



Document Automation Experts

smartPayables Integration for Finance Systems



+61 1300 378 836 (Australia) | +44 203 355 1237 (UK)
+1 213 291 0523 (USA) | +63 2 817 4901 (Philippines)
info@redmap.com | www.redmap.com

Table of Contents

- 1. Introduction 4
 - 1.1. Purpose of this document 4
- 2. Overview 5
 - 2.1. What is smartPayables 5
 - 2.2. Benefits of a smartPayables solution 6
 - 2.3. Required integration 6
 - 2.4. Data transfer approach 7
 - 2.5. Data Downloads 7
 - 2.5.1. Required data 7
 - 2.5.2. Approach 7
 - 2.6. Data Uploads 8
 - 2.6.1. Required data 8
 - 2.6.2. Approach 8
 - 2.7. Linking to the document 9
 - 2.7.1. Approach 9
- 3. Staging table schemas 10
 - 3.1. Lookup_Entity table 10
 - 3.2. Lookup_Invoice table 10
 - 3.3. Lookup_PurchaseOrder table 10
 - 3.4. Lookup_PurchaseOrder_Line table 12
 - 3.5. Lookup_Receipt table 12
 - 3.6. Lookup_Receipt_Line table 12
 - 3.7. Lookup_Vendor table 14
 - 3.8. Lookup_Account table 14
 - 3.9. Lookup_Item table 15
- 4. Data download CSV file formats 16
- 5. Data upload CSV file formats 18

Document Control

Document Author	Soruban Kanapathipillai
Document Date	13 June 2013
Current Version	0.8
Document Title	smartPayables Integration for Finance Systems
Project Number	

Document Revisions

Date	Version	Author	Comments
11/2/2013	0.1	Soruban Kanapathipillai	First draft
12/2/2013	0.2	Ben Woolley	Second draft
14/2/2013	0.3	Soruban Kanapathipillai	Third draft
18/3/2013	0.4	Soruban Kanapathipillai	Adjustments to invoice data download file structure
05/06/2013	0.5	Afshin Baktashi	Extended the scope to include Foreign invoices PO Portal RCTI Project job costing
11/06/2013	0.6	Ben Woolley	Edits
13/06/2013	0.7	Soruban Kanapathipillai	Minor revisions
15/06/2013	0.8	Afshin Baktashi	Added diagram and merged in various comments Modify Invoice table structure to allow for payment approval process

Document Distribution

Name	Company	Position	Phone	Email



Copyright Reserved. 2013

The staff of Redmap Networks Pty Limited has taken due care in preparing this Report. However, nothing contained herein modifies or alters in any way the standard terms set out in the Redmap Software Support and Development Arrangements under which it is provided.

All rights in Software including this Manual, Sample Material and methods are reserved and are supplied on the express condition that the client acquires only a licence to use, and the title and copyright vest entirely in Redmap Networks Pty Limited. Reproduction (except as provided herein) in any form or by any means is not permitted nor shall it be leant, sold, hired out or made available to persons other than the staff of the client agencies. The client undertakes to prevent copying or use by 3rd parties and the due observance of the above.

1. INTRODUCTION

1.1. Purpose of this document

This document is the functional specification for the development of xConnectors. xConnectors are the integration between the Redmap smartPayables solution and 3rd party finance systems. The main aim of this specification is to ensure that there is standard approach to each implementation. This is required for long term sustainability and supporting of the xConnectors.

2. OVERVIEW

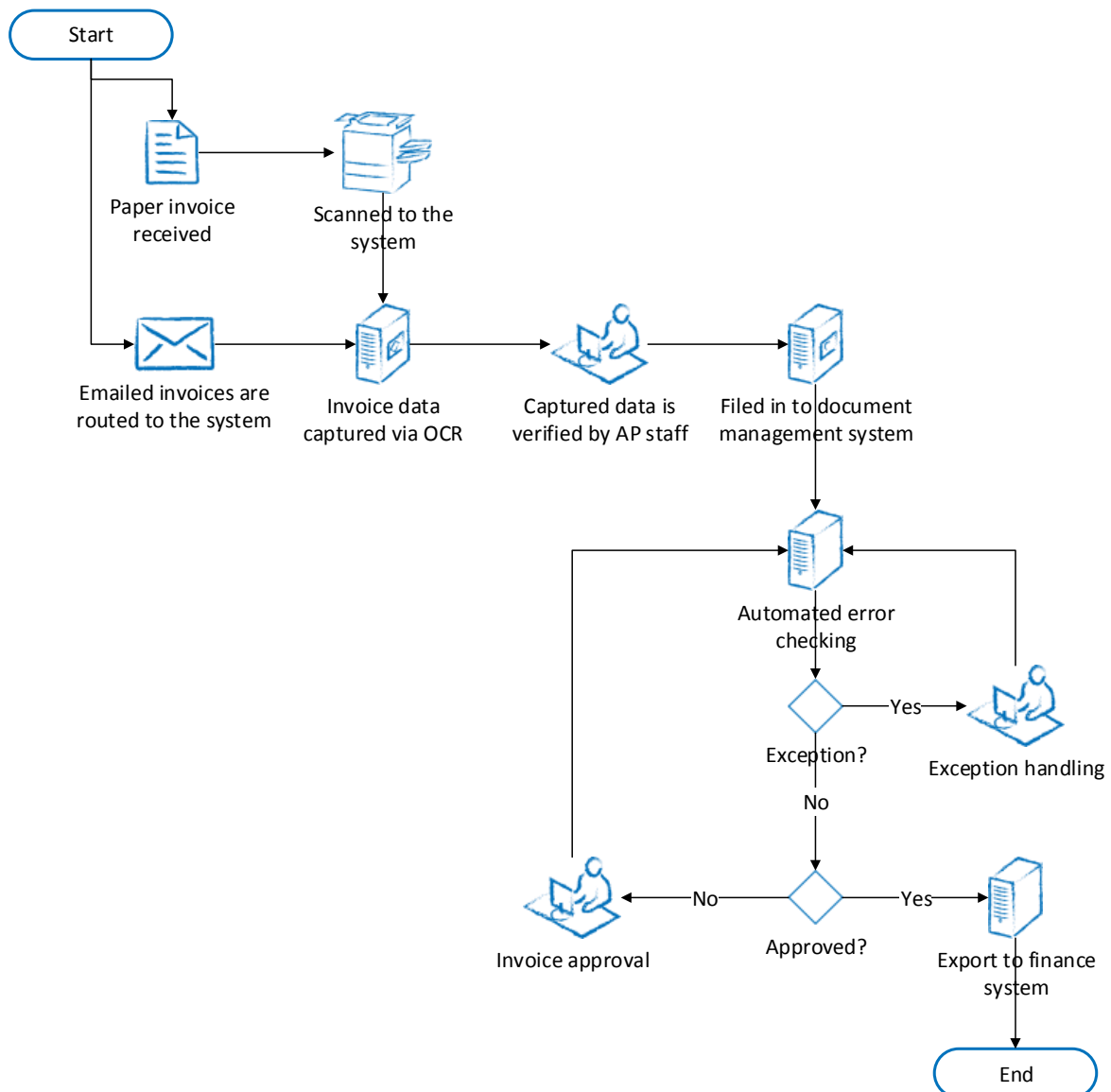
2.1. What is smartPayables

Redmap's smartPayables solutions allow businesses to automate the processing, approval and uploading of vendor invoices to their finance system.

A typical smartPayables solution:

- ➔ Receives invoices via paper or email
- ➔ Captures data from the invoices
- ➔ Electronically archives a copy of the invoice
- ➔ Applies business rules (e.g. error checking, PO matching against ERP)
- ➔ Manages invoices approvals workflows
- ➔ Exports the appropriate invoice data to the finance system

An example of this workflow is illustrated below:



2.2. Benefits of a smartPayables solution

The smartPayables solution is designed to allow businesses to retain the value in their existing investments in their finance system, whilst providing them with the efficiencies and cost savings that come with automating tedious, time-consuming processes such as data entry, error checking and filing.

smartPayables solutions can deliver:

- ➔ Decrease the cost to code and enter invoices by at least 50%;
- ➔ Automated Purchase Order and Delivery Order matching;
- ➔ Streamline the approval of the invoices and remove all the wasted labour in managing these approvals;
- ➔ Eliminate all costs in filing invoices for retention; and
- ➔ Immediate visibility of all outstanding payables.

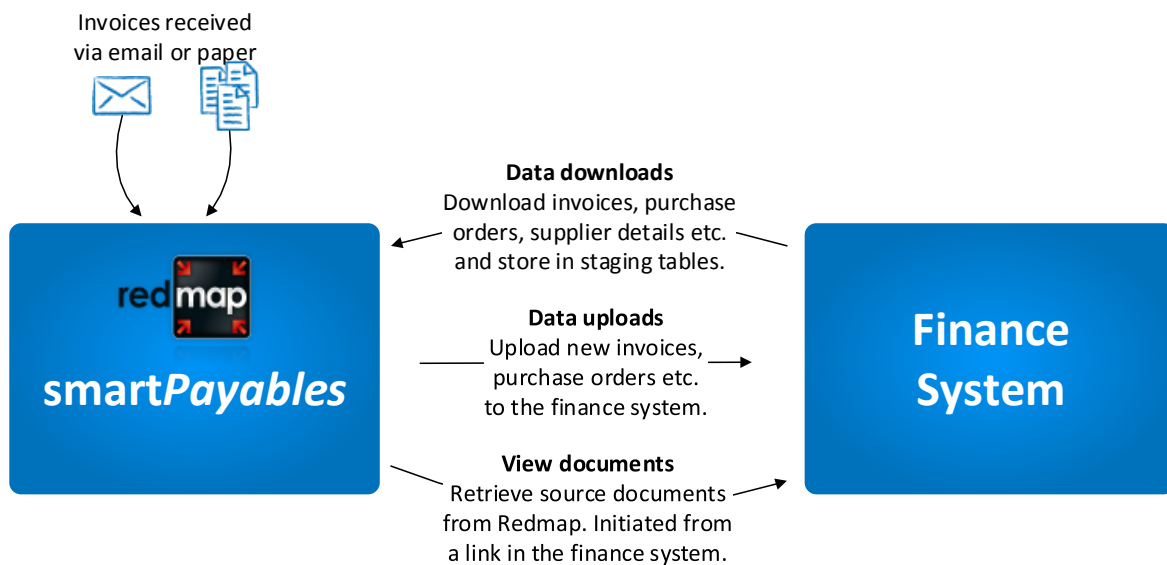
2.3. Required integration

The smartPayables solution must be integrated to the finance system to achieve the required outcomes.

The required integration between the Redmap smartPayables solution and the finance system covers the following items:

- ➔ Data downloads
 - Getting staging data from the finance system and storing it in staging tables within the smartPayables system for further use.
- ➔ Data uploads
 - The finance system shall accept details related to new transactions (e.g. invoices, purchase orders, purchase receipts, document links) from the smartPayables solution; and
 - The finance system shall create the appropriate records within the finance system based on the details provided.
- ➔ View documents
 - The finance system exposes a URL to the user on relevant records (e.g. invoice screen) to be able to retrieve the source document(s) related to that record (e.g. view the invoice details from within the finance system and click on a link on the screen to retrieve the electronic copy of the invoice image from the smartPayables solution).

This flow of information is depicted in the following diagram:



2.4. Data transfer approach

The transfer of data between SmartPayables and the ERP is typically achieved via saving the data into text files. The files are stored in a predetermined folder for either system to collect.

The text files shall include the following characteristics:

- ➔ Data shall be stored as comma separated values (CSV) in the files;
- ➔ Each value in the text file shall be wrapped in double quotation marks (“”);
- ➔ The files shall include column headers on the first row of each file; and
- ➔ The files shall have a file type of .CSV.

Whilst this is a typical approach and most ERP systems support this methodology it is recognized that there exists a need to maintain agility and flexibility in supporting multiple types of interfaces when the requirement needs a different approach. Other approaches to consider are:

- ➔ Using XML rather than CSV as the file format;
- ➔ Using fix length structured files rather than CSV format;
- ➔ Using FTP or other file transfer mechanisms to ensure Cloud support;
- ➔ Web services in the case of hosted platforms;
- ➔ Direct updating of the Redmap staging database tables.

2.5. Data Downloads

Various components within the Redmap smartPayables solution require access to data stored within the finance system. This data is used within the smartPayables solution as the basis for applying the various business rules defined by the customer. We have taken the approach to access all ERP data via staging tables to ensure that standardisation is possible and also hosted platforms in the public or private cloud are also supported. It is worth noting that in certain cases a nightly update of the staging data may not be adequate as the customer may wish to pay certain invoices on the day that they are posted.

2.5.1. Required data

The following pieces of data must be made available to the smartPayables solution by the finance system:

- ➔ Purchase orders
- ➔ Purchase receipts
- ➔ Invoices
- ➔ Vendors
- ➔ General ledger accounts
- ➔ Items

2.5.2. Approach

File transfer

The reference data shall be exported to our staging database via a selected file (csv, xml) by the finance system.

The finance system shall place these files in a predefined location on the customer's network.

The finance system shall automatically export a fresh set of data on a predefined scheduled (typically nightly).

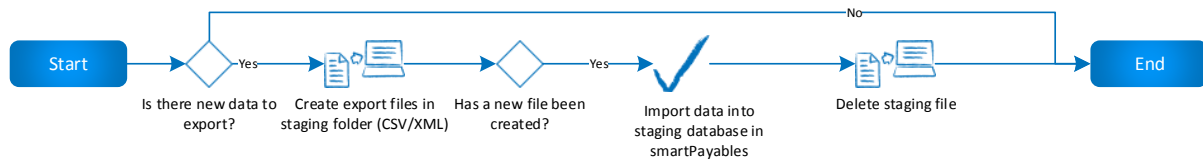
Note: If the solution is hosted in the Cloud then FTP is required to transfer the files across the networks.

The text files shall be imported in to a series of staging tables using an existing, automated CSV/xml file import mechanism available within the Redmap smartPayables solution.

The files in the nominated folder shall be removed from the folder after the data in the file is imported in to the smartPayables solution.

Each text file shall contain the data intended for a different staging table (the tables the data values are to be imported into are defined in section 3).

The following diagram shows illustrates the process of data transfer between systems and is applicable in both direction.



Web services

The reference data shall be exported to our staging database via web services available by the finance system.

2.6. Data Uploads

Once purchase orders, purchase receipts or invoices are processed within the smartPayables solution the smartPayables solution shall make this data available to the finance system. The finance system shall then create the relevant transactions within the finance system based on the details provided.

2.6.1. Required data

The following pieces of data shall be made available from the smartPayables solution to the finance system.

- ➔ Non-PO invoice
- ➔ PO-based invoice
- ➔ Non-PO credit note
- ➔ PO-based credit note
- ➔ Document link
- ➔ Open Purchase orders
- ➔ Open Purchase receipts

2.6.2. Approach

File transfer

The data to be uploaded shall be exported to a series of text files (CSV/xml) by the smartPayables solution.

The smartPayables solution shall place these files in a predefined location on the customer’s network.

The smartPayables solution shall export the data to the folder when the data is ready to be uploaded to the finance system (i.e. the invoice has passed all error checks and has been approved for entry in to the finance system).

The files shall be uniquely named, but each filename shall be prefixed with a defined value to clearly identify the type of data that has been exported in the file (e.g. Invoice_20130101.csv shall indicate that the file contains invoice transactions).

The finance system shall poll the nominated folder regularly and import the data from any files in the folder. The files in the folder shall be removed from the folder after the data in the file is imported in to the finance system. It is the responsibility of the process pulling the data into the finance system to ensure the data is successfully imported and the files have been removed from the folder.

Note: It is recommended that a process is considered for moving the files into another folder and maintaining a copy for a period of 30 days.

The columns to be included in each text file are listed in section 5 of this document.

Web Services

In the event that web services are available to allow for uploading of data into the finance system the data uploading is usually done in real time and on a transaction basis. As an example as the final approval for a particular invoice is complete the web service would be called to upload the invoice data in real time. The design of this approach will largely depend on the existing web service capability and will be designed on a case by case basis.

2.7. Linking to the document

Once documents (e.g. invoices) are processed through the smart*Payables* solution and the relevant transactions are exported to the finance system (via the data upload mechanisms), the transaction details shall be in the finance system and the source document shall be in the smart*Payables* solution. A user in the finance system should be provided with access to the source document from within the finance system.

2.7.1. Approach

The finance system shall expose a mechanism to the user on relevant screens (e.g. invoice screen). The mechanism shall open a web browser to the document link URL defined for the record.

3. STAGING TABLE SCHEMAS

The following staging tables shall be created to host various data that the smartPayables solution components rely upon.

3.1. Lookup_Entity table

The Lookup_Entity table shall contain a list of entities (e.g. discrete commercial entities or financial databases). The table shall have the following structure:

Field	Type	Description / Notes
EntityCode	Nvarchar(20)	Primary key
EntityName	Nvarchar(50)	

3.2. Lookup_Invoice table

The Lookup_Invoice table shall contain a list of invoices already in the finance system. The table shall have the following structure:

Field	Type	Description / Notes
EntityCode	Nvarchar(20)	Primary key
InvoiceNumber	Nvarchar(50)	Primary key
InvoiceDate	Datetime	
PONumber	Nvarchar(50)	
InvoiceDueDate	DateTime	
InvoiceStatus	NVarchar(10)	Typically used for status such as hold and payment status – Note that multiple flags could be supported. e.g NY – first N referring to not paid and second Y referring to posted.
InvoiceCurrencyCode	Nvarchar(10)	
InvoiceCurrencyRate	Nvarchar(10)	
POJobID	Nvarchar(20)	Project or job code Related to project job costing
POJobCostCentre	Nvarchar(20)	Cost centre code related to project job costing
VendorCode	Nvarchar(20)	
TaxCode	Float	
Tax	Float	
Total	Float	

3.3. Lookup_PurchaseOrder table

The Lookup_PurchaseOrder table shall contain a list of purchase orders already in the finance system. The table shall have the following structure:

Field	Type	Description / Notes
EntityCode	Nvarchar(20)	Primary key
PONumber	Nvarchar(50)	Primary key
PODate	Datetime	
VendorCode	Nvarchar(20)	
POCurrencyCode	Nvarchar(10)	
POCurrencyRate	Nvarchar(10)	
Total	Float	
POJobID	Nvarchar(20)	Project or job code Related to project job costing
POJobCostCentre	Nvarchar(20)	Cost centre code related to project job costing
POStatus	Nvarchar(10)	Typically used for status such as back order flag – Note that multiple flags could be supported. e.g NY – first N referring to not on back order and second Y referring to posted.
PORaiser	Nvarchar(100)	The name of the person that raised the purchase order

3.4. Lookup_PurchaseOrder_Line table

The Lookup_PurchaseOrder_Line table shall contain a list of lines for the purchase orders in the finance system. The table shall have the following structure:

Field	Type	Description / Notes
EntityCode	Nvarchar(20)	Primary key
PONumber	Nvarchar(50)	Primary key
POLineID	Int	Primary key - Sequence
POLineItemCode	Nvarchar(50)	Internal stock code
SupplierStockCode	Nvarchar(50)	Supplier stock code
POLineAccount	Nvarchar(50)	
POLineDescription	Nvarchar(100)	
POLineQuantity	Float	
POLineQuantityRecieved	Float	Quantity of items received
POLineUnitPrice	Float	Unit price for item
POLineTax	Float	
POLineTotal	Float	
POLineStatus	Nvarchar(10)	Typically used for status such as back order flag – Note that multiple flags could be supported e.g NY – first N referring to not on back order and second Y referring to posted.

3.5. Lookup_Receipt table

The Lookup_Receipt table shall contain a list of receipts from the finance system. The table shall have the following structure:

Field	Type	Description / Notes
EntityCode	Nvarchar(20)	Primary key
ReceiptNumber	Nvarchar(50)	Primary key
ReceiptDate	Datetime	
PONumber	Nvarchar(50)	
ReceiptStatus	Nvarchar(50)	

3.6. Lookup_Receipt_Line table

The Lookup_Receipt_Line table shall contain a list of lines related to the receipts in the finance system. The table shall have the following structure:

Field	Type	Description / Notes
EntityCode	Nvarchar(20)	Primary key

Field	Type	Description / Notes
ReceiptNumber	Nvarchar(50)	Primary key
ReceiptLineID	Int	Primary key
ReceiptLineItem	Nvarchar(50)	
ReceiptLineQuantity	Float	

3.7. Lookup_Vendor table

The Lookup_Vendor table shall contain a listing of existing vendors within the finance system. The table shall have the following structure:

Field	Type	Description / Notes
EntityCode	Nvarchar(20)	Primary key
VendorCode	Nvarchar(20)	Primary key Vendor's ID in finance system
VendorABN	Nvarchar(11)	Vendor's ABN
VendorName	Nvarchar(200)	Vendor's name
VendorPhone	Nvarchar(25)	Vendor's main phone number used for additional lookup if required
AccountCode	Nvarchar(20)	The default account code to be applied for invoice transactions for this vendor. This shall simplify data entry by automatically filling in the GL Code in cases where the vendor mostly only uses a single GL Code.
GSTApplicable	Nvarchar(5)	Flag to indicate whether the vendor is registered for GST
DefaultGL_Code	Nvarchar(25)	Default GL account code for this vendor.
AccountStatus	Nvarchar(1)	Flag to indicate if the account is active or not.
Currency_code	Nvarchar(10)	Relevant currency code for vendor
Tax_Code	Nvarchar(10)	Relevant vendor tax code
VendorReference	Nvarchar(20)	Optionally Used to store additional vendor details such as BPayBillerCode

3.8. Lookup_Account table

The Lookup_Account table shall contain a listing of General Ledger Accounts from the finance system. The table shall have the following structure:

Field	Type	Description / Notes
EntityCode	Nvarchar(20)	Primary key
AccountCode	Nvarchar(20)	Primary key General ledger account code

Field	Type	Description / Notes
AccountDescription	Nvarchar(200)	General ledger account description
AccountType	Nvarchar(50)	Optional vendor account type

3.9. Lookup_Item table

The Lookup_Item table shall contain a listing of items from the finance system. The table shall have the following structure:

Field	Type	Description / Notes
EntityCode	Nvarchar(20)	Primary key
ItemCode	Nvarchar(20)	Primary key
ItemDescription	Nvarchar(200)	
ItemUnitPrice	Float	
ItemGLCode	NVarchar(50)	
SupplierStockCode	Nvarchar(50)	Supplier stock code

4. DATA DOWNLOAD CSV FILE FORMATS

The following CSV file formats are included as a guideline to the file formats available. In the absence of any other design it is recommended to use the provided reference. It is possible however to change the format of the files if the chosen finance system already supports a particular format.

The CSV files to be used to transfer data from the finance system to the Redmap smartPayables solution shall conform to the following formats.

File name prefix	Columns	Description / Notes
Invoice_	EntityCode InvoiceNumber InvoiceDate PONumber InvoiceCurrencyCode InvoiceCurrencyRate POJobID POJobCostCentre VendorCode TaxCode InvoiceTax InvoiceTotal InvoiceDescription	
Invoice_Line_	InvoiceNumber InvoiceLineID InvoiceLineDescription InvoiceLineGLCode InvoiceLineTax InvoiceLineAmount	
Vendor_	EntityCode VendorCode VendorABN VendorName AccountCode GSTApplicable	
GLAccount_	EntityCode AccountCode AccountDescription AccountType	

File name prefix	Columns	Description / Notes
Item_	EntityCode ItemCode ItemDescription ItemUnitPrice ItemGLCode	
PO_	EntityCode PONumber PODate VendorCode POTotal Status PORaiser	
PO_Line	PONumber POLineID POLineItem POLineAccount POLineDescription POLineQuantity POLineTax POLineTotal	
Receipt_	EntityCode ReceiptNumber ReceiptDate PONumber ReceiptLineID ReceiptLineItem ReceiptLineQuantity ReceiptStatus	
Receipt_Line	ReceiptNumber ReceiptLineID ReceiptLineItem ReceiptLineQuantity	

5.

5. DATA UPLOAD CSV FILE FORMATS

The following CSV file formats are included as a guideline to the file formats available. In the absence of any other design it is recommended to use the provided reference. It is possible however to change the format of the files if the chosen finance system already supports a particular format.

The CSV files to be used to transfer data from the finance system to the Redmap smartPayables solution shall conform to the following formats.

File name prefix	Columns	Description / Notes
Invoice_	EntityCode	
	InvoiceNumber	
	InvoiceDate	
	VendorCode	
	InvoiceDescription	
	InvoiceLineID	
	InvoiceLineDescription	
	InvoiceLineGLCode	
	InvoiceLineTax	
	InvoiceLineAmount	Exclusive of tax
	InvoiceTax	
	InvoiceTotal	Inclusive of taxes
POInvoice_	EntityCode	
	PONumber	
	InvoiceNumber	
	InvoiceDate	
	VendorCode	
	InvoiceDescription	
	InvoiceLineID	
	InvoiceLineItemCode	
	InvoiceLineQuantity	
	InvoiceLineTax	
	InvoiceLineAmount	
	InvoiceTax	
	InvoiceTotal	
Credit_	EntityCode	
	CreditNumber	
	CreditDate	

File name prefix	Columns	Description / Notes
	VendorCode	
	CreditDescription	
	CreditLineID	
	CreditLineDescription	
	CreditLineGLCode	
	CreditLineTax	
	CreditLineAmount	Exclusive of tax
	CreditTax	
	CreditTotal	Inclusive of taxes
POCredit_	EntityCode	
	PONumber	
	CreditNumber	
	CreditDate	
	VendorCode	
	CreditDescription	
	CreditLineID	
	CreditLineItemCode	
	CreditLineQuantity	
	CreditLineTax	
	CreditLineAmount	
	CreditTax	
	CreditTotal	
PO_	EntityCode	
	PONumber	
	PODate	
	VendorCode	
	PODescription	
	POLineID	
	POLineItemCode	
	POLineQuantity	
	POLineTax	
	POLineAmount	
	POTax	

File name prefix	Columns	Description / Notes
	POTotal	
Receipt_	EntityCode	
	ReceiptNumber	
	ReceiptDate	
	PONumber	
	ReceiptLineID	
	ReceiptLineItemCode	
	ReceiptLineQuantity	
Link_	DocumentType	One of the following values: <ul style="list-style-type: none"> ➔ Invoice ➔ CreditNote ➔ PO ➔ Receipt
	ID	The identifier for the record (e.g. the invoice number or the PO number)
	URL	The URL at which the document(s) can be accessed