EFT Processing
For Sage 300
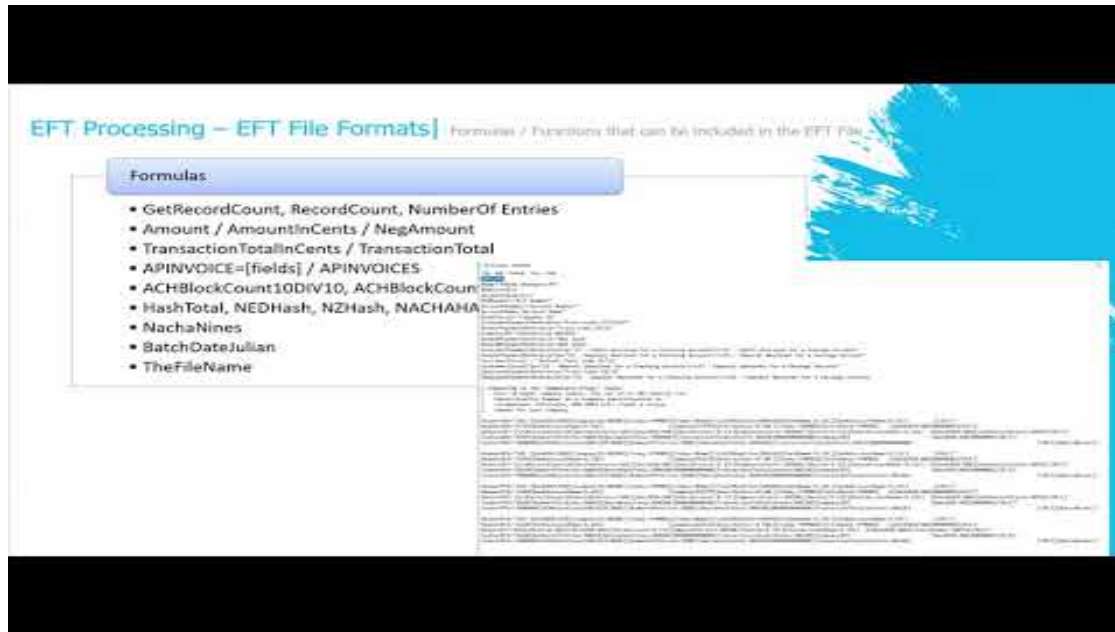
Orchid
Getting Started Guide:

Work Smarter with Orchid

This page intentionally left blank.

EFT Formats Technical Info

Watch Spotlight video on EFT Processing File Formats.

BANK FORMATS - OVERVIEW

EFT Processing installs with a number of system defined file formats. These formats are contained in a file **formats.ini** which is found in the ..\Sage\ELxxA directory.

Additional bank formats are added on demand by the Orchid support team for supported versions of EFT Processing.

Amending File Formats

The formats.ini file should never be changed on site. In the latest product updates, the formats.ini file includes a hash calculation to ensure the file is not tampered with. If the file is amended in anyway, it cannot be used in Create EFT File and you will receive an error message.

To change an existing bank format (or create a new one), copy the original bank format from the formats.ini file to the userformats.ini file and make the required changes.

When upgrading

If upgrading from a prior version with customisations done in the formats.ini file, you need to copy the changes to the formats.ini to the userformats.ini. Refer to details in

Sections in the ini file

The formats.ini file include multiple sections.

> Refer to EFT Processing online help for more details.
>
> EFT Help: https://help.orchid.systems/sage300addons/Content/Product-Home/EFT-Processing.htm

Each bank is a section in the ini file. A section is denoted by Square Brackets for example :

[USA_WF] for the Wells Fargo bank layout.

Within each section, you can define the following

[**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

> **Example:**
>
> HeaderAP="A000000001[OriginatorIdNumber:%10C][FileCreationNumber:%04D].."
>
> DetailAP = "C[RecordCount:%09D][OriginatorIdNumber:%-10C]460.."
>
> FooterAP = "Z[RecordCount:%09D][[TransactionTotalInCents:%014D] .."

Rows can include fields from

> **Example:** Bank Setup Validation and renaming.
>
> **[USA_WF]**
>
> **Name=**"USA Wells Fargo ACH"
>
> **BSBSize=**9
>
> **AccountSize=**9:17
>
> **BSBNumber=**"R/T Number"
>
> **AccountNumber=**"Account Number"
>
> **AccountName=**"Account Name"
>
> **SourceExtra1=**"Application ID"
>
> **SourceExtra2=**"Remote ID (RID)"
>
> **SourceExtra3=**"File ID"
>
> **SourceExtra4=**"Company ID"
>
> **Header=**
>
> **BatchHeader=**
>
> **DetailAR=**
>
> **DetailAP=**
>
> **BatchFooter=**
>
> **Footer=**

Validation examples:

Example: Bank Setup Validation and renaming.	
BSBSize=9	BSB Number will be validated to 9 in length
AccountSize=9:17	Account Number will be validated to be between 9 and 17 in length
BSBNumber="R/T Number"	BSB Number will be re-labeled R/T Number
AccountNumber="Account Number"	Account Number will be Account Number
AccountName="Account Name"	Account Name will be labeled Account Name
SourceExtra1="Application ID"	one extra field on the Bank will be enabled called "Application ID"
SourceExtra2="Remote ID (RID)"	second extra field on the Bank will be enabled called "Remote ID (RID)
SourceExtra3="File ID"	Third extra field on the Bank will be enabled called "File ID"
SourceExtra4="Company ID"	Fourth extra field on the Bank will be enabled called "Company ID"

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.

AccountSize is 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.

RENDER=API

Since the latest product update for Sage 300 version 2019, EFT Processing includes two dll engines to generate the EFT Files:

- The original dll ELEFT.dll works only with Sage 300. It includes support for optional fields on AP Payments and AR Receipts headers. However, it doesn't support LUA functions for conditional formulas in the file. While this dll is supported, enhancements are not added to it.
- EFT.dll works with various ERP Applications. It includes standard predefined fields and includes support for LUA statements to write conditional statements in the file format. Any enhancements to EFT Processing are done in the EFT.dll only. To use the newer dll, the file format must include the line RENDER=API.

> **Note:** If the RENDER=API is deleted or omitted, EFT Processing will use the ELEFT.dll.

Tips when converting from an older format to render=API

With the latest dll, you can't include the Sage 300 table names directly into the formats.ini or user-formats.ini. You need to use the pre-defined field names.

For example, you need replace the fields below:

Fields [CSCOM.*] in Company Profile, change to [Company.*]

Field [ELCUS.EMAIL] for email address, change to [DestAdviceEmail].

Field [APTCR.IDRMIT] for cheque number, change to [FirstCheque]

Field [APTCR.DOCNBR] for Document Number, change to [DocNum]

Field [APTCR.CODECURN] for Vendor Currency Code, change to [Dest.ccy]

USER FORMATS - UPDATING EFT BANK FORMATS

EFT Processing installs with a number of system defined file formats. These formats are contained in a file formats.ini which is found in the ..\Sage\ELxxA directory. The formats.ini file cannot be changed on site.

When processing a file layout, EFT Create File first checks the userformats.ini file for the bank selected. If found, it uses the layout from the userformats.ini. If not found, it uses the format from the system formats.ini file.

To change an existing bank format (or create a new one), copy the original bank format from the formats.ini file to the userformats.ini file and make the required changes.

> **Note:**
>
> The userformats.ini can be copied to the **customization directory** to use standard Sage 300 functionality of database and user control.
>
> When processing a file layout, EFT Create File first checks the customization directory for userformats.ini. If not found, it checks the ELXXA (e.g. EL68A for version 2021) folder of the Sage 300 install for userformats.ini and if not found it uses formats.ini in the ELXXA folder.

For a list of row types, field names and validations, refer to Bank Formats - Overview on page 3

> **Note:** If you change the formats.ini file, you will get a "File Hash" Error when using the file in Create EFT File function.

ROW TYPES

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 at export file at the beginning of a batch, repeated per batch number.
DETAIL	Detail is row at export file for each payment/receipt/refund being made.
BATCHFOOTER	Batch footer is row at 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"

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[Amoun-
tInCents:%010D]0[EntryDateJulian][DestBSB:%-9D][DestAccount:%-
12C]000000000000000000000000[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[Amoun-
tInCents:%010D]0[EntryDateJulian][DestBSB:%-9D][DestAccount:%-
12C]000000000000000000000000[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][BatchNum-
ber:%04D]000000000000000000000[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 trans-

action 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 4 additional row types that could be used for the bank format and work in conjunction with 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	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.
APINVOICES_ARRECEIPT	Row for applied AR Document in receipt entry, repeated for each document.
APINVOICES_ARREFUND	Row for applied AR Document in refund entry, repeated for each document.

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.
	To remove a special character in the field.
^r	[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. (eg D0.2 indicate 2 decimal places with period as decimal separator)
		  - 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

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 Australian 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

FIELD NAMES

The following fields are available for inclusion in the formats.

From the batch header:

BatchNumber	The batch number
BatchDate	The batch date, or the system date, whichever is later.
BatchDateJulian	The batch date as a julian number. For most Canadian banks
TrueBatchDate	The real batch date

From the batch entry:

EntryNumber	The entry number.
EntryDate	The entry date
EntryDateJulian	The entry date in Julian Format
Amount	The amount (eg $100.52 is 100.52)
AmountInCents	he amount in cents (eg $100.52 is 10052)

Source account information (from EFT Banks):

SourceBank	The Bank Code from the Bank Setup screen.
SourceBankName	The Bank Name from the Bank Setup screen.
SourceBST	The BSB number from the Bank Setup screen.
SourceAccount	The Account Number from the Bank Setup screen.
SourceAccountName	The Account Name from the Bank Setup screen.
SourceReference	Either the A/P Payment Reference or the A/R Receipt Reference from the Bank Setup screen depending if an AR or AP batch is being exported
SourceExtra1	An extra field which can be turned on for banks. Turn on by putting "Extra1=abc" in the formats.ini file
SourceExtra2	An extra field which can be turned on for banks. Turn on by putting "Extra1=abc" in the formats.ini file
SourceExtra3	An extra field which can be turned on for banks. Turn on by putting "Extra1=abc" in

| | the formats.ini file |
| **SourceExtra4** | An extra field which can be turned on for banks. Turn on by putting "Extra1=abc" in the ini file |

Destination account information (from EFT Customers or EFT Vendors):

DestId	Vendor or Customer ID field from the EFT Customers or EFT Vendor screen
DestShortName	The A/P Vendor short name (APVEN.SHORTNAME) or A/R Customer short name (ARCUS.TEXTSNAM)
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
DestReference	The reference field from the EFT Customers or EFT Vendor screen
DestExtra1..50	Extra field on EFT Customers or EFT Vendors Setup screen. Extra field can be turn on in formats.ini by putting : VendorExtra1="abc" or CustomerExtra1="abc". "abc" is the field name to be displayed in the EFT Customer or EFT Vendor setup screen. From version 56, it can be up to 50 customised extra fields.
DestName	The A/R Customer Name or A/P Vendor Name
DestAddress1..4	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

Destination account information (from Vendor Remit To on the batch entry):

DestRemitName	The A/P Vendor Remit-To Name
DestRemitAddress1..4	The A/P Vendor Remit-To Address lines 1 to 4
DestRemitCity	The A/P Vendor Remit-To City
DestRemitState	The A/P Vendor Remit-To State
DestRemitPostcode	The A/P Vendor Remit-To Postcode
DestRemitCountry	The A/P Vendor Remit-To Country
DestRemitPhone	The A/P Vendor Remit-To Phone
DestRemitFax	The A/P Vendor Remit-To Fax
DestRemitEmail	The A/P Vendor Remit-To E-mail
DestRemitContact	The A/P Vendor Remit-To Contact Name
DestRemitContactPhone	The A/P Vendor Remit-To Contact's phone
DestRemitContactFax	The A/P Vendor Remit-To Contact's fax
DestRemitContactEmail	The A/P Vendor Remit-To Contact's e-mail

Applied Document Information:

InvoiceDate	Applied Document 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	TThe 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
TransactionTotalInCents	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. Eg. [Today+3D] [BatchDate-1D]
Reference	The reference field from Create EFT File screen
Description	The description field from Create EFT File screen
TransactionTotalInCents	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. The use of this BCRLF specific for formats without footer row and with requirement that the last record should not have carriage return. Instead of placing CRLF at the end of detail row format, use BCRLF at the starts of detail 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 .. InitRe-	To reset record counter to zero. There are

cord5Count	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 .. InitAmount5	To reset amount summary field to zero. There are 5 amount summary fields available.
AddAmount1 .. AddAmount5	To add the amount to summary field. In order to do the summary, the Amountx field has to be defined. Eg : Amount1="[SourceAmount]" DetailAP="[AddAmount1],[BankAccountNumber],[BatchCurrency], [Amount],[EntryDate:YYYYMMDD], [DestBankName:%-35C], [DestAccountNumber], [DestAccountName:%-35C][CRLF]"
GetAmount1 .. GetAmount5	To display or print the Amount summary. Eg: Footer="99,[NumberofEntries], [GetAmount1:%D0.2][CRLF]"
[AddAmountY=X] Eg: [AddAmount2=1]	To add amount of a "Y" field to the summary of "X". 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. On the format, it has to declare to 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	This date field can be used as active pay-

	ment date. When the EffectiveDate field is used in the format, it will display a date field on Create EFT screen.
SessionDate	Sage 300 sign on date.
FILENAME	The actual generated EFT File name

Fields from Other tables

Fields from certain tables can also be used in the formats.ini.

This is only available if you are using the older EFT.dll (where the format doesn't include render=api)

The layout is [tablename.fieldname].

> **Example:** To use the company name from the Common Services file CSCOM you would use [CSCOM.CONAME]

Below are tables that can be used in the formats.ini:

> CSCOM
>
> BKACCT
>
> APBTA
>
> APTCR
>
> APTCP
>
> APVEN
>
> APOBL (Invoice line only)
>
> ARTCR
>
> ARTCP
>
> ARBTA
>
> ARCUS
>
> ARRFH

AROBL (Invoice Line only)

ELBANK

ELCUST

ELVEN

if using EFT Payroll, you can also use:

CPEMPL or UPEMPL,

CHK (for CPCHKH or UPCHKH)

ELEMPH

ELEMPD

Optional Field

(from Sage 300 version 6.0 onwards) and if the format doesn't include render=api

The layout is [tablename.optionalfield].

> **Example:** [ARCUSO.PRIORITYCODE:%-5C].

ARCUSO - AR Customer Optional Field Values

ARTCRO - AR Receipt/Adjustment Optional Field Values

ARRFHO - AR Refund Optional Field Values

APVENO - A/P Vendor Optional Field Values

APTCRO - A/P Payment/Adjustment Optional Field values

> **Note:** You can't include AP or AR Invoice optional fields in a format.

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 32

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 FIELD

CPA005MultiBankDisplay account number of the source bank from the selected range of batch. Same bank use in multiple batch will appear only once in this function. The information to show on the Header file of CPA005 format as the settlement account.

CPA005MultiBankIndexDisplay 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.

OneFilePerEntry=1This switch to indicate for each entry to go to an individual file.

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

`UTF8=1` : This to indicate characters 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.

Custom field for Fixed file naming in EFT Bank Setup:

```
Fixed File Name:
{dd} or {DD} day number, eg '02'
{mm} or {MM} month number, eg '02'
{yy} or {YY} year, eg '02'
{yyyy} or {YYYY} year, eg '2011'

{N} run number eg '2'
{n} 1 digit run number eg '2'
{nn} 2 digit run number eg '02'
{nnn} 3 digit run number eg '002'
{nnnn} 4 digit run number eg '0002'

{B} batch number eg '2'
{b} 1 digit batch number eg '2'
{bb} 2 digit batch number eg '02'
{bbb} 3 digit batch number eg '002'
{bbbb} 4 digit batch number eg '0002'

{S} file sequence number eg '2'
{s} 1 digit file sequence number eg '2'
{ss} 2 digit file sequence number eg '02'
{sss} 3 digit file sequence number eg
'002'
{ssss} 4 digit file sequence number eg
'0002'
```

{entry}Entry Number, this to be use with the option that each entry to go to individual file

Added on version 6.0 onwards 4th Feb 2016.

(Check SIN_MIZUHO_CSV or SIN_MIZUHO_TT format for the usage).

Eval field

There are two new fields - Eval and EvalAtEnd. EvalAtEnd is done after all the detail records are written.

Within Eval* there are several functions available:

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

Div(x,y) - Divide x with y, DIV(6,2) -> 3

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

ReplaceAccents – to replace accent character to latin character.

Eg. DExtra4="[Eval(ReplaceAccents(getfield("DestExtra4"))] "

[Eval(toBase64(MD5(Trim(GetField("DestExtra1")))))] – returns encryption in MD5 calculation – version 62 onwards

[Eval(sha256(Trim(GetField("PayLine"))))] – returns encryption in SHA256 calculation - Version 63 onwards

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

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

DetailAP2-----23

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

For example, to calculate the hash you could use:

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.

That basically adds the sum into Amount1. The final "" makes it output nothing.

To add the hash into the file you could use:

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

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

UOBFieldCheckSummary(hashIndex, value), eg:

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.

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

6229871656812109876543 48 00000113007200 LEON'S CATERING 0026013570000002

822000000100987165680000000000000000000113001113064019 026013570000001

900000100000100000001009871656800000000000000000000011300

9999999999999999999999999999999999999999999999999999999999999999999999999-999999999999

9999999999999999999999999999999999999999999999999999999999999999999999999-999999999999

9999999999999999999999999999999999999999999999999999999999999999999999999-999999999999

9999999999999999999999999999999999999999999999999999999999999999999999999-999999999999

9999999999999999999999999999999999999999999999999999999999999999999999999-999999999999

LUA Functions

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

To create a LUA function, you need the following:

- So the first thing, it needs the line 'RENDER=API'.

- Then section for the LUA function.

Example:

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

LUA function in EFT Formats.ini

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

- You can then create a variable using the function

Example:

DRef is the variable where you call the function and passing the parameter, in this case DestReference.

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

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

```
[ANZ_IB4B]
Name="ANZ_IB4B Custom Payment Reference"
Render=API
filetypes="payment,receipt,refund"
BSBSize=6
AccountSize=5:9
BSBNumber="BSB Number"
AccountNumber="Account Number"
AccountName="Account Name"
BankAPPaymentReference="User ID"
BankARPaymentReference="User ID"
CustomerPaymentReference="Payment Reference"
VendorPaymentReference="Payment Reference"
VendorExtra1="Biller Code"
VendorExtra2="BPAY Customer Reference No"

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

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

HEADERAR="0[BankBSB:%3D-%3D][BankAccountNumber:%-9D] 01ANZ        [BankAccountName:%-26C][BankReference:%06C]EFTRECEIPTS [Eff
DetailAR="1[DestBSB:%3D-%3D][DestAccount:%-9D] 13[AmountInCents:%010D][DestAccountName:%-32C][DRef:%-18C][BankBSB:%3D-%3D][B
FOOTERAR="7999-999         [TransactionTotalInCents:%010D]0000000000[TransactionTotalInCents:%010D]

HEADERAP="0[BankBSB:%3D-%3D][BankAccountNumber:%-9D] 01ANZ        [BankAccountName:%-26C][BankReference:%06C]EFTPAYMENTS [Eff
DetailAP="1[DestBSB:%3D-%3D][DestAccount:%-9D] 50[AmountInCents:%010D][DestAccountName:%-32C][DRef:%-18C][BankBSB:%3D-%3D][B
FOOTERAP="7999-999         [TransactionTotalInCents:%010D][TransactionTotalInCents:%010D]0000000000
```

UPGRADING EFT PROCESSING KEY STEPS

Install and Activate

After installing the new version of Sage 300 System Manager and relevant modules (for the version you are upgrading to):

Step 1: Run the EFT Processing installation program for the same version as the System Manager version.

Step 2: Refer to Installing EFT Processing

Example: If you upgrade from Sage 300 version 2022 to version 2023, you need to install EFT Processing version 2023. This will create EL70A folder in your Sage program directory.

Step 3: Follow the prompts to confirm the file location of the Sage 300 programs and shared data directory (these should be defaulted from the earlier Sage 300 System Manager installation).

Step 4: Activate the new version in Sage 300 Administrative Services.

Step 5: Enter the Serial Number and Activation Code in the EFT Processing License Screen.

User formats

Step 1: Transfer the custom format to the new version (or new server):

Locate the userformats.ini file in the Sage 300 programs directory, or customization directory, in the previous version or old server;

Copy it over to the ELXXA directory in the programs directory (or customization directory if you are using customization directories).

Example: If you are upgrading from version 2021 to version 2022, you need to copy the files from EL68A to the EL69A folder.

Step 2: Generate a test EFT File and compare the output to a file generated with a previous version. Investigate discrepancies.

> **Tip:**
>
> If there is no userformats.ini file, then the customization must have been done in the formats.ini file. In prior versions, it was possible although not recommended as changes can be overwritten when installing a product update.
>
> In the latest product updates, the formats.ini file includes a hash calculation to ensure the file is not tampered with. If the file is amended in anyway, it cannot be used in Create EFT File and you will receive an error message.
>
> In this case:
>
> - Open up the old formats.ini file
>
> - Locate the file format used by your client (e.g. CAN_RBCACH)
>
> - Copy the format section into a new text file called userformats.ini
>
> - Save this file into the EL70A program (or customization) directory as required

Custom Forms

If you use Custom Forms for remittance advices, copy them to the new ELxxA>ENG folder or the customization directory.

Test the custom layout.

> **Tip:** Test sending emails. If the upgrade is done on a new server, you may need to adjust the details used to send emails.

Payroll upgrade tips

In addition to the steps above, we recommend you test the emailing payslip process based on the latest C/P or U/P report.

In the test database:

- Copy the previous payslip report to the new version's directory (or redo changes based on the new default Sage 300 report.
- Export a range of EFT Employee email address
- Import a tester's email address
- Run the Print Payroll Payslip process.
- Check the email subject and body , the attached PDF Content.
- Confirm that the PDF password matches the formula/employee password
- Re-import the EFT Employee email address

EFT BANKS FORM - OVERVIEW

Use this form to setup EFT details for the banks setup in Sage 300 Bank Services.

> **Note:** The field labels are renamed for different banks depending on the bank formats setup in Formats.ini, when the file type is selected on the Payments and Refunds Tab, Receipts Tab or the Payroll Tab

- Choose the Bank (using the Finder or the Arrows)
- Enter the Bank Details -
- Bank/Branch Code, this should be the BIC/Swift bank code, usually 8 to 11 characters.
- Account Number and Account Name
- IBAN Code, this is the International Banking Account Number – up to 34 characters. (typically required for payments to Europe)
- Additional / configurable fields for the bank: e.g. in Canada: Destination Data Centre and Short Name.

See Also

If using a clearing bank to facilitate bank reconciliation, refer to **Setting up and Using EFT Clearing Bank**.

If you upload the file to a FTP Site (yours or one provided by the bank), **Setup SFTP Output Destination**.

For details on how to use each field, view **EFT Banks Setup Screen**

Payments and Refunds Tab

Use this Tab to configure how EFT Processing creates files for Payments and Refunds, including the output destination, the file type, how to name the file

`File Sequence Number`: the file will be called the File Sequence number on the bank, and after using the file sequence number, EFT Processing will increase the file sequence number by 1.

> **Note:** If the EFT File format includes the File Sequence Number, you generally select the "File Sequence Number" option to name the file. If you select Batch Numbering to name the file, EFT Processing uses the batch number as the [FSN] value.

`Batch Number` : the file will be called APbbbb.{ext} or ARbbbb.{ext} where bbbb is the batch number being exported.

`Pay Period End Date` YYYYMMDD: for EFT Payroll files

`Fixed File Name` : User defined file name using any characters plus the following variables in the file name :

- {yyyy} for the year, {mm} for the Month, {dd} for the Day
- {bbbb} up to 4 numbers for the batch number
- {N} gives you a running number starting from 0 each day. This is often used in conjunction with batch number to keep track of the number of exports in a calendar day.

- {S} gives you a running sequence number for the selected bank- You can use up to 4 numbers for the file sequence number. You typically use this option instead of batch number for a unique number per file.
- {entry} for the entry number, this works with option that each entry to be generated in a single file.

> **Example:** You can use up to 4 digits for {B}, {S} and {N}
> Example for Batch Number 2
> {b} for 1 digit, e.g. 2
> {bb} for 2 digits, e.g. 02
> {bbb} for 3 digits, e.g. 002
> {bbbb} for 4 digits, e.g. 0002

`Global Sequence 1..10`. You are able to share a Global File Sequence number between multiple banks if required. EFT Processing will increase the global file sequence number used by one after creating the file.

> **Note:** If the EFT File format includes the Global File Sequence Number, you need to select the "Global File Sequence Number" option to name the file.

> **Example:** Global File Sequence Example below

Receipts Tab

Use this Tab to configure how EFT Processing creates files for Direct Debit Transactions

Payroll Tab

Use this Tab to configure how EFT Processing creates files for payroll payments.

> **Important!** Before changing any of the settings or other information in the EFT Banks form, make sure you understand the impact the options have on processing with your Bank.

GLOBAL FILE SEQUENCE EXAMPLE

The File Sequence will be shared between whichever module/bank codes that use the same Sequence number.

Refer to Global FSN and EFT Banks Form - Overview on the previous page

Case scenario :

You have 2 accounts with RBC, one account (RBC01 bank code) is for trade operational pay-able/receivable and the other one (RBC bank code) is for administrative expense.

You apply for single Originator ID/ User ID to upload Debit/Credit Transaction files to RBC Portal.

RBC required that each file submitted to the portal has running sequence number.

In EFT Bank Setup for these 2 bank codes, set to use Global Sequence Number.

The numbering will be then based on Global FSN in the EFT Option.

As below it is stating that the next running number is 9.

When Create EFT file is run for payment batch, it will get the next sequence number as the file name.

The subsequent file run either from AR Receipt Batch for the same bank code or RBC bank code, it will share the same Sequence Numbering.

EFT Format Examples

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.

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[Trans-actionTotalInCents:%012D][SourceExtra4:%-10C] [SourceBSB:%8D]0000001[CRLF]"

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

EFT PAYROLL FORMAT EXAMPLE

The format below includes examples of variables to be used in EFT Payroll formats.

XPEMPL: Either CPEMPL if using Canadian Payroll or UPEMPL if using US Payroll.

CHK: Either CPCHKH if using Canadian Payroll or UPCHKH if using US Payroll

ELEMPH: EFT Employees header fields.

ELEMPD: EFT Employee detail file.

Example:

XPEMPL.LASTNAME to add theEmployee surname

ELEMPH.IDEMP for the employee ID

ELEMPD.PAYREF for payment reference

This page intentionally left blank.

OrchidSystems

SYDNEY | AUSTRALIA

Office: 2/273 Alfred Street, North Sydney
New South Wales 2060 Australia

Phone: +61 2 9099 1077
Email: info@orchid.systems

www.orchid.systems

Sage Tech Partner

Work smarter with Orchid