EFT Formats

Sage Intacct EFT Processing includes a number of system defined file formats. You need to import the formats suitable for your banking requirements.

For technical details on EFT Formats in Sage Intacct, see EFT Formats Technical Info on page 3

Tip: Send the bank specifications to your Sage partner who will liaise with Orchid Support to obtain the format file. Orchid Systems can created new formats if required.

HOW TO USE THE SCREEN

To open the screen, choose Orchid EFT > EFT Formats.

Add a format by clicking the **Add** button on the screen.

`Name`: Enter the File Format name

`Description`: Enter the format description:

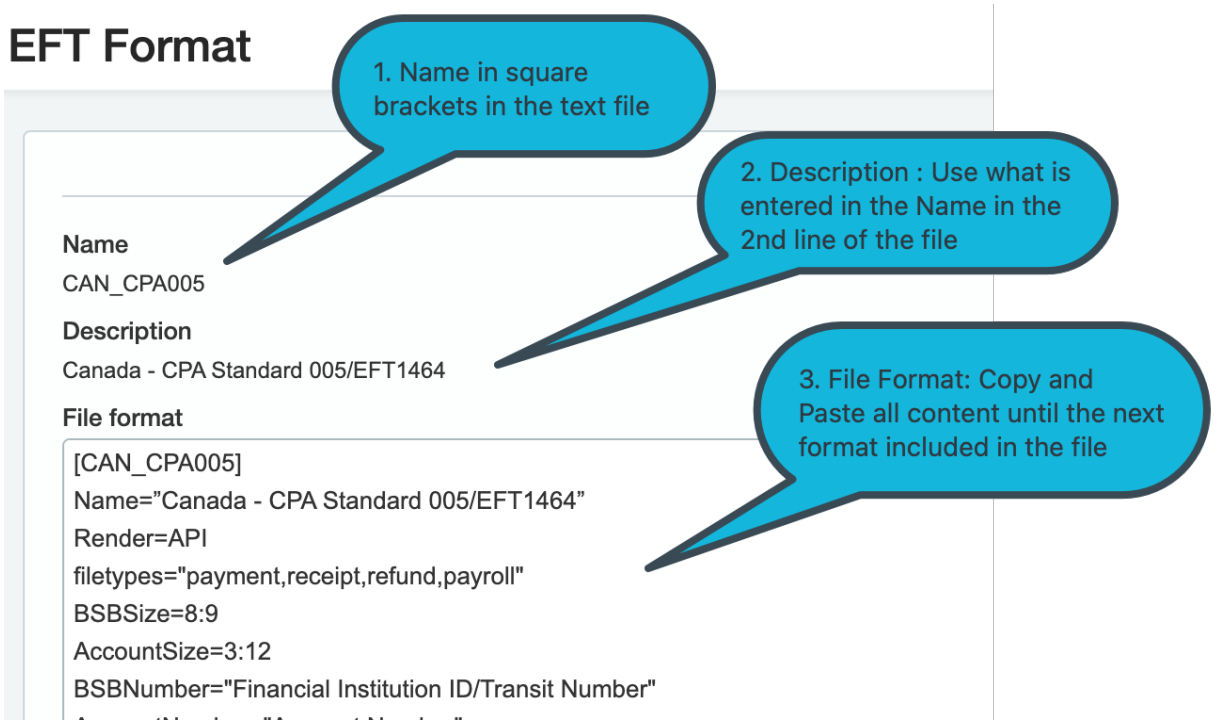
`File Format`: Copy and Paste the exact content from the formats.ini file that you are sent by your Sage partner.

Example:

`Name`: Name between the square brackets in the file, and generally starts with the first 3 letters of the country.

`Description`: Value between the quotes in the Name - 2nd line of the file

`File Format`: Copy and Paste all content until the next format included in the text file (next set of square brackets)



Important! Do not tamper with the File format content unless you have reviewed the bank specifications and understand the detailed file structure.

FILE FORMAT DETAILS

For detailed description of the File Formats, refer to EFT Formats Structure on page 3

Fields that can be included in a file format: Field Names

EFT Formats Technical Info

Watch the Spotlight video on using EFT File Formats in Sage Intacct

EFT FORMATS STRUCTURE

EFT Processing formats include several sections:

[**Format**] - Unique ID of the File format in Square brackets

Name - description of the bank layout

> **Note:** Render=API - Do not change this line

File Format

File format includes several sections:

1. Extra field declaration, including validations, label and help tips

- BSB or sort code name - customised for the bank layout
- BSB or sort code size - customised for the bank layout
- Account name - customised for the bank layout
- Account size - customised for the bank layout
- Up to 10 extra fields on the Bank file
- Up to 50 extra fields on the Vendor file
- Up to 50 extra fields on the Customer file

2. Row Declaration

Each row type can have multiple lines with a number at the end to indicate the sequence. DetailAP1 and DetailAP2 .

- The definition of the header row(s) - general header for all transactions or an A/P specific header or an A/R specific header
- The definition of the detail row(s) - general detail for all transactions or an A/P specific detail or an A/R specific detail
- The definition of the footer row(s) - general footer for all transactions or an A/P specific detail or an A/R specific footer
- Rows can include fields from the Company, Bank, Vendor/Customer, Payment and Applied Invoices records. For details, refer to Field Names

Example:

HeaderAP="A000000001[OriginatorIdNumber:%10C][FileCreationNumber:%04D].."

DetailAP = "C[RecordCount:%09D][OriginatorIdNumber:%-10C]460.."

FooterAP = "Z[RecordCount:%09D][[TransactionTotalInCents:%014D] .."

Validation examples:

BSBSize is the size of the BSB field. This can either be one number, which means the BSB is a fixed size, or two numbers in the format min:max.

AccountSizeis the size of the Account Number field. This can either be one number, which means the Account Number is a fixed size, or two numbers in the format min:max.

Note: The Size is used for data validation.

Making a field required

Use ! in front of a field name. EFT will check if the field has a value.

> **Example:** [!DestAccountNumber] means that the Vendor Bank Account field cannot be blank when creating the EFT File.

More Information

Field Names in EFT File Formats

Row Types below

ROW TYPES

Overview

Within each section, there are 5 types of rows:

HEADER	File Header is row at the beginning of export file.
BATCHHEADER	Batch Header is row for the export file at the beginning of a batch, repeated per batch number.
DETAIL	Detail is row in the export file for eachpayment/receipt/refund being made.
BATCHFOOTER	Batch footer is row in the export file at the end of a batch, repeated per batch number.
FOOTER	File Footer is row at the end of export file.

- The row types are not case sensitive. DETAIL or Detail or detail is interpreted in the same way.
- Each row type is optional.
- Some banks have the same format for AR and AP, some banks have different formats.

Example: For Example, CAN_CPA has the same header and footer format for AR and AP, but different detail formats:

[CAN_CPA]

Name="Canadian CPA"

Render=API

BSBSize=8

AccountSize=3:12

Header="A[RecordCount:%09D][SourceReference:%10C][BatchNumber:%04D]0[BatchDateJulian][SourceBankName:%-5C] [Blank:%-1409C][CR][LF]"

DetailAP = "C[RecordCount:%09D][SourceReference:%-10C][BatchNumber:%04D]460[AmountInCents:%010D]0[EntryDateJulian][DestBSB:%-9D][DestAccount:%-12C]00000000000000000000000000[SourceAccountName:%-15C][DestAccountName:%-30C][SourceAccountName:%-30C][SourceReference:%-10C][DestId:%-19C]000000000 [Reference:%-15C][Blank:%-22C][Blank:%-2C][Blank:%-11C][CRLF]"

DetailAR = "D[RecordCount:%09D][SourceReference:%-10C][BatchNumber:%04D]460[AmountInCents:%010D]0[EntryDateJulian][DestBSB:%-9D][DestAccount:%-12C]00000000000000000000000000[SourceAccountName:%-15C][DestAccountName:%-30C][SourceAccountName:%-30C][SourceReference:%-10C][DestId:%-19C]000000000 [Reference:%-15C][Blank:%-22C][Blank:%-2C][Blank:%-11C][CRLF]"

Footer = "Z[RecordCount:%09D][SourceReference:%-10c][BatchNumber:%04D]00000000000000000000[TransactionTotalInCents:%014D][NumberOfEntries:%08D][Blank:%-01396C][CRLF]"

Row type by transaction type

Each row type can have an optional AR /AP /RF at the end of the row type for a specific format definition for that transaction type. A row type on its own implies a shared format for all transaction types.

EFT first checks for xxAP or xxAR. If it doesn't find it, it uses xx.

(eg HEADERAP or HEADERAR, HEADER) So CAN_CPA has a shared header and footer, but a different DETAIL for AR and AP.

Example: DetailAR is specific for AR Receipt transaction type, DetailRF is for refund transaction and DetailAP is for AP Payment transaction. Detail applies to all transaction types.

Row Sequence

Each row type also can have multiple lines with a number at the end to indicate the sequence.

For example: Header1, Header2 or DetailAR1, DetailAR2, DetailAR3.

Additional Row Types

There are additional row types that can be used for the bank format and work in conjunction with the DETAIL row type to show associated records applied to the original payment entry. To display it in the DETAIL regardless of the row types below, use

DETAIL = "[APINVOICES]"

APINVOICE works for any type (Payment or Receipt).

If the format requires a different definition for A/R Payments, then it can be configured as APINVOICES_ARRECEIPT

APINVOICE	Row for applied AP Documents (Invoice/ CreditNote/ DebitNote/ Prepayment) in payment entry, repeated for each document.
APINVOICES_MISCPAYMENT	Row of detail of AP Miscellaneous Payment. Only applies to Sage 300
APINVOICES_ARRECEIPT	Row for applied AR Document in A/R Payments entry, repeated for each document.
APINVOICES_ARREFUND	Row for applied AR Document in refund entry, repeated for each document. Only applies to Sage 300.

Note: References to Refund or RF is included in some format definition. However, it is not applicable to Sage Intacct.

Example: Following is sample of a bank format using variation of row types.

[AUS_OCH]

Name="Australia Orchid Sample Bank"

BSBSize=0:12

AccountSize=0:20

BSBNumber="Bank Swift Code"

AccountNumber="Bank Account Number"

AccountName="Account Name"

BankAPPaymentReference="Payment Reference"

BankARPaymentReference="Receipt Reference "

CustomerPaymentReference="Customer Receipt Reference "

VendorPaymentReference="Vendor Payment Reference"

APINVOICE="I,[InvoiceNumber:%-12C],[InvoiceDate:DDMMYY],[InvoiceDescription:%-75C],[InvoiceNetAmount:%15D],[CRLF]"

HeaderAP="H,P[CRLF]"

DetailAP1="P,ACH,[BatchNumber:%08D][EntryNumber:%08D],[BankAccountNumber:%-34C],[EntryDate:DD/MM/YYYY], [DestRemitName:%-35C],[DestBSB:%-11C],[DestAccountNumber:%-34C],[Amount:%15D][CRLF]"

DetailAP2="[APINVOICES]"

FooterAP="T,[NumberOfEntries:%05D],[TransactionTotal:%14D]"

FORMAT OF ROW DETAILS

Anything is **[**brackets**]** is a field. Anything else is plain text.

Important! A field can start with **!**indicating it cannot be blank. During the Create EFT file, if the particular field doesn't have value, it will trigger an error.

The Common syntax of a field is: [fieldname:format]

The following format is available for conditional formatting/checking :

[!field]	Indicates e a value is mandatory for the field. During Create EFT file, if the particular field doesn't have value, it will trigger an error.
[#field]	if this field is blank then suppress the whole row and the row count is not increased.
^	In the first position of a row If after including all fields the row is blank, the row will be suppressed and the row count is not increased.
^r	To remove a special character in the field. [Field:^r.-35P]. This means any special character to be replace with '.' 35P means total number of characters after all replacements is 35 printable.

Field Formatting

The Value of a field can be formatted to be displayed in specific structure.

A format mask is made up of multiple sections.

You can have any number of sections.

There are two types of sections:

1. Format

A format section starts with a %

If the section is left justified then the next character in the mask is a -

If the section is padded with 0 then the next character in the mask is a 0

Then comes the size of the field.

Then comes the type of the field:

A	upper case alpha
a	mixed case alpha
N	upper case alpha-numeric
n	mixed case alpha-numeric
D	numeric
C	upper case printable
c	mixed case printable

- Plain text

This is just plain text. For example A - then the letter A would be written to the file

Different masks are available for different field types.

ALPHA NUMERIC FORMAT

To format an Alpha or a Numeric field you can apply different masks.

The following masks are available:

1	Indicator	Always use %
2	Alignment or order of value	If the value to be left justified use - character. R to indicate the size of the value to be evaluated from the right.
3	Zero Padded	For numeric value and value that is lesser than the size, it can be padded with zero.
4	Size	Number of characters to be displayed.
5	Type	Here are the A upper case alpha a mixed case alpha N upper case alpha-numeric n mixed case alpha D numeric C upper case printable c mixed case printable
6	Numeric decimal places and formats	For type D, it can be formatted with one the following : • D - numeric (same as D0) • D0 - numeric (sign, pad, value) only writes sign if it is negative • D1 - numeric (pad, value, sign) always writes sign • D2 - numeric (pad, sign, value) only writes sign if it is negative • For Dx you can add another number to indicate number of decimal places. (e.g. D0.2 indicate 2 decimal places with period as decimal separator) - if you omit the period (e.g. D02 vs D0.2) then no decimal point will be printed - after D you can have a comma (e.g. D,0) if you want a comma instead of a period

FOR DATE FIELDS, THE FORMAT CAN CONSIST OF:

YYYY	The four digit year (eg 2006, 2007)
YY	The two digit year (eg 06, 07)
MM	The one or two digit month (eg 1, 2, 11, 12, etc)
DD	The two digit day (eg 01, 02, 11, 12, etc)
D	The one or two digit day (eg 1, 2, 11, 12, etc)
HH	The two digit 24 hour time
H	The one or two digit 24 hour time
h	The one or two digit 12 hour time
mm	The two digit minute
m	The one or two digit minute

Example: for January 16 2026:

YYYYMMDD would be 20260116

YY/M/D would be invalid since / is not recognized

[Batchdate:YYYYMMDD][Today:YYMMDD][Today:HHmm]

NUMERIC FORMATTING

D - numeric (same as D0)

D0 - numeric (sign, pad, value) only writes sign if it is negative

D1 - numeric (pad, value, sign) always writes sign

D2 - numeric (pad, sign, value) only writes sign if it is negative

- for Dx you can add another number (eg D0.2) which is number of decimal places
- if you omit the period (eg D02 vs D0.2) then no decimal point will be printed
- after D you can have a comma (eg D,0) if you want a comma instead of a period

Example: Sample masks:

%05D	Zero-padded, right-justified, 5 numeric characters
%24N	Right-justified, 24 uppercase alpha-numeric characters
%-35c	Left-justified, 35 mixed case print-able characters
%3D-%3D	Formatted Aus-tralian BSB ie 999-999

VALIDATION FUNCTIONS IN EFT FORMATS

For capital letters only: %-10C would convert to uppercase.

For Swift only, ^S-10C would replace illegal characters with a space, and uppercase the rest.

Note: When using ^, the next character can either be a character to replace illegal characters with, 'r' to remove illegals, 'S' to replace illegal swift characters with a space.

Examples:

Here are some more examples:

"%-10N" + "12345._890" --> NULL

"^ -10N" + "12345._890" --> "12345 890" - replace illegal with space

"^.-10N" + "12345._890" --> "12345..890" - replace illegal with .

"^r-10N" + "12345._890" --> "12345890 " - remove illegals

"^.3D-^.3D" + "123456" --> "123-456" - replace illegal with .

"^.3D-^.3D" + "12A456" --> "12.-456" - replace illegal with .

"^.3D-^.3D" + "12AZZZ" --> "12.-..." - replace illegal with .

"^.3D-^.3D" + "12A4C6" --> "12.-4.6" - replace illegal with .

"^Sc" + "INV[10]" --> "INV 10" - replace illegal swift characters with space

"^Sc" + "INV120111" --> "INV120111" - replace illegal swift characters with space

FIELDS AVAILABLE IN FILE FORMATS

The following fields are available for inclusion in the formats.

Note: References to Refund or RF is included in some format definition. However, it is not applicable to Sage Intacct.

Batch information

This applies to the A/P , A/R and Employee Expenses Payments.

From the payment:

[BatchNumber]	EFT File ID
[BatchDesc]	Batch Description
[BatchCurrency]	Currency code selected in batch header
[BatchDate]	The batch date, or the system date, whichever is later.
[BatchDateJulian]	The batch date as a julian number. Use for most Canadian banks.
[TrueBatchDate]	The real batch date

From the payment detail:

[EntryNumber]	Entry number.
[EntryDate]	Payment date
[EntryDateJulian]	Payment date in Julian Format
[Amount]	Payment amount in bank's currency (eg $100.52 is 100.52)
[AmountInCents]	Payment amount in bank's currency in cents (eg $100.52 is 10052)
[sourceamount]	Payment amount in vendor's/customer's currency
[sourceamountincents]	Payment amount in vendor's/customer's currency in cents

[entry.desc]	Payment Memo (Sage Intacct)
[ereference]	A/P Bill Invoice Reference

Source account information

From EFT Banks:

[BankID]	The Bank Code from the Bank Setup screen.
[BankName]	The Bank Name from the Bank Setup screen.
[BankBSB]	The BSB / Branch / Routing number from the Bank Setup screen.
[BankAccount]	The Account Number from the Bank Setup screen.
[BankAccountName]	The Account Name from the Bank Setup screen.
[BankReference]	EFT Bank – Payment Reference field. (It may have a different label based on the formats).
[BankExtra1]	An extra field which can be turned on for banks. If enabled, you will see a line: "Extra1=abc" in the formats.ini file **abc** will be the field label on the EFT Bank screen.
[BankExtra2] to [BankExtra10]	The same method applies to the other nine extra fields.

From Cash Management > Account Information

[bank.ccy]	The statement currency of the bank code
[bank.address1] [bank.address2] [bank.address3]	Bank Address details

[bank.city]	
[bank.state]	
[bank.zip]	
[bank.country]	

From Cash Management > Account Information

The field names are self-explanatory:

[Company.name]

[Company.taxnumber]

[company.brn]

[Company.address1], [Company.address2], [Company.address3], [Company.address4]

[Company.city]

[Company.state]

[Company.zip]

[Company.country]

[Company.phone]

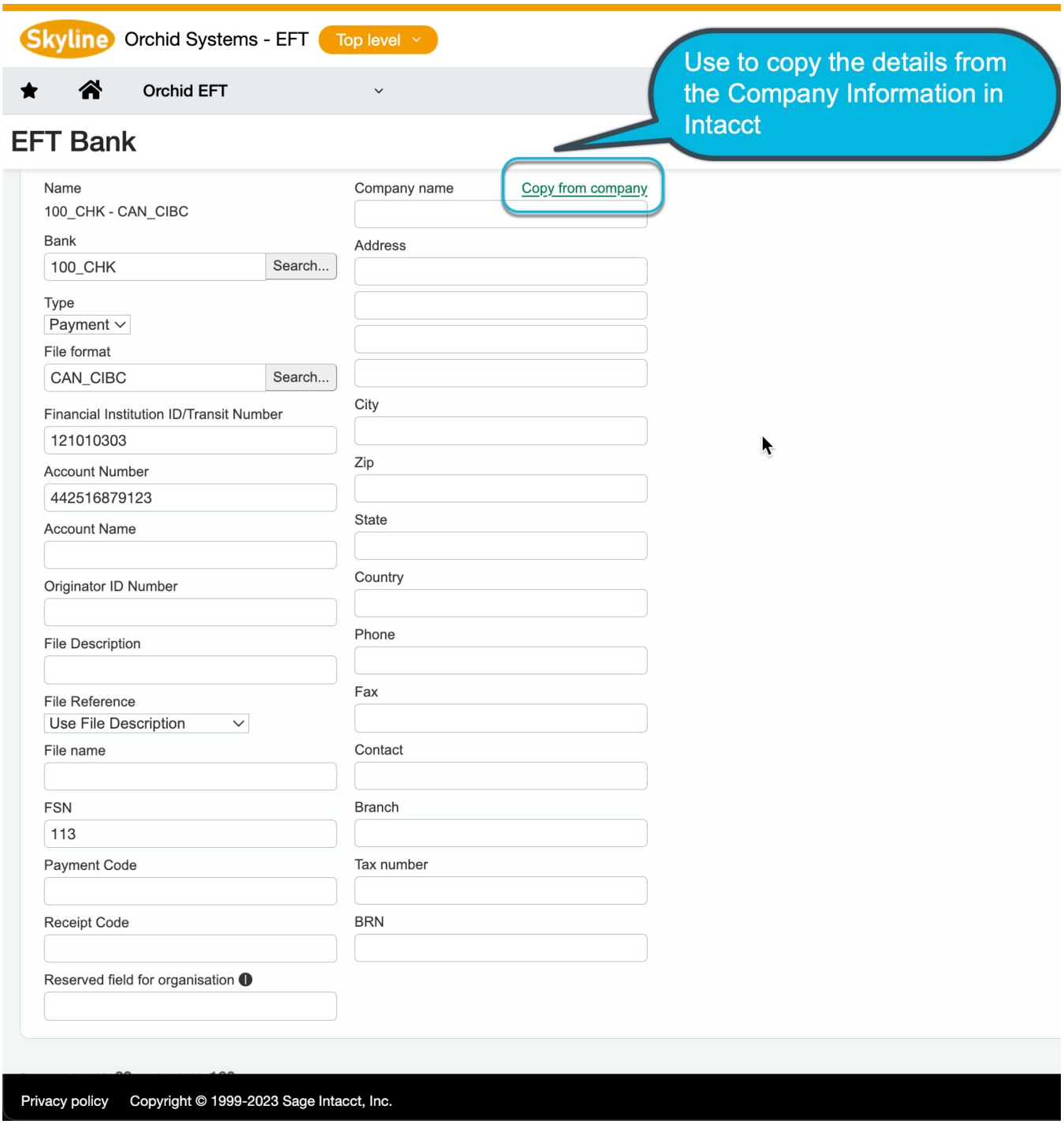
[Company.fax]

[Company.contactname]

[Company.branch]

[Company.taxnumber]

[Company.brn]



Destination account information

From EFT Customers or EFT Vendors:

[DestId]	Vendor or Customer ID field from the EFT Customers or EFT Vendor screen

[DestBankName]	The bank name field from the EFT Customers or EFT Vendor screen
[DestBSB]	The BSB field from the EFT Customers or EFT Vendor screen
[DestAccount]	The account number field from the EFT Customers or EFT Vendor screen
[DestAccountName]	The account name field from the EFT Customers or EFT Vendor screen
[DestIBAN]	IBAN from the EFT Vendor or EFT Customer screen
[DestBIC]	BIC Code from the EFT Vendor or EFT Customer screen
[DestReference]	The reference field from the EFT Customers or EFT Vendor screen
[DestExtra1]	Extra field on EFT Customers or EFT Vendors Setup screen. If enabled, there is a line in the formats.ini: VendorExtra1="abc" or CustomerExtra1="abc" "abc" is the field name to be displayed in the EFT Customer or EFT Vendor setup screen. If writing a custom format, include the lines in the userformats.ini file.
[DestExtra2]to [DestExtra50]	Same as above

Destination account From A/P Vendor or A/R Customer

[DestName]	The A/R Customer Name or A/P Vendor Name
[DestAddress1]to [DestAddress4]	The A/R Customer Address lines 1 to 4 or A/P Vendor Address lines 1 to 4
[DestCity]	The A/R Customer City or A/P Vendor City
[DestState]	The A/R Customer State or A/P Vendor State
[DestPostcode]	The A/R Customer Postcode or A/P Vendor Postcode
[DestCountry]	The A/R Customer Country or A/P Vendor Country
[DestPhone]	The A/R Customer Phone or A/P Vendor Phone
[DestFax]	The A/R Customer Fax or A/P Vendor Fax
[DestEmail]	The A/R Customer E-mail or A/P Vendor E-mail
[DestContact]	The A/R Customer Contact Name or A/P Vendor Contact Name
[DestContactPhone]	The A/R Customer Contact Phone or A/P Vendor Contact Phone
[DestContactFax]	The A/R Customer Contact Fax or A/P Vendor Contact Fax
[DestContactEmail]	The A/R Customer E-Mail or A/P Vendor Contact E-mail
[Dest.ccy]	A/P Vendor or A/R Customer currency code

Applied Document Information:

[InvoiceDate]	Applied Document Date
[InvoiceDateDue]	Applied Document Invoice Due Date
[InvoiceNumber]	Applied Document Number
[InvoiceDescription]	Applied Document Description
[InvoiceGrossAmount]	Applied Document Gross Amount (prior Discount)
[InvoiceDiscountAmount]	Applied Document Discount Amount
[InvoiceNetAmount]	Applied Document Net Amount (after discount / payable amount)
[FirstInvoiceNumber]	The first applied document number

Miscellaneous:

[Today]	The run date
[Reference]	The reference field from Create EFT File screen
[Description]	The description field from Create EFT File screen
[Trans-actionTotalInCents]	The total amount in cents (eg $100.52 is 10052)
[TransactionTotal]	The total amount (eg $100.52 is 100.52)
[NumberOfEntries]	Number of detail lines so far. This is counted from 1.
[RecordCount]	Number of lines so far. This is counted from 1.
[CR]	The A/P Vendor Remit-To Fax
[LF]	A line feed character
[CRLF]	A carriage return / line feed
[Blank]	A blank field. Useful for when there are large runs of spaces or 0. Set the pad length field to the size of the run.
[Today]	The run date
[+xD] or [-xD]	To add or minus number of days to the date field. **Example:** [Today+3D] [BatchDate-1D]
[Reference]	Reference field from the **Create EFT File** screen
[Description]	Description field from the **Create EFT File** screen
[Trans-actionTotalInCents]	The total amount in cents (eg $100.52 is 10052)
[TransactionTotal]	The total amount (eg $100.52 is 100.52)

[NumberOfEntries]	Number of detail lines so far. This is counted from 1.
[RecordCount]	Number of lines so far. This is counted from 1.
[CR]	A carriage return
[LF]	A line feed character
[CRLF]	A carriage return / line feed
[Blank]	A blank field. Useful for when there are large runs of spaces or 0. Set the pad length field to the size of the run.
[BCRLF]	A carriage return / line feed will only be inserted if there is already a record /value in the file. For formats without a footer row and where the last record must not have a carriage return, use BCRLF at the beginning of each detail row instead of placing CRLF at the end of the row.
[FieldIDModifier]	Alphabet from A to Z. To indicate a particular batch has been run for Create EFT file process for the number of time. The first run will have 'A' as indicator, subsequent run 'B', 'C' and so on.
[InitRecord1Count].. [InitRecord5Count]	To reset record counter to zero. There are 5 record counters available.
[IncRecord1Count]. [IncRecord5Count]	To include a record count. When a [IncRecordxCount] field is placed on a row, it will increment by one.
[GetRecord1Count].. [GetRecord5Count]	To print the record count number.
[InitAmount1] .. [InitA-mount5]	To reset amount summary field to zero. There are 5 amount summary fields available.
[AddAmount1] to [AddA-mount5]	Use to add the amount to a summary field. In order to do the summary, the [Amountx]field has to be defined. **Example:** Amount1="[BankAmount]" DetailAP="[AddAmount1],[BankAccountNumber],[BatchCurrency], [Amount],[EntryDate:YYYYMMDD],[DestBankName:%-35C], [DestAc-countNumber], [DestAccountName:%-35C] [CRLF]"
[GetAmount1] to [GetA-mount5]	To display or print the Amount summary. Eg: **Example:** Footer="99,[NumberofEntries],[GetAmount1:%D0.2][CRLF]"
[AddAmountY=X]	To add amount of a "Y" field to the summary of "X".

Eg: [AddAmount2=1]	Eg. A requirement to add Payor / Account Number with the beneficiary Account Number and display it at the footer as the check digit for the transfer file. In the format, declare field Y as an amount field, add it to the summary and display at the footer. Amount1="[DestAccount:%010D]" Amount2="[BankAccountNumber:%010D]" Detail1="[AddAmount1][AddAmount2=1]01[AmountInCents:%012D][BankAccountNumber:%010D][DestAccount:%010D][CRLF]" Footer1="99[TransactionTotalInCents:%018D][GetAmount1:%R10C][CRLF]"
[EffectiveDate]	EffectiveDate entered on the EFT File List.
[FILENAME]	The actual generated EFT File name

FIELD LABELS ON SETUP SCREENS AND HELPFUL HINTS

Field labels can be renamed in the formats.ini or the userformats.ini file. See "EFT Banks Form - Overview" on page 1

Field labels on setup screens

On the formats.ini, the following standard fields can be re-labelled on the screen to use terminology that is familiar to the users. The standard field names are in bold, and the new label is given in quotes.

BSBNumber="Financial Institution ID/Transit Number"

AccountNumber="Account Number"

AccountName="Account Name"

BankAPPaymentReference="Payment ID"

BankARPaymentReference="Receipt ID"

VendorPaymentReference="Payment Reference"

CustomerPaymentReference="Receipt Reference"

USER DEFINED FIELDS AND FIELD TIPS

User Defined Fields can be configured to include additional information in the file based on the Bank format specification. Fields are activated in the formats.ini / userformats.ini files.

Tip: Tips can be added to provide information about the field on the Sage 300 EFT Setup screens where the field values are updated.

For user defined fields, there are up to 9 extra fields available in the Bank Setup and up to 50 extra fields available in the EFT Customers Setup or EFT Vendors Setup.

To activate the display on setup screen, declare the field and give it a name.

The syntax for extra field is: **UserDefinedFieldName = "[fieldname:format]"**

Eg.

BankExtra1="UserName"

BankExtra2="Password"

VendorExtra1="AccountType (1=Checking, 2=Savings, 3=Transmission)"

CustomerExtra1="AccountType (1=Checking, 2=Savings, 3=Transmission)"

To give a field extra information, we can use a tip.

When a tip is declared on the formats, on screen it will show as hyperlink.

Example: VendorExtra1="Payment"

VendorExtra1Tip="For USA : CCD=Corporate, PPD=Personal, leave blank for CAN"

BankExtra1="Account Qualifier"

BankExtra1Tip="02 - US Settlement Account domiciled in US\r\n04 - Canadian Settlement Account domiciled in Canada"

BankExtra2="Payment Method Code"

BankExtra2Tip="This can be:\r\n\t01 = Cheque\r\n\t02 = EFT/ACH\r\n\t03 = EDI\r\n\t04 = ERA (Email or Fax Advice)"

Note: On the Tip there are few predefined keys:

\r\n To insert a new row for the remaining text after this key.

\t To insert a tab for the remaining text after this key.

Example: For example, in a situation where a beneficiary name is allowed for up to 35 characters (on entry screen is up to 60 characters) and no trailing spaces allowed if it is less than 35 characters, a user defined field can be created to retrieve the 35 characters of beneficiary name and then used in the row with left aligned and exact length format.

AccName="[DestAccountName:%-35C]"

Header1=""[BankExtra1:%-C]","[BankExtra2:%-C]","[BankExtra3:%-C]","D","1","1"[CRLF]"

Header2=""Batch[BatchNumber:%06D]","[BatchDate:YYYY/MM/DD]"[CRLF]"

Detail1=""[DestID:%-C]","[AccName:%-C]","[DestExtra1:%-C]","[DestBSB:%06D]", "[DestAccount:%011D]","[AmountInCents:%-C]"[CRLF]"

CUSTOM FUNCTIONS

Special Fields

The following special fields can be included in a format.

[CPA005MultiBank] Displays the account number of the source bank from the selected batch range. If the same bank is used in multiple batches, it will only appear once. Use in the header of the CPA005 format as the settlement account.

[CPA005MultiBankIndex] Displays the Index of the bank of the source bank from the selected range of batch. This information to show on detail line and to indicate which settlement account to be used for the payment.

Switch options

The following codes provide additional options to control how EFT Processing creates the file.

`OneFilePerEntry=1` Use this switch to indicate that each entry must go to an individual file.

`{entry}Entry Number`: Use in conjunction with OneFilePerEntry=1 option.

`IsEFT=0` This switch indicates to skip checking whether the vendors/customers are setup in EFT Processing. This is mainly used for formats that deal with Miscellaneous Receipt/Payment entry that don't have Vendor/Customer code.

`UTF8=1`: Use to indicate characters have to be printed in UTF8.

`AllowMiscellaneousPayments=1`: Use this switch to indicate that a vendor with no record in EFT vendor is allowed in the EFT file (mainly for positive pay)

`AllowMiscellaneousRefunds=1` : This switch indicates that a customer that has no record in EFT customer is allowed in the EFT file (mainly for positive pay)

`AllowMiscellaneousReceipts=1` : This switch indicates that a customer that has no record in EFT customer is allowed in the EFT file.

File naming

Custom field for file naming in EFT Bank Setup:

Eval fields

There are two functions (Eval and EvalAtEnd) to manipulate the fields read from the database.

`EvalAtEnd` is done after all the detail records are written.

Within `Eval` and `EvalAtEnd`, there are several functions available:

Available functions:

`Trim(str)` - trims trailing spaces

`Join(str, ...)` - joins multiple strings together

`Left(str, n)` - returns the left-most n characters of a string

`Right(str, n)` - returns the right-most n characters of a string

`Space(n)` - returns a string of n spaces

`Space(n, char)` - same as Space but you can specify the character, eg Space(5, "0")

`AsciiAdd(str)` - adds the ascii value of each character and returns the sum

`NumberAdd(str)` - adds the numeric value of each character and returns the sum

`AddAmount(n, amt)` - adds amt to Amountn, returns new amount

`GetAmount(n)` - returns Amountn

`GetAmount(n, prec)` - returns Amountn, formatted with prec decimal places

`SetAmount(n, amt)` - set Amountn to amt, returns amt

`GetField(str)` - get a field, eg "BankAccountName"

`Mul(x,y)` – multiplication of x and y

`Add(x,y)` – Add x and y. The add function adds 2 absolute numbers.

To add negative amount or subtraction, use Lua. See Using Lua in Custom Functions on page 26

`Div(x,y)` - Divide x with y, DIV(6,2) -> 3 - This function no longer works as part of Eval. Use a LUA function to divide. Using Lua in Custom Functions on page 26

`Replace(str,findStr,replaceWith)` - eg:Replace("hello world", "world", "Stacy") --> "hello Stacy"

`ReplaceAccents` – to replace accent character to latin character.

Example:
DExtra4="[Eval(ReplaceAccents(getfield("DestExtra4"))] "

If the value of DestExtra4 is MARCHé, DExtra4 = MARCHe

`Replace(strFind, strValue)` – eg. as below :

DetailAP1="[InitRecord3Count]622[DestBSB:%9D][DestAccount:%17C] {TOTALRECORDCOUNT} [CRLF]"

DetailAP2-----23

DetailAP24="Eval(Replace("{TOTALRECORDCOUNT}",GetField("AddCount")))][CRLF]"

Including a large number of applied invoices in the file

APINVOICES has a limit of 1000 invoices.

If you want to include applied invoices in the file and the number of invoices applied in the payment entry is more than 1000, use EVAL to bypass the limit.

`EVAL(Documents(9999,APINVOICE"))`, where 9999 is maximum number of invoices to be included.

Calculating hash code

To calculate the hash code you could use the following to add the sum into Amount1. The final "" makes it output nothing.:

Detail1="[Eval(

AddAmount(1,AsciiAdd(Trim(Left(GetField("BankAccountName"), 70))));

AddAmount(1,AsciiAdd(Trim(Left(GetField("BankAccountName"), 70))));

AddAmount(1,NumberAdd(Trim(GetField("BankAccountNumber"))));

AddAmount(1,AsciiAdd(Trim(Left(GetField("BatchCurrency"), 70))));

AddAmount(1,NumberAdd(GetField("Amount")));

"")]"

Note: It would all be one line - separated into multiple lines for readability.

To add the hash into the file you could use:

Detail2="[EvalAtEnd(SetAmount(2,0);AddAmount(2,GetAmount(1));AddAmount(2,GetField("Num-berOfEntries"));GetAmount(2, 0))]"

That adds Amount1 and NumberOfEntries together, then prints it with no decimal places.

Special hash code calculations

`UOBFieldCheckSummary(hashIndex, value)`

Example: Eval(UOBFieldCheckSummary(18, "000000000000120000"))] produces "8249".

`UOBSetPaymentCode(paymentType)`. paymentType can be 'P', 'R', or 'C'.

This does the same as the first if. It sets amount8 to 20/22/30 (eg 'Eval(GetAmount(8))')

There's also `UOBSetHashCode()`, which sets amount 9 (eg 'Eval(GetAmount(9))'), and does the same as the second if.

Encryption formulas

`MD5` – returns encryption in MD5 calculation – version 62 onwards

`sha256`- encryption in SHA256 calculation - Version 63 onwards

Examples:
[Eval(toBase64(MD5(Trim(GetField("DestExtra1")))))]

[Eval(sha256(Trim(GetField("PayLine"))))]

Nachanines

This is used as the last field in the footer section of the file when a file format needs to be in block of 10 lines.

[NachaNines] is adding lines filled with the character "9" at the end of the file to keep the file in the block of 10 lines.

For example, If the total number of lines in the file is 8 lines,[NachaNines] will insert 2 lines of 94 bytes of '9' characters.

Example:

This file has 5 lines, the bank requires 10 lines.

NachaNines field adds 5 lines of '9'

FooterAP3="9000001[ACHBlockCount10Div10:%06D][GetRecordCount:%08D][HashTotal:%R010D][TransactionTotalInCents:%012D][TransactionTotalInCents:%012D] [CRLF][NachaNines]"

101 026013576113064019 1904151152B094101BANK FOOD INC

5220FOOD INC 1113064019CCD 190415200101 1026013570000001

622987165681210987654348 00000113007200 LEON'S CATERING 0026013570000002

8220000001009871656800000000000000000113001113064019 026013570000001

9000001000001000000010098716568000000000000000000011300

9999999999999999999999999999999999999999999999999999999999999999999999999999999-999999999999

9999999999999999999999999999999999999999999999999999999999999999999999999999999-999999999999

9999999999999999999999999999999999999999999999999999999999999999999999999999999-999999999999

9999999999999999999999999999999999999999999999999999999999999999999999999999999-999999999999

9999999999999999999999999999999999999999999999999999999999999999999999999999999-999999999999

LUA Functions

In the latest product updates in version 2019 and above, you can use LUA functions if the format uses RENDER=API.

Refer to Using Lua in Custom Functions below

USING LUA IN CUSTOM FUNCTIONS

Lua Functions

In the latest product updates in version 2019 and above, you can use Lua functions if the format uses RENDER=API.

Tip: Refer to Lua documentation for tips to get started with Lua scripting.

To use a Lua function in a format, the formats needs the line 'RENDER=API'.

Add a section for the Lua function.

The section for the Lua function starts with >>> lua and ends with <<< lua

Example:
The example below checks if Ref is blank then the function returns first applied Invoice Number.

LUA function in EFT Formats.ini

```
>>> lua
function GetRef(Ref)
    if Ref == "" then return GetField("FirstInvoiceNumber")
    else return(Ref)
    end
end
<<< lua
```

Using the function

You can then create a variable using the function

Example:
DRef is the variable where you call the function (GetRef) and pass the parameter, in this case the DestReference field.

DRef="[Eval(GetRef(Trim(GetField("DestReference"))))] "

Add the variable in the format definition

Lastly, use the variable on the line where you want it to be displayed.

EFT FORMAT EXAMPLE

The format below includes examples of multiple row types, different row types for different transaction types as well as various usage of the field formatting masks.

Example: Format Example : for the USA JP Morgan Chase bank

[USA_JPMC]

Name="JP Morgan Chase"

BSBSize=9

AccountSize=9:17

BSBNumber="R/T Number"

BSBNumberTip="Enter the 9 Digit R/T Number"

AccountNumber="Account Number"

AccountName="Account Name"

BankExtra1="Application ID"

BankExtra2="Remote ID (RID)"

BankExtra3="File ID"

BankExtra4="Company ID"

BankAPPaymentReference="A/P Payment Reference"

BankARPaymentReference="A/R Receipt Reference"

CustomerPaymentReference="Payment Reference"

CustomerPaymentReferenceTip="Enter the Payment Reference to be sent to the customer"

VendorPaymentReference="Payment Reference"

VendorPaymentReferenceTip="Enter the Payment Reference to be sent to the vendor"

HeaderAR1="LOGDX2010270388TB1TX00317*CH04000000000CTXC3173C0000000000000000[CRLF]"

HeaderAR2="101 [SourceBSB:%9D][SourceExtra3:%010D][Today:YYMMDD]
[Today:HHMM]1094101JPMORGAN CHASE [SourceAccountName:%-23C] [CRLF]"

HeaderAR3="5225[SourceAccountName:%-16C] [SourceExtra4:%-10C]PPD[Description:%-10C]
[BatchDate:YYMMDD] 1[SourceBSB:%8D]0000001[CRLF]"

HeaderAP1="LOGDX2010270388TB1TX00317*CH04000000000CTXC3173C0000000000000000[CRLF]"

HeaderAP2="101 [SourceBSB:%9D][SourceExtra3:%010D][Today:YYMMDD]
[Today:HHMM]1094101JPMORGAN CHASE [SourceAccountName:%-23C] [CRLF]"

HeaderAP3="5200[SourceAccountName:%-16C] [SourceExtra4:%-10C]PPD[Description:%-10C]
[BatchDate:YYMMDD] 1[SourceBSB:%8D]0000001[CRLF]"

Detail="6[DestReference:%2D][DestBSB:%9D][DestAccount:%-17C][AmountInCents:%010D][DestId:%-15C][DestAccountName:%-22C] 0[SourceBSB:%8D][BatchNumber:%07D][CRLF]"

FooterAR1="8225[NumberOfEntries:%06D][WellsFargoHashTotal:%010D]000000000000[Trans-actionTotalInCents:%012D][SourceExtra4:%-10C] [SourceBSB:%8D]0000001[CRLF]"

FooterAR2="9000001000001[NumberOfEntries:%08D][WellsFargoHashTotal:%010D]000000000000[TransactionTotalInCents:%012D] [CRLF]"

FooterAP1="8200[NumberOfEntries:%06D][WellsFargoHashTotal:%010D]000000000000[TransactionTotalInCents:%012D][SourceExtra4:%-10C] [SourceBSB:%8D]0000001[CRLF]"

FooterAP2="9000001000001[NumberOfEntries:%08D][WellsFargoHashTotal:%010D]000000000000[TransactionTotalInCents:%012D] [CRLF]"

This page intentionally left blank.