from POS transaction (with foreign credit card) - Maestro / magnetic strip
05	008	'05008'	Direct debit from credit card turnover
05	010	'05010'	Direct debit from POS transaction (with foreign credit card) - Maestro / EMV
05	011	'05011'	Direct debit from POS transaction - electronic cash, magnetic strip track 2, EMV
05	015	'05015'	Direct debit from POS transaction - POZ
05	019	'05019'	Direct debit from POS transaction - German ELV procedure
05	021	'05021'	Direct debit from POS transaction (with foreign credit card) EAPS/EMV and magnetic strip
51	000	'51000'	Credit of a credit transfer (e.g. commercial payment)
51	505	'51505'	Correction - Direct debit from POS transaction- electronic cash
51	506	'51506'	Correction - Direct debit from POS transaction (with foreign credit card) - Maestro / magnetic strip
51	510	'51510'	Correction - Direct debit from POS transaction (with foreign credit card) - Maestro / EMV
51	511	'51511'	Direct debit correction from POS transaction - electronic cash, magnetic strip track 2, EMV

Text Key (Field 7a)	Text Key Addition (Field 7b)	Content of Field 7	Explanation
51	521	'51521'	Direct debit correction from POS transaction - (with foreign credit card) EAPS/EMV and magnetic strip
53	000	'53000'	Wages, salary, pension credit
54	XXJ	'54XXJ'	Employment savings benefits (VL)
56	000	'56000'	Payments of public institutions
67	000	'67000'	Remittance credit with check-sum - protected processing instructions
68	000	'68000'	Credit from blank remittance / payment form
69	000	'69000'	Credit for remittance for charitable contributions

Constant part, 2nd record section

Field	Length in Bytes	Format	Content	Explanation
15	27	an	Name	Payer (for credit transfer) / payee for direct debit); left aligned, names used should be as short as possible
16	27	an	Remittance information	Information given should be as brief as possible. The information must refer exclusively to the payment transaction at hand. At the start of the data field, the information should be entered left aligned. The payee (for credit transfer)

Field	Length in Bytes	Format	Content	Explanation
				/ payer (for direct debit) may want to access the information to check it manually, or, in the case of a direct debit, the payee needs this information if the payment cannot be credited and is sent back to the payee unpaid.
17a	1	an	Currency attribute	"1"= Euro
17b	2	an	X '20'	Reserve
18	2	n	Number of extensions	00 = no extension following 01-15 =number of extensions of 29 bytes

Variable part, 2nd record section (continued)

This variable part and the constant part form a single unit. It is provided only if additional information has to be entered that exceeds the data fields in the constant part. Up to 6 record sections of 128 bytes each can be specified for record C. It may contain all the following:

- 1 extension for payee (for a credit transfer) or payer (for a direct debit) **(01)**
- Up to 13 extensions for remittance information (all **02**)
- 1 extension for payer (for a credit transfer) or payee (for a direct debit) **(03)**.

Field	Length in Bytes	Format	Content	Explanation
19	2	n	Identifier of extension	01 = Name of the payee (for credit transfer) or payer (for direct debit) 02 = Remittance information 03 = Name of the payer (for credit transfer) or payee (for direct debit)
20	27	an	Payee (for credit transfer) or payer	Left aligned Basically, the bank

Field	Length in Bytes	Format	Content	Explanation
			(for direct debit) / remittance information / payer (for credit transfer) or payee (for direct debit)	always returns remittances and direct debits without the content of the extensions, under "remittance information". For this reason, the payer (for credit transfer) or payee (for direct debit) must include the necessary remittance information in the constant part of record C (see explanations to field C 16).
21	2	n	Identifier of the extension	(as for field 19)
22	27	an	Data of the extension	(as for field 20)
23	11	an	X '20'	Used as record section delimiter (should not be taken into account when stating the record length in field C 1)
	128			

Variable part, 3rd record section

Field	Length in Bytes	Format	Content	Explanation
24	2	n	Identifier of extension	(as for field 19)
25	27	an	Data of extension	(as for field 20)
26	2	n	Identifier of extension	(as for field 19)
27	27	an	Data of extension	(as for field 20)

Field	Length in Bytes	Format	Content	Explanation
28	2	N	Identifier of extension	(as for field 19)
29	27	an	Data of extension	(as for field 20)
30	2	n	Identifier of extension	(as for field 19)
31	27	an	Data of extension	(as for field 20)
32	12	an	X '20'	Used as record section delimiter (should not be taken into account when stating the record length in field C 1)
	128			

The 4th to 6th record sections are available for any additional extensions that may be necessary. The structure of the 4th and 5th sections corresponds to that of the 3rd section. Record section 6 contains only one extension.

Note

Our example does not describe how to create the 3rd record section.

Record E (Data Trailer)

Record E is used for performing checks. It occurs only once in each logical file.

Field	Length in Bytes	Format	Content	Explanation
1	4	n	Record length	Constant: '0128'
2	1	an	Record type	Constant: "E"
3	5	an	X '20'	Reserve
4	7	n	Number of C records	Used for performing checks
5	13	n	Zero	Reserve, right aligned
6	17	n	Arithmetic sum of account numbers of field 5 of the C records	Used for performing checks
7	17	n	Arithmetic sum of the bank codes of	Used for performing checks

Field	Length in Bytes	Format	Content	Explanation
			field 4 of the C records	
8	13	n	Arithmetic sum of the Euro amounts of field 12 of the C records	Used for performing checks
9	51	an	X '20'	Used as record section delimiter
	128			

Checking Records C

After receipt and before transmission of a file in diskette format, you must check the C data records manually as follows:

Field	Content	Data Format
German bank code of destination financial institution / place of payment (Field C 4)	Must be a valid bank code as per the directory of the Deutsche Bundesbank, first digit neither 0 nor 9	n
German account number of the payee (for credit transfer) / payer (for direct debit) (Field C 5)	Not equal to zero	n
Internal customer number (Field C 6)	1st byte equal to zero	n
Text key -Direct debit -Credit transfers (Field C 7a)	Equals 04, 05 Equals 51, 53, 54, 56	n
German bank code: first financial institution instructed/first place of collection (Field C 10)	1st digit not equal to 0 or 9	n
German account number: payer (for credit transfer) / payee (for direct debit) (Field C 11)	Not equal to zero	n
Amount (Field C 12)	Not equal to zero	n
Name of the payee (for credit transfer) / payer (for direct debit) (Field C 14)	Not equal to X '20'	an
Name of the payer (for credit transfer) / payee (for direct debit) (Field C 15)	Not equal to X '20'	an
Currency attribute (Field C 17a)	"1" = Euro	an
Number of extensions (Field C 18)	Equals 00-15	n
Identifier of extension (Field C 19; C 21; c 26; etc., variable part)	Equals 01, 02, 03, and so on, in ascending order: 01 no more than once 02 no more than 13 times 03 no more than once	n

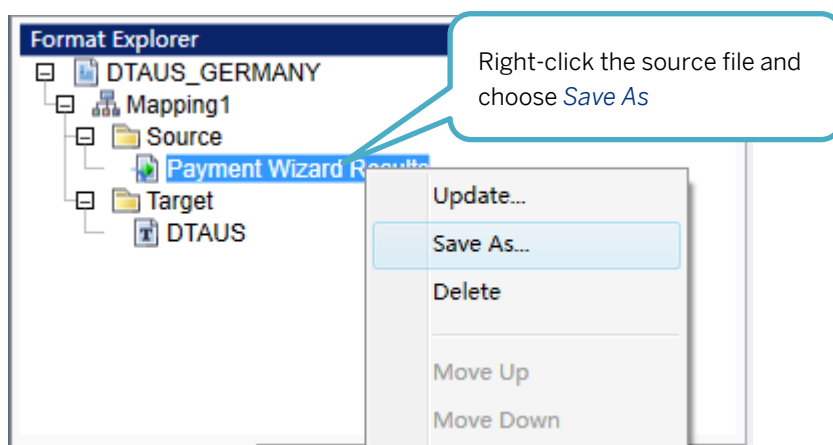
The check sums obtained by adding the number of C records, the [Amount](#) field (C 12), "the German account number of the payee (for credit transfer)/payer (for direct debit)" (C 5), and "German bank code of the destination financial institution/place of payment" (C 4) have to match the check data in record E.

Appendix 2: Payment Wizard Results Source

To understand the content and structure of the *Payment Wizard Results* file, you can export this file as an RPT file and open it with SAP Crystal Reports.

Procedure

1. Export the source file *Payment Wizard Reports*:



2. Open the exported file in SAP Crystal Reports (or the version for SAP Business One). The following figures are the preview of the report:

Payment Run Results Report

v1.43

Disclaimer: This report is designed to work with Electronic File Manager to export bank payment files from SAP Business One. It is not intended for preview and print by end users.

This report layout could be modified by SAP as the time goes by, so be cautious if you want to use it beyond the intended purpose.

Payment Run Date	04.09.2012 00:00:00
Bank Format Name	SAPSEPASTANDAR
Wizard Name	Wiz20120904n1 Wiz20120904n1
Wizard Code	62
Payment Method	Outgoing BT

Taken from report header a

Company Details

Company Name	OCE Computers OCE Computers	Zip Code	
Company Address	GERMANY	County	
Country	DE	Street No.	
State	DE	Phone1	
City		Phone2	
Street		Email	info@oec.de
Building		Company Tax Num	DE499003845
Federal Tax ID	DE499003845 DE499003845		
Company Address Line	GERMANY		

Taken from report header b

House Bank Account Details

Swift Num	DEUTDEBHAM	Bank User Num 3	
Bank Country	DE	Bank User Num 4	
Bank Code	20070024	IBAN	DE86200700240778893800
Bank Account	7788938	ISR Type	2
Bank User Num		ISR Biller ID	
Bank User Num :		BISR	N
Bank Branch	Deutsche Bank Hamburg	IBAN validation	DE86200700240778893800
		Swift Num - Clean	DEUTDEBHAM

Taken from
group header
#1

Payments by DocDueDate 04.09.2012

BP Information

BP Name	VAN PLC	BP Bank Code	10070024
BP Type	Vendor	BP IBAN	DE30100700240333333300
BP Zip	76133	BP Bank Swift No.	DEUTDEBBER
BP City	76133	BP Bank Street	Ollenhauerstraße 3
BP Street	Karlsruhe	BP Bank Zip	13403
	Karlsruhe	BP Bank City	Berlin
BP Street No.	Kaiserstr. 132	BP Bank Country Key	
BP Country	Kaiserstr. 132	BP Bank Country	DE
BP State	DE	BP ISR Biller ID	
BP State Name	DE	BP ISR Type	2
BP Building	Baden-Württemberg	BP Mandate ID	01
	Baden-Württemberg	BP Mandate Sign Date	01.01.2012
BP Code	V70000	BP Federal Tax ID - Business	
BP Bank Name	Deutsche Bank Privat und C	BP Federal Tax ID - Individu	
BP Bank Account	3333333	BP Textobjekt	Company/Private C
BP Bank Branch	Berlin Filiale	BP IBAN - validation	DE30100700240333333300
		BP Bank Swift No. - clean	DEUTDEBBER
BP Address	Kaiserstr. 132, , 76133, Karlsruhe, Baden-Württemberg, DE		

Taken from
report group
header #3

Payment Info

Payment Doc. Num.	111	Transaction Id	2067
Payment Amount FC	0.00	Payment Series	15
Payment Amount LC	2380.00	Payment Series Name	Primär
	2380	Payment Doc. Date	04.09.2012 00:00:00
Payment Currency	EUR	Payment Doc. Due Date	04.09.2012
	EUR	Payment Intended Purpose	
Payment Exchange Rate	1.00	Payment Counter Reference	
Paid Document Count	1	Payment Comments	
Payment Post. Date	04.09.2012 00:00:00	Ref1	111
Payment Reference		Ref2	
Payment Means	Outgoing BT	Payment Type	46
Payment Type - End To End	46/111		

Taken from
report group
header #4

Payment Documents Info			
BP Reference Num.	...	Document Remarks	...
Document Type	18	Document Payment Reference	...
Document Type Description	A/P Invoice	Document Currency	EUR
Document Total LC	2380.00	Document Num.	390
Document Total FC			390
Document Total SC	2380.00	Document Payment Amount FC	2380.00
Document Tax LC	380.00	Amount Before Discount FC	2380.00
Document Tax FC		Net Amount Before Discount FC	2380.00
Document Balance Due FC	2380.00	Document Post. Date	04.09.2012 00:00:00
Document Discount Percentag	0.00		04.09.2012
Document Discount FC	0.00	Document Value Date	04.10.2012 00:00:00
		Line Id	1
		Doc. Payment Amount With Currency	EUR 2380.00

Taken from
report group
header #4 -
Details

BP bank information from Paid Document (in case of Single Payment - ONLY)			
3333333	10070024	DE	DE3010070024033333300

1

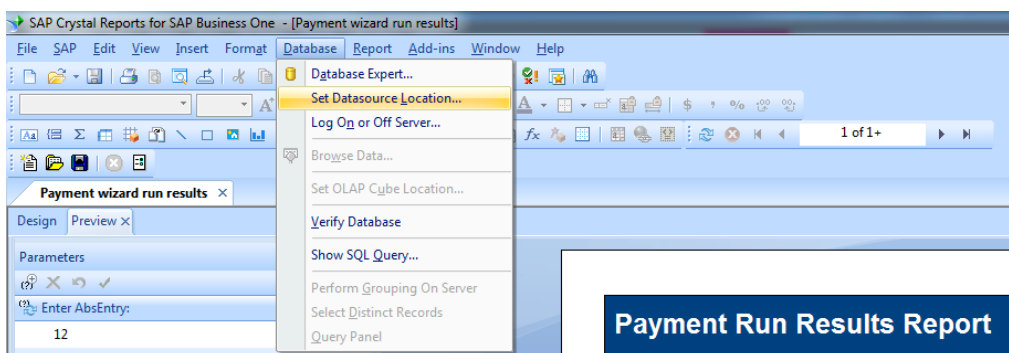
Report update on 04.09.2012 16:42:30

Changing Database Connection

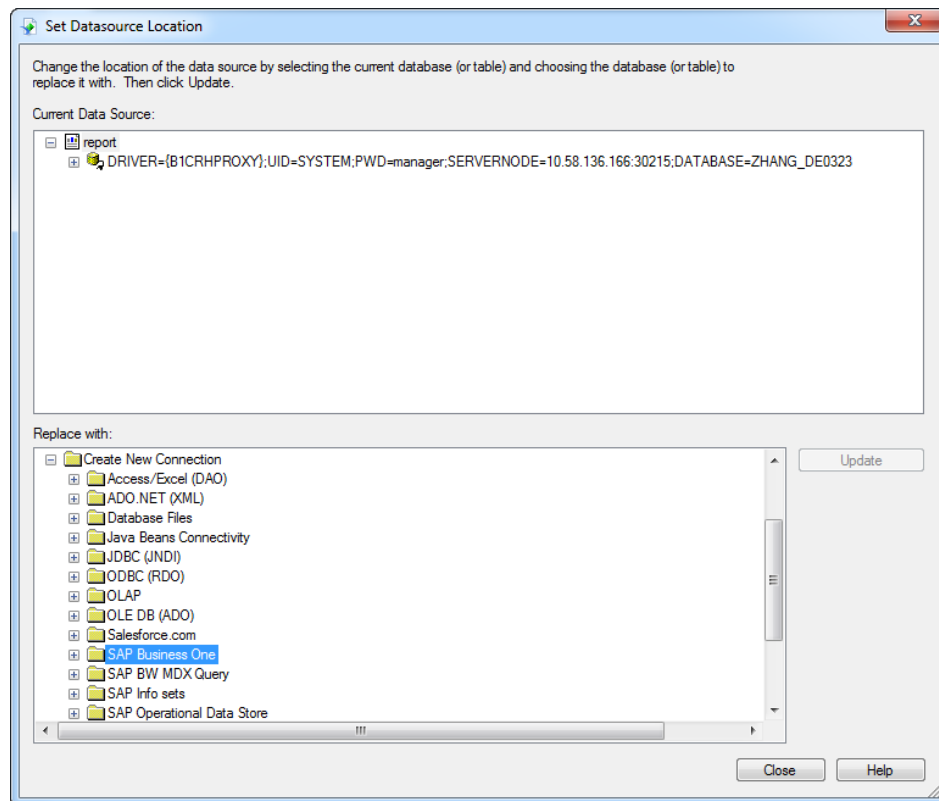
If you want to see data from a payment run that you conducted in your local database, you first need to change the database connection.

Procedure

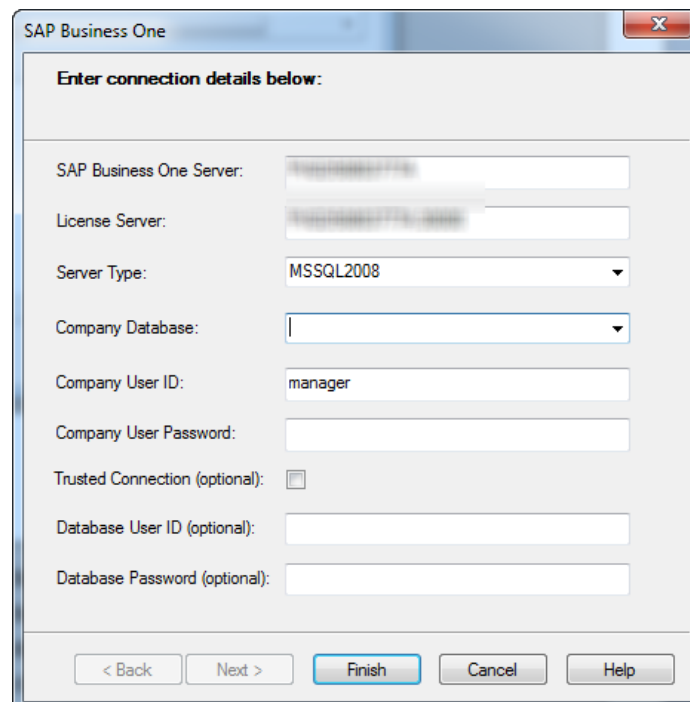
1. From the *Database* menu, choose *Set Datasource Location...*



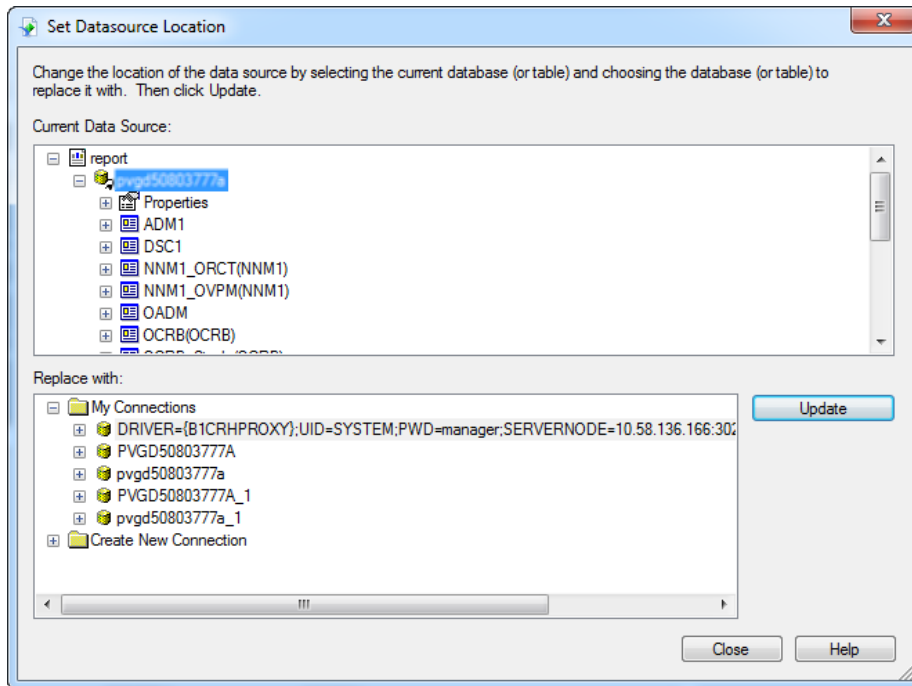
- To choose a new connection, in the lower pane, under *Create New Connection*, select *SAP Business One*.



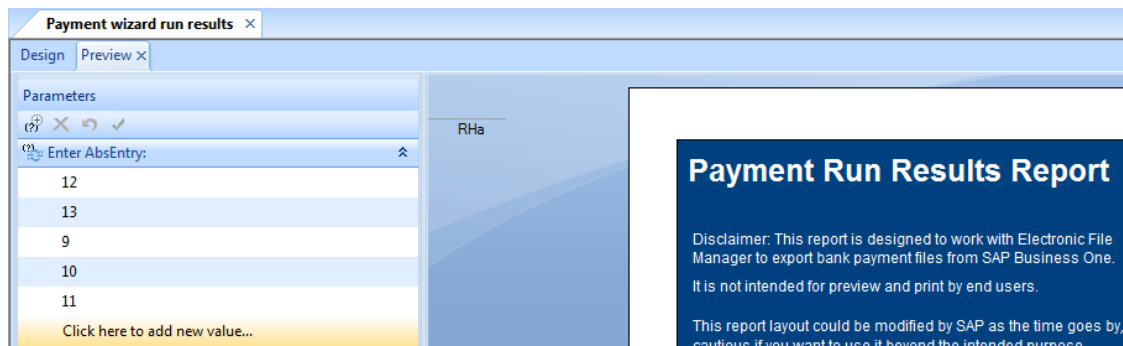
- Specify necessary information to connect to your company database.



4. Select the data connection you have just created.



5. In the preview mode, enter an AbsEntry of a payment run from your database.



Tip:

You can locate a valid AbsEntry by running the following query in the SQL Server studio or the SAP Business One client application:

```
SELECT * FROM OPEX T0
```

Appendix 3: Functions

For a detailed explanation of all functions available for the EFM, you can refer to the online help that is available by pressing **F1** in the EFM.

Note

The EFM does not support XPath functions.

floor(number)

In addition to the functions in the online help, the floor(number) function can also be used.

Description: Returns the largest integer that is not greater than the argument

Sample:

floor(3.5) = 3

floor(-1.3) = -2

floor(4) = 4



www.sap.com/contactsap

Material Number

© 2013 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System ads, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, xApps, xApp, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.