



Coupa NetSuite P2P Integration Guide

Doc Version 11.0.1
Last Updated: 2022-12-06 – Bundle 11.0.0

Contents

Coupa NetSuite P2P Integration Guide	1
Contents	2
P2P Dataflow and Integration points	5
Overview P2P.....	5
Overview Expenses.....	6
Integration Method.....	6
Coupa P2P & Expense Bundle	7
Client Installation/Upgrade Overview	7
Activities	7
Locate the Coupa Bundle Script	8
Confirm & Install contents of the Bundle.....	9
Configuration of the Bundle	10
Netsuite Vendor to Coupa Supplier	11
Overview	11
Script Sequence Explained.....	11
Deployment Parameter Setup	11
Activities	11
Tips to help you succeed.....	11
Pre-Requisites.....	11
Script parameters	12
Coupa SIM Supplier to NetSuite Vendor	17
Overview	17
Script Sequence Explained.....	17
Deployment Parameter Setup	17
Activities	17
Tips to help you succeed.....	17
Pre-Requisites.....	17
Script parameters	18
Special Considerations with the SIM Bundle	21
Netsuite Exchange Rate to Coupa Exchange Rates 2.0	22
Overview	22
Script Sequence Explained.....	22
Activities	22
Tips to help you succeed.....	22
Pre-Requisites.....	22
Dynamic - Coupa Chart of Account to Coupa	25
Overview	25
Script Sequence Explained.....	25
Activities	25
Tips to help you succeed.....	25
Pre-Requisites.....	25
Coupa Invoice 1.0 & 2.0 to Netsuite Vendor Bill.....	32
Overview	32
Script Sequence Explained.....	32



Activities	32
Tips to help you succeed.....	32
Pre-Requisites.....	32
See “Common Custom Field Examples” below	35
Netsuite Payments to Coupa Invoice & ER Payments 2.0.....	43
Overview	43
Script Sequence Explained.....	43
Activities	43
Tips to help you succeed.....	43
Pre-Requisites.....	43
Coupa Expense Report to Netsuite Expense Report	47
Overview	47
Script Sequence Explained.....	47
Activities	47
Tips to help you succeed.....	47
Pre-Requisites	47
Coupa Expense Report Payment to Netsuite ER Payment	52
Overview	52
Script Sequence Explained.....	52
Activities	52
Pre-Requisites.....	52
Tips to help you succeed.....	52
Coupa Expense Report to Netsuite Vendor Bill	55
Overview	55
Script Sequence Explained.....	55
Activities	55
Tips to help you succeed.....	55
Pre-Requisites.....	55
Coupa Charge to Netsuite Credit Card transaction.....	60
Overview	60
Script Sequence Explained.....	60
Activities	60
Tips to help you succeed.....	60
Pre-Requisites.....	60
Coupa PrePayment to Netsuite PrePayment	64
Overview	64
Script Sequence Explained.....	64
Activities	64
Tips to help you succeed.....	64
Pre-Requisites.....	64
Coupa Invoice Payment to Netsuite Vendor Bill Payment	67
Overview	67
Script Sequence Explained.....	67
Activities	67
Tips to help you succeed.....	67
Pre-Requisites.....	67
Coupa Expense Report Payment to Vendor Bill Payment	70
Overview	70
Script Sequence Explained.....	70



Activities	70
Tips to help you succeed.....	70
Pre-Requisites.....	70
Coupa Dynamic Discounting to Netsuite Invoice	73
Overview	73
Script Sequence Explained.....	73
Activities	73
Tips to help you succeed.....	73
Pre-Requisites.....	73
APPENDIX	75
Expense PreApproval Process.....	75

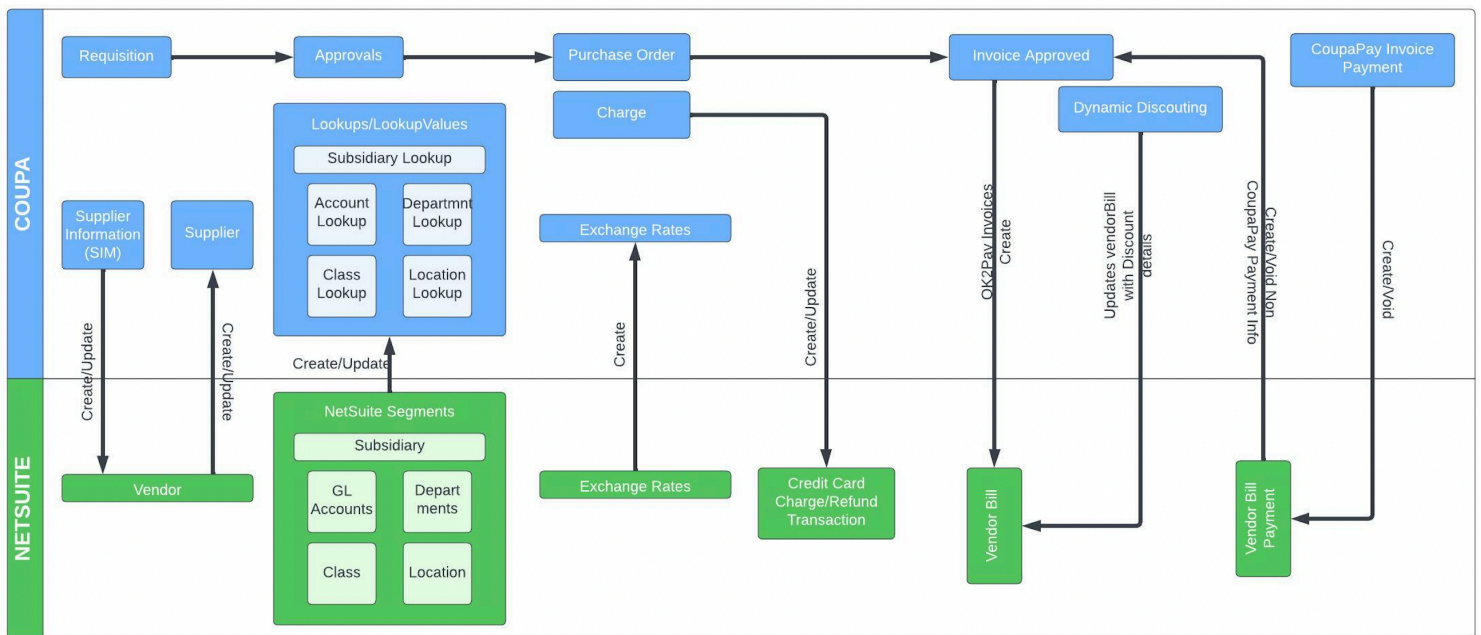
P2P Dataflow and Integration points

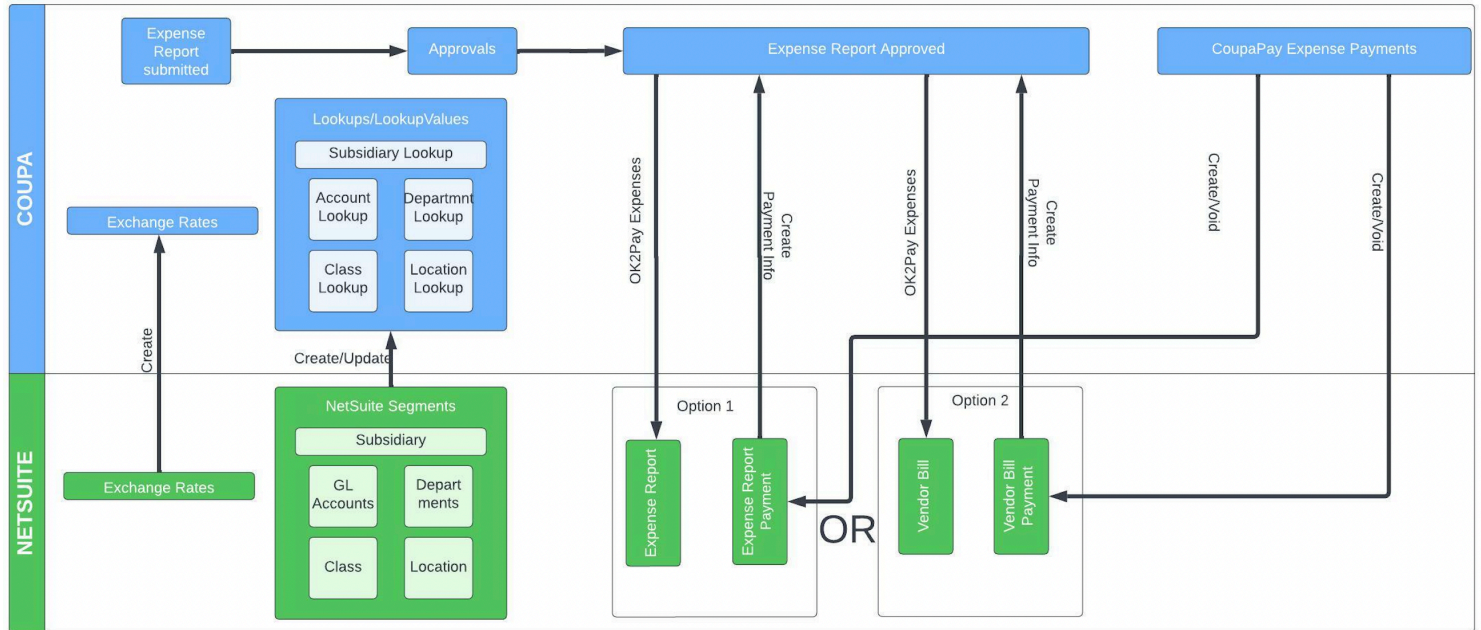
Overview P2P

The diagram below indicates the integration points between Netsuite and Coupa. Typically we have master data - suppliers, Netsuite GL Account attributes, exchange rates and transaction data - bill payments coming from Netsuite into Coupa and approved invoices going from Coupa into Netsuite.

Customers using Coupa Pay may take Virtual Card Charges for PO Payments from Coupa to Netsuite as Credit Card Transactions and Coupa Invoice Payments from Coupa to Netsuite as Vendor Bill Payments. (highlighted red)

With SIM (Supplier Information Management), new supplier records can be initiated via a supplier information request from within Coupa. Once all the information is populated, the supplier record is created in NetSuite and simultaneously activated in Coupa. The diagram below indicates the additional integration points which include SIM.





*expense report will not be created for Expense PreApprovals

Integration Method

Coupa integrates seamlessly with Netsuite using the Coupa REST API and Netsuite's Suitescript. Coupa provides rich, robust access to read, edit, or integrate data via a [RESTful API](#). This UTF-8 XML based interface allows you to create, update, and take action on individual records within Coupa by making an HTTP request call.

Built on industry standard JavaScript that ensures portability and rapid developer productivity, SuiteScript enables full-featured application-level scripting capabilities throughout NetSuite. Suitescript provides a complete set of flexible scripting models to support customization needs, including user event-based scripts and scheduled scripts.

The Coupa Netsuite Integrations have been built to meet and comply with the documented practices for architecture, development, privacy and security of the Netsuite SuiteCloud platform and is certified and approved by the "[BUILT FOR NETSUITE](#)" program.

Coupa Netsuite Bundle scripts should not be modified. Modified scripts will not be supported by Coupa Support.



Coupa P2P & Expense Bundle

Client Installation/Upgrade Overview

Coupa leverages SuiteBundler, which makes it easy to package and deploy customizations and applications, built on the SuiteCloud platform. Whether you are moving customizations from a sandbox environment to production or looking for the latest Coupa to Netsuite integration package, SuiteBundler lets you easily manage groups of customizations, manage versioning and deploy customizations nearly instantaneously.

Activities

You will need to do the following deploy the Coupa / Netsuite Bundler

- | | |
|---|---|
| 1 | Locate the Coupa Bundle Script |
| 2 | Confirm contents of the Bundle |
| 3 | Install the Bundle |
| 4 | Confirm successful installation of the Coupa Bundle |
| 5 | Begin configuration |



Locate the Coupa Bundle Script

Step 1.1

Ensure you have given your Netsuite Account ID to your Coupa Implementation Manager.

Locate the Bundler

Navigation Path: Customization | SuiteBundler | Search & Install Bundles

Transactions	Lists	Reports	Customization	Documents	Setup	Support	SuiteS
			Customization Manager				Viewing: This Mon
			Lists, Records, & Fields				
			Forms				
			Scripting				
			Plug-ins				
			Centers and Tabs				
			SuiteBundler		Create Bundle		
					Search & Install Bundles		
					Bundle Audit Trail		

Step 1.2

Search for “Coupa P2P + Expenses” and click on Bundle

Search & Install Bundles

Search

Basic | Advanced

LEAVE THE KEYWORDS BOX BLANK AND CLICK SEARCH TO VIEW THE MOST POPULAR SUITEAPPS

KEYWORDS

“Coupa P2P + Expenses”

[Installation Terms of Service](#)

NAME	BUNDLE ID
Coupa P2P + Expenses Bundle	84306



Confirm & Install contents of the Bundle

Step 1.3

Switch to the Components View and verify the contents match below:

NAME	ID
Custom Fields	
Transaction Body Fields	
CoupaER Number	custbody_coupa_er_number
Coupa Revision Number	custbody_coupa_er_internalrevision
Transaction Column Fields	
Coupa ERReason	custcol_coupa_er_reason
Coupa ER Description	custcol_coupa_er_desc
CoupaER Line Id	custcol_coupa_er_lineid
Custom Lists/Records	
Lists	
Coupa Vendor PhoneFax List (3 records)	customlist_vendor_phonefax_list
File Cabinet	
Files	
Coupa-VendorPayment.js	7072
Coupa-FxRate.js	7074
Coupa-ExpenseReport.js	7070
Coupa-Invoice.js	7071
Coupa-DynamicAcct.js	7069
Coupa-Vendor.js	7073
Saved Searches	
Account	
Coupa Account Integration Search	customsearch_coupa_accs_accsearch
Class	
Coupa Class Integration Search	customsearch_coupa_accs_classsearch
Contact	
Coupa Primary Contact Search	customsearch_coupa_vendor_primarycontact
Department	
Coupa Department Integration Search	customsearch_coupa_accs_deptsearch
Location	
Coupa Location Integration Search	customsearch_coupa_accs_locsearch
Subsidiary	
Coupa Subsidiary Integration Search	customsearch_coupa_accs_subsearch
SuiteScripts	
Scheduled	
Coupa Invoice Payment Integration	customscript_coupa_invoice_payment
Coupa Exchange Rate Integration	customscript_coupa_fxrate
Coupa Expense Report Integration	customscript_coupa_expense_report
Coupa Invoice Integration	customscript_coupa_invoice
User Event	
Coupa Dynamic COA Integration	customscript_coupa_dynamic_accounts
Coupa Vendor Integration	customscript_coupa_vendor

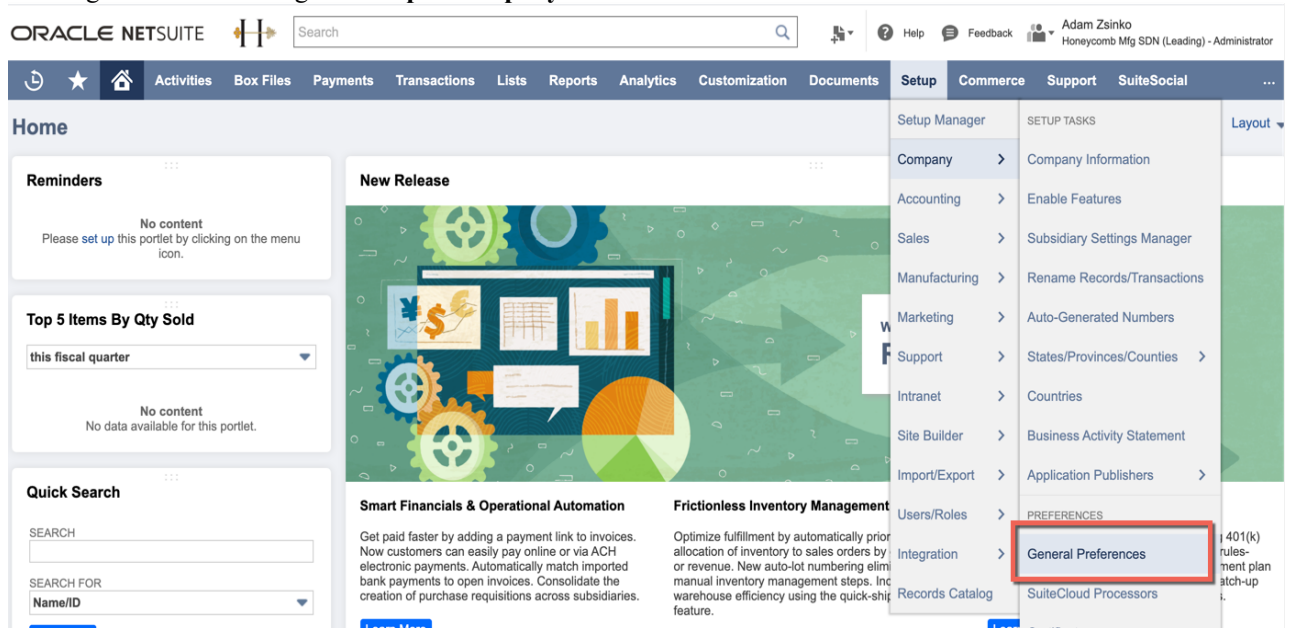


Configuration of the Bundle

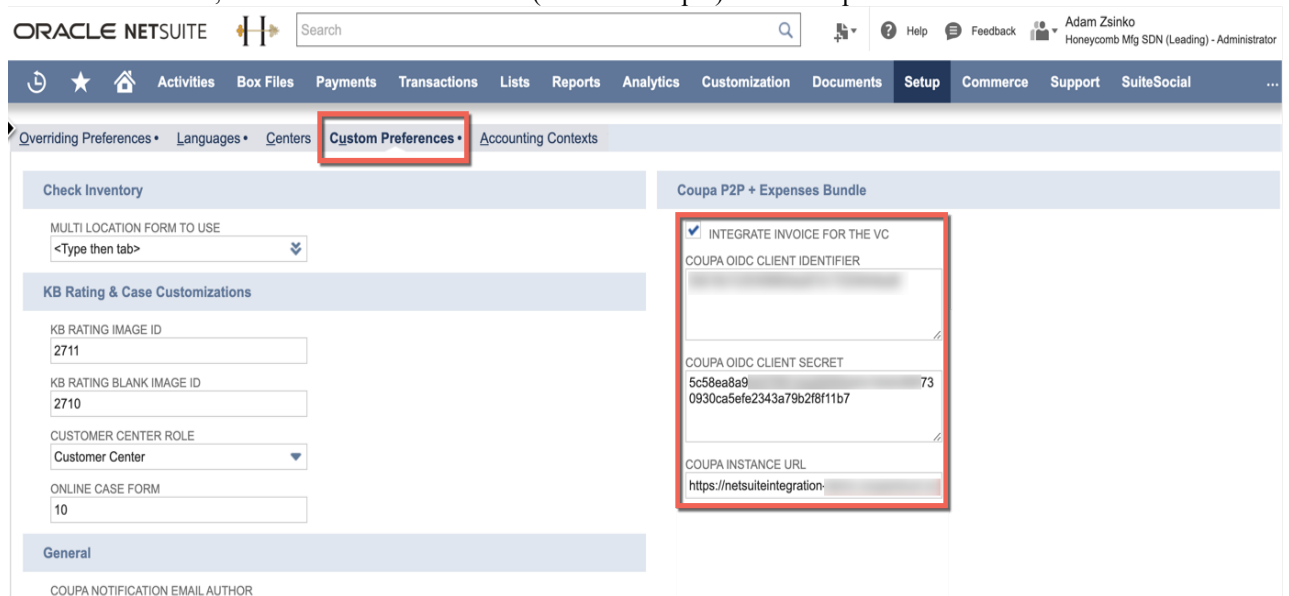
- API - OIDC configuration (optional)

With Coupa Release 31 secure Auth 2.0 API connectivity has been introduced. Netsuite Bundle is supporting this new API connectivity method.

1. Create OIDC connection as per [Coupa Success Article](#)
2. Setup OIDC scope for read/write for the objects integrated via the Netsuite Bundle
3. Login to Netsuite and go to **Setup > Company > General Preferences > Custom Preference**



4. Enter ClientID, Client Secret & OIDC URL (created in step 2) under Coupa P2P Bundle section



Netsuite Vendor to Coupa Supplier

Overview

This integration syncs up Vendor records between Netsuite and the Supplier record in Coupa. Netsuite is the vendor master and the records flow from Netsuite into Coupa. The User Event Suitescript is used to capture the “create” and “update” events. Based on the event type either a new Supplier is created in Coupa or an existing Supplier in Coupa is updated.

Script Sequence Explained

This integration is real time as the Vendor record is created/updated in Netsuite

- The Coupa API call to create/update the supplier record is called as soon as the Netsuite user event is captured
- The `nlapiLoadRecord('vendor', recordid)` function is used to read the Netsuite vendor record
- The `nlapiRequestURL` function is used to do a HTTP POST/PUT to create/update the supplier record in Coupa.

Deployment Parameter Setup

Now that you have successfully installed the bundle, you will need to decide which parameters are relevant for your Coupa implementation.

Activities

You will need to do the following to ensure a successful deployment of the Coupa Vendor Integration Script

- 1 **Review Parameter List**
- 2 **Find and Define Deployment Parameter Values**

Tips to help you succeed

- Ensure your vendor record in Netsuite is associated with the appropriate subsidiary values and currency association
- Ensure the supplier name is unique in Coupa
- Only load NetSuite vendors in Coupa that you will create transactions for in Coupa
- Once vendor records are created in Coupa ensure supplier record has appropriate Coupa configuration based on your business process

Pre-Requisites

- Coupa P2P & Expenses Bundle deployed
- Coupa API Key or OIDC auth created
- You will need to baseline the Coupa Suppliers via the CSV UI Loader

Review Parameter List

Review parameter list and decide which ones are relevant to your company's implementation of Coupa.

Please note the fields that are mandatory are noted with an asterisk *

[Script parameters](#)

Script Definition					Script Deployment	
Label	Example	Id	Type	Comments	Example	Your Value
Coupa URL*	The Coupa instance the integration connects to	_vendor_url	Free-Form Text		https://abc.coupa host.com	
Coupa API Key*	The API key for Coupa's REST based API	_vendor_apikey	Free-Form Text	A Coupa Admin can generate the Coupa API Key in Coupa https://xxx.coupahost.com/api_keys	4dtyr7657fgfh5757fghhg	
Error Email Notification Address	Email addresses to which exception emails to be sent	_vendor_erroremailnotify	Free-Form Text		abc@abc.com , abcd@abcd.com	
Account Name*	As per standard, it will be company name with which environment its been setup (Test or Prod)	_vendor_accountname	Free-Form Text	<Test/Prod>	Coupa Test	
Use Company Name as Display Name?	Use Company Name as "Display name". Send's the Vendor's "Company Name" into Coupa. Coupa sets this to the "Display Name" of new Supplier.	_vendor_use_display_name	Free-Form Text	<Company Name>	custentity_use_company_name	

Script Definition					Script Deployment	
Label	Example	Id	Type	Comments	Example	Your Value
Phone/Fax Format	In General Preferences under NetSuite Setup, Phone Number format should be selected. Same format should be selected for the script to process any phone number fields with respect to the format.	_vendor_phonelfaxformat	List/Record	customlist should be created as vendor_phonelfax_list and below values to be assigned 123 456 7890 123-456-7890 123.456.7890	123.456.7890	
Invoice Matching Level	Invoice matching level that is defined for this supplier.	_vendor_invoicematchlevel	Free-Form Text	If no parameter is defined 2-way match will default for all records. 2-way or 3-way or :NetSuite_field_id	2-way	
Payment Method	Payment method	_vendor_paymentmethod	Free-Form Text	if there's no default value, please set it as :customfield or :existingfieldname from where the script will get the value. Value can be invoice or coupa_pay	invoice	
Invoice Emails	Registered email addresses allowed to send invoices	_vendor_invoice_emails	Free-Form Text	existing field or customfield. Multiple email addresses in this existing field or customfield should be delimited by , (comma)	_invoice_emails	

Script Definition					Script Deployment	
Label	Example	Id	Type	Comments	Example	Your Value
Send Invoice To Approvals	If yes, then invoices will all be sent thru approvals, regardless of total amount.	_vendor_sendinvoices_to_approv	Free-Form Text	the value could be default:customfield if there's no default value, please set it as :customfield or :existingfieldname from where the script will get the value. if there's only default, enter only default value	Yes	
Include in Coupa	Configuration that allows/disallows a supplier record to be sent to Coupa	_vendor_customfieldincludeonly	Free-Form Text	Internal id of an existing field or custom field. The custom field on the vendor record should be a checkbox	custentity_coupa_Include	
Allow Invoicing From CSN	If yes, then the supplier can create invoices against their POs or Contracts in CSN	_vendor_allowinvoicing_from_csn	Free-Form Text	the value could be default:customfield if there's no default value, please set it as :customfield or :existingfieldname from where the script will get the value. if there's only default, enter only default value	Yes:	
Content Group Field	Allows the supplier record in Coupa to be associated with content group(s)	_vendor_contentgroup_field	Free-Form Text	the value should be the internal id of the field that holds the name of the content field. Multiple content groups are delimited by , (comma)	_coupa_content_group	
PO Email Override	Allows overriding the standard mapping for the PO Email field	_vendor_poemailoverride	Free-Form Text	the value should be the internal id of the field that holds the PO emails. Multiple POs can be delimited by , (comma)	_coupa_poemail	
Custom Fields Count*	Define the number of custom fields to be integrated.	_vendor_customfieldscount	Integer Number	0,1,2,...	0	

Script Definition					Script Deployment	
Label	Example	Id	Type	Comments	Example	Your Value
Custom Field x (where x is an integer)	custom field of coupa gets syncup with custom field of netsuite. Note that no data transformation allowed	_vendor_customfieldx (where x is an integer)	Free-Form Text	CoupacustomfieldId:N etSuitecustomfield:Co upaFieldType:NetsuiteFieldType:default	eligible-1099:is1099eligible:Boolean:Boolean	
Content Group Mapping	Map subsidiary to Coupa Content Group	custscript_vendor_content_grp_mapping	Free-Form Text	Added 6.1.2	NetSuite_Subsidiary_Name1==Coupa_Content_Group1; NetSuite_Subsidiary_Name2==Coupa_Content_Group2	
Vendor Subsidiary mapping	Map all subsidiaries to ContentGroup or a custom field	custscript_vendor_subs_to_cust_field	Free-Form Text	Added 7.0.0		
*Skip Phone Number	Skipping the phone number on integration	_vendor_skipphonenum	Check Box		T or F	
*Po Method Override	Sync PO Method from NS to Coupa	_vendor_pomethodoverride	List/record		Prompt/email	
*Coupa Supplier ID	Sync subsidiary into Coupa	_vendor_id_field	FreeForm		245	
*Subsidiaries	Sync subsidiaries to Coupa	_vendor_subs_to_cust_field	FreeForm			
*Content Groups	Sync Content Groups	_vendor_contentgroup_field	FreeForm			

2

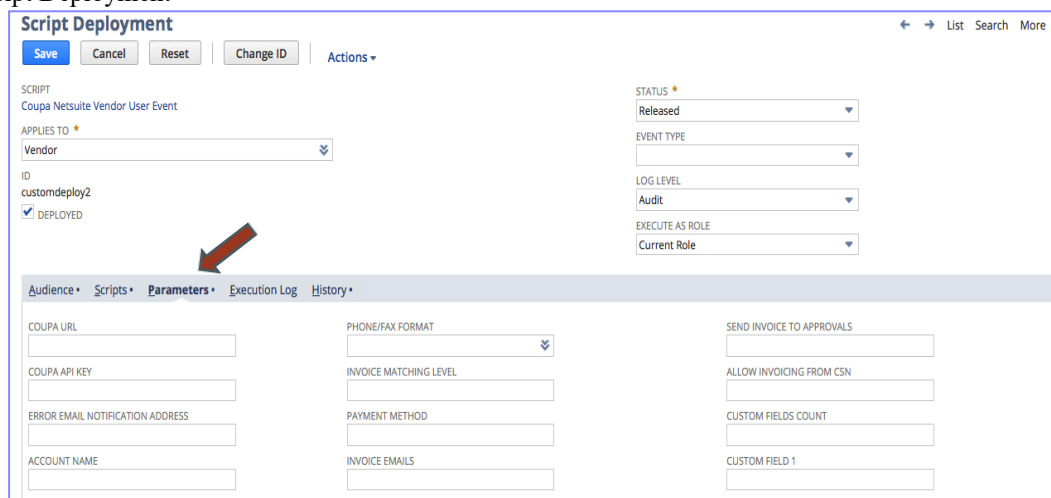
Find and Define Deployment Parameter Values

Navigation Path: Customization | Scripting | Scripts
Find the “Coupa Vendor Integration” script

Now you can click the Parameter menu item and start entering your specific values, please note your specific parameter values may vary based on your Coupa implementation.

Please refer back to the parameters table for guidance.

Script Deployment



Script Deployment

Save Cancel Reset Change ID Actions

SCRIPT
Coupa Netsuite Vendor User Event

APPLIES TO
Vendor

ID
customdeploy2

☒ DEPLOYED

STATUS
Released

EVENT TYPE

LOG LEVEL
Audit

EXECUTE AS ROLE
Current Role

Audience • Scripts • **Parameters** • Execution Log • History •

COUPA URL

COUPA API KEY

ERROR EMAIL NOTIFICATION ADDRESS

ACCOUNT NAME

PHONE/FAX FORMAT

INVOICE MATCHING LEVEL

PAYMENT METHOD

INVOICE EMAILS

SEND INVOICE TO APPROVALS

ALLOW INVOICING FROM CSN

CUSTOM FIELDS COUNT

CUSTOM FIELD 1

After you are done you can click the Save button and now the Script is successfully deployed.

Coupa SIM Supplier to NetSuite Vendor

Overview

If you have purchased the SIM module, you will be able to create new suppliers in NetSuite via Coupa SIM integration. You will also be able to update them via integration. This integration allows you to standardize your supplier onboarding and require much of the vendor information to be provided directly by the supplier. Once provided, the information will be able to flow into NetSuite and create a new vendor or update it. Once the vendor is created in NetSuite, the Netsuite Vendor to Coupa Supplier activates the vendor in Coupa and updates the Coupa record with the SIM form information.

Script Sequence Explained

This integration is scheduled as the Coupa SIM request triggers the creation of a pending supplier in Coupa which is only available for transactions after successfully creating a Vendor in Netsuite.

- The SIM request goes to the supplier
- The Supplier fills out the requested information and submits it back to your Coupa system
- The final SIM information is sync'd into Netsuite to create a new Netsuite Vendor record.
- The Coupa API call to update the Coupa supplier record is called as soon as the Netsuite vendor is updated which activates it in Coupa.
- The `nlapiLoadRecord('vendor', recordid)` function is used to read the Netsuite vendor record
- The `nlapiRequestURL` function is used to do a HTTP POST/PUT to update the supplier record in Coupa.

Tips to help you succeed

- Ensure you have purchased SIM
- Ensure you have created your SIM Forms for New Supplier, External Supplier and Internal Supplier updates.
- Create your SIM Sequential Form Mappings or your previously created NetSuite suppliers, trigger the information request using the Supplier list in Coupa.

Pre-Requisites

- Coupa P2P & Expenses Bundle deployed
- Coupa API key or OIDC auth created with custom permissions

Deployment Parameter Setup

Now that you have successfully installed the bundle, you will need to decide which parameters are relevant for your Coupa implementation.

Activities



You will need to do the following to ensure a successful deployment of the Coupa SIM Integration Script

1 Review Parameter List

2 Find and Define Deployment Parameter Values

Review Parameter List

Review parameter list and decide which ones are relevant to your company's implementation of Coupa. Please note the fields that are mandatory are noted with an asterisk *

Script parameters

Label	Description	Id	Type	Example	Comment
Coupa URL*	The Coupa instance the integration connects to	_coupa_sim_url	Free-	https://abc.coupahost.com	
Coupa API Key*	The API key for Coupa's REST based API	_coupa_sim_api_key	Free-Form Text	4dtyr7657fgfh5757fgfhg Note: Don't forget to edit the permissions of the key, because the default API Key permissions did not yet include SIM as of R20	
Vendor Mapping	Defines the mappings between coupa SIM custom fields and netsuite custom fields for Vendor. In the example to the right, we want to make sure every new vendor is setup as "email" po method, so instead of using the default SIM PO Method, we create a custom field lookup, and only include email as the default lookup choice. Note with this example we are also mapping the supplier's default commodity to ensure that new requisitions will default the commodity code for us based on the supplier.	_coupa_sim_vendor_mapping	Free-Form Text	po-transmission-method/external-ref-num==custen_tity_coupa_po_method_rec;default-commodity/id==custen_tity_coupa_default_commodity	
Address Mapping	Defines the mappings between coupa SIM custom fields and netsuite custom fields for Addresses	_coupa_sim_address_mapping	Free-Form Text	Coupacustomfield1==NScustomfield1;Coupacustomfield2/id==NScustomfield2	
Bank Rec Mapping	Defines the mappings between coupa SIM custom fields and netsuite Bank record fields fields	_coupa_sim_bank_mapping	Free-Form Text	Coupacustomfield1==NScustomfield1;Coupacustomfield2/id==NScustomfield2	
Contact Rec Mapping	Defines the mappings between coupa SIM custom fields and netsuite contact fields	_coupa_sim_contact_mapping	Free-Form Text	Coupacustomfield1==NScustomfield1;Coupacustomfield2/id==NScustomfield2	
Default Subsidiary ID	Defines the default subsidiary ID if subsidiary is not mapped using	_coupa_sim_default_sub	Free-Form	i.e. 5	

Label	Description	Id	Type	Example	Comment
	custom below		Text		
Custom Subsidiary Coupa ID	Defines Coupa SIM custom field which houses the NetSuite Subsidiary. Create a lookup in Coupa, load the subsidiaries and their Netsuite IDs, and then create a custom field on the SIM record. This will define the primary subsidiary in the netsuite vendor record.	_coupa_sim_custom_sub	Free-Form Text	subsidiary/external-ref-num	
“TO” Email Addresses for errors	Where to email errors in this integration	_coupa_sim_error_to	Free-Form Text	tim@gogogo.com	
“FROM” Email address for errors	Who will be displayed as the sender in those error emails	_coupa_sim_error_from	Free-Form Text	tim@gogogo.com	
Vendor “INCLUDE IN COUPA” Parameter ID	This allows the SIM interface to check the box on the new Netsuite Vendor record which causes the vendor integration to include this new vendor in future integrations.	_coupa_sim_vendor_include	Free-Form Text	custentity_vend_include_in_coupa	
Payment File Format ID	This is the coupa custom field which houses the custrecord_2663_entity_file_format for future payments to this vendor. Newly created bank details will be marked as Primary, and the existing primary record will be demoted.	_coupa_sim_payment_format	Free-Form Text	netsuite-payment-file-format/external-ref-num	
Vendor Integration PO METHOD Parameter	This is where to store the PO Method.	_coupa_po_method_parameter	Free-Form Text	Netsuitecustomfield1	
Use Display Name as Company Name	This will override the Supplier Name with the whatever was specified in the “Display Name”	_coupa_sim_display_name	Free-Form Text	<Checkbox>	
Multiselect Subsidiary	Add support for Multi subsidiary on vendor recording in NS	custscript_coupa_sim_sub_multiselect	Check box		Added 6.0.1
*Skip bank Account Details	Skip integrating bank account details	_coupa_sim_bank_syncs	Free Form Text	T/F	Updated 10.0.0
*Custom Bank Module	AvidXBank support	_sim_custom_bank	CheckBox	T/F	

2

Find and Define Deployment Parameter Values

Navigation Path: Customization | Scripting | Scripts
Find the “Coupa SIM Integration” script deployment



Now you can click the Parameter menu item and start entering your specific values, please note your specific parameter values may vary based on your Coupa implementation.

Please refer back to the parameters table for guidance.

Script Deployment

Save

Cancel

Reset

Change ID

Actions

SCRIPT

Coupa SIM Integration

TITLE *

Coupa SIM Integration

ID

customdeploy_coupa_sim_scheduled

☒ DEPLOYED

STATUS *

Testing

SEE INSTANCES

Status Page

LOG LEVEL

Debug

EXECUTE AS ROLE

Administrator

PRIORITY *

Standard

CONCURRENCY LIMIT

1

☒ SUBMIT ALL STAGES AT ONCE

YIELD AFTER MINUTES *

60

BUFFER SIZE *

1

ScheduleParametersExecution LogSystem Notes

COUPA URL

COUPA API KEY

VENDOR MAPPING

ADDRESS MAPPING

BANK REC MAPPING

CONTACT REC MAPPING

DEFAULT SUBSIDIARY ID

CUSTOM SUBSIDIARY COUPA ID

"TO" EMAIL ADDRESSES FOR ERRORS

"FROM" EMAIL ADDRESS FOR ERRORS

VENDOR "INCLUDE IN COUPA" PARAMETER ID

PAYMENT FILE FORMAT ID

VENDOR INTEGRATION PO METHOD PARAMETER

☒ USE DISPLAY NAME AS COMPANY NAME?

After you are done you can click the Save button and now the Script is successfully deployed.



Special Considerations with the SIM Bundle

- You must have a Primary Contact with a First Name, Last Name, and Email address, in the NetSuite Vendor or you will not be able to trigger an external SIM update form from the Coupa supplier record. This is especially applicable for initial production cutover when you integrate all of your Netsuite Vendors into Coupa expecting to phase them into SIM over time.
- If you allow a supplier or internal user to set the default Commodity Code in SIM, it will integrate over to Netsuite, but it will not integrate back into Coupa from the Netsuite Vendor record without modifying the Vendor Suite Script.



Netsuite Exchange Rate to Coupa

Exchange Rates 2.0

Overview

This integration syncs up Currency Exchange Rates between Netsuite and Coupa. NetSuite is the Exchange Rates master and the records flow from Netsuite into Coupa. The Scheduled SuiteScript is used to capture the “create” events and that will call the Coupa POST API to create the new entry. It will also create the inverse entry in Coupa.

From Bundle Release 6.0.1 Exchange Rate script 2.0 is introduced. Please upgrade to use this new script.

Exchange Rate script 1.0 will not be supported from 8.0.0.

Script Sequence Explained

This integration is scheduled per your requirements to support the Currency Exchange Rates record in Netsuite

- The Coupa API call to create the Currency Exchange Rates record is called on a predefined schedule
- The `nlapiLoadRecord(currency)` function is used to read the Netsuite currency record
- The `nlapiRequestURL` function is used to do a HTTP PUT to create the corresponding Exchange Rate records in Coupa

Activities

You will need to do the following to ensure a successful deployment of the Coupa Vendor Integration Script

1

Review Parameters

2

Find and Define Deployment Parameter Values

Tips to help you succeed

- Ensure your currency codes in Coupa are enabled
- Only load exchange rates in Coupa that you will create transactions for

Pre-Requisites

- Coupa P2P & Expenses Bundle scheduled
- Coupa API key or OIDC Auth created

Review Parameter List

Review parameter list and decide which ones are relevant to your company's implementation of Coupa. Please note the fields that are mandatory are noted with an asterisk *

Script parameters

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your Value
Coupa URL*	The Coupa instance the integration connects to	_coupa_fxrates_url	Free-Form Text		https://abc.coupahost.com	
Coupa API Key*	The API key for Coupa's REST based API	_coupa_fxrates_apikey	Free-Form Text	A Coupa Admin can generate the Coupa API Key in Coupa https://xxxx.coupahost.com/api_keys	4dtyr7657fgfh5757fghhg	
Error Email Notification Address	Email addresses to which exception emails to be sent	_coupa_fxrates_erroremailnotify	Free-Form Text		abc@abc.com , abcd@abcd.com	
Account Name*	As per standard, it will be company name with which environment its been setup (Test or Prod)	_coupa_fxrates_accountname	Free-Form Text	<Company Name> <Test/Prod>	Coupa Test	
UTC Offset	The <i>UTC offset</i> is the difference in hours and minutes from Coordinated Universal Time	_coupa_fxrates_utcoffset	Integer Number		10	
Error Email From	Error Email sender	custscript_coupa_fxrate_mr_error_from	Free-Form Text			Added P2P 6.0.1 and P2O 2.0.0
Error Email to	Error Email Recipient	custscript_coupa_fxrate_mr_error_to	Free-Form Text			Added P2P 6.0.1 and P2O 2.0.0

1

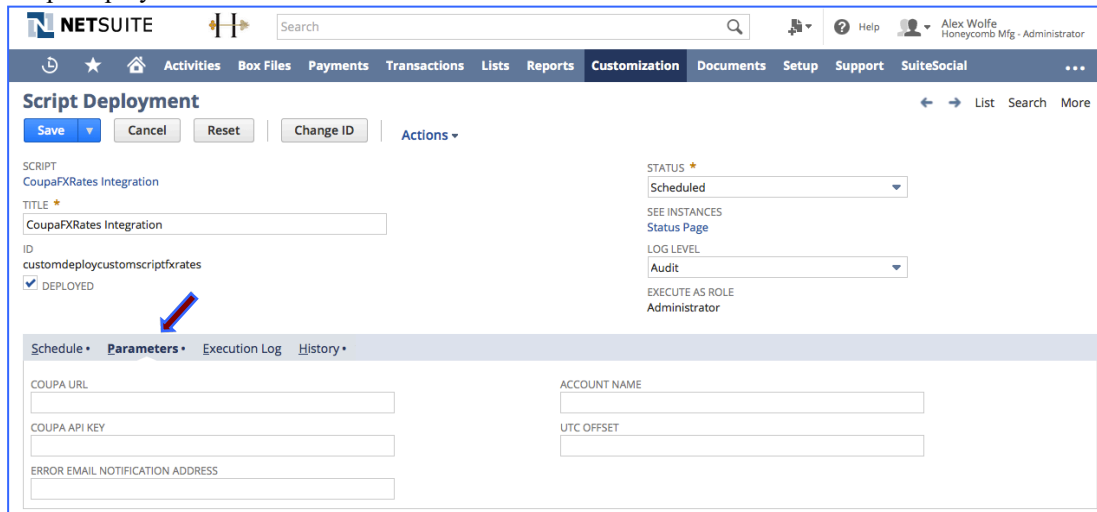
Find and Define Deployment Parameter Values

Navigation Path: Customization | Scripting | Scripts
Find the “Coupa Exchange Rate Integration” script

Now you can click the Parameter menu item and start entering your specific values, please note your specific parameter values may vary based on your Coupa implementation.

Please refer back to the parameters table for guidance.

Script Deployment



NETSUITE Search

Activities Box Files Payments Transactions Lists Reports Customization Documents Setup Support SuiteSocial

Script Deployment

Save Cancel Reset Change ID Actions

SCRIPT
CoupaFXRates Integration

TITLE
CoupaFXRates Integration

ID
customdeploycustomscriptfxrates

DEPLOYED

STATUS
Scheduled

SEE INSTANCES
Status Page

LOG LEVEL
Audit

EXECUTE AS ROLE
Administrator

Schedule Parameters Execution Log History

COUPA URL

COUPA API KEY

ERROR EMAIL NOTIFICATION ADDRESS

ACCOUNT NAME

UTC OFFSET

After you are done you can click the Save button and now the Script is successfully deployed

Dynamic - Coupa Chart of Account to Coupa

Overview

This integration syncs up key parts of your Netsuite Chart of Account (COA) definition between Netsuite and the Chart of Accounts in Coupa. Netsuite is the COA master and the valid values for each segment flow from Netsuite into Coupa. The User Event Suitescript is used to capture the “create” and “update” events. Based on the event type either a new value is created in Coupa or an existing entry in Coupa is updated.

Script Sequence Explained

This integration is real time as the Netsuite Chart of Account record is created/updated it will then be reflected in Coupa

- The Coupa API calls to create/update the LookUp Values in Coupa as soon as the Netsuite user event is captured
- The `nlapiLoadRecord('vendor', recordid)` function is used to read the Netsuite vendor record
- The `nlapiRequestURL` function is used to do a HTTP POST/PUT to create/update the supplier record in Coupa.

Activities

You will need to do the following to ensure a successful deployment of the Coupa Vendor Integration Script

- 1 Define Coupa Lookups and Coupa Chart of Accounts**
- 2 Review Parameters**
- 3 Review Scripts**
- 4 Find and Define Deployment Parameter Values**

Tips to help you succeed

- Each Subsidiary can be a separate Chart of Account in Coupa if you require granularity based on your Netsuite COA setup
- The delivered script will create Lookup Values in Coupa for the following: Subsidiary, Class, Account, Department, Location
- If you need segments based on other values than above, Coupa can support that with integration via our lookup tables. Please ask your implementation consultant.

Pre-Requisites

- Coupa P2P & Expenses Bundle deployed
- Coupa API key or OIDC Auth created
- Lookups objects & COA defined in Coupa
- You will need to baseline the Coupa Lookup Values via the CSV UI Loader
- Add “Include in Coupa” Checkbox custom field in Netsuite, on the appropriate objects you are using as part of your Netsuite COA

Define Coupa Lookups and Coupa Chart of Accounts

Summary:

The dynamic accounting script, transforms creates, and edits on departments, classes, locations, accounts and subsidiaries into Lookup Values in Coupa. Marking a record in Netsuite as inactive, or deselecting the Include In Coupa checkbox will mark the Lookup Value in Coupa as inactive. Name and hierarchical changes to all records (except Subsidiary) will be reflected in Coupa by a name change to the Lookup Value.

Configuration Decisions:

Define your Coupa Chart of Accounts based on the number of Netsuite subsidiaries that you have.

Your Chart of Account name must equal the NetSuite Subsidiary name

In Netsuite Dynamic Accounting Checkbox turned on in the deployment parameters of this script

Delete the COA related deployment objects that are not relevant for your implementation.

You will need to have the “Include In Coupa” checkbox created for each COA related object

Subsidiary Account Segment Parameter will be Segment-1, Segment-2, etc...based on your Coupa COA definition

Naming / Formatting Conventions:

Based on your implementation define the Coupa Lookups with the following naming convention, where XYZ is the name of your Coupa Chart of Account

acct_XYZ

dept_XYZ

clas_XYZ

locn_XYZ

subs_XYZ




On the Coupa Lookup Object the following must be adhered to ensure referential integrity between Coupa and Netsuite:

Accounts, the account number must be the leading part of the name and the external reference number. Please see below for example:













Lookup Values								
<div> Create Load from file Export to View: All Advanced 4000 sales </div>								
Showing results for 4000 sales x								
Lookup	Name	Description	External Ref Num	Default	Active	External Ref Code	Parent Name	Actions
acct_Square, Inc.	4000 Sales	None	4000	No	Yes	03849314000	038493 Testing Expense C	
acct_Looper Holdings LLC	4000 Sales	None	4000	None	No	4000	None	
acct_Square K.K. (Japan)	4000 Sales	None	4000	None	No	4000	None	
acct_Squareup Pte. Ltd.	4000 Sales	None	4000	None	No	4000	None	
acct_[Square, Inc. Eliminations]	4000 Sales	None	4000	None	No	4000	None	
Per page 15 45 90								

- For all other look up values, the name can be anything but the Netsuite Internal ID must be in the external reference number. Please see below for example:

Lookup Values

Create Load from file Export to View All Advanced dept_americas								
Showing results for dept_americas x								
Lookup	Name	Description	External Ref Num	Default	Active	External Ref Code	Parent Name	Actions
dept_AMERICAS	Sales	None	3	None	Yes	3	None	
dept_AMERICAS	Engineering : Development : Integrations	None	15	None	Yes	15	None	
dept_AMERICAS	Sales : Test notinclude on create	None	16	None	Yes	16	None	
Per page 15 45 90								

- Examples:
 - [Multiple Coupa Chart of Accounts](#): that has 3 Netsuite subsidiaries named US, UK and CA; all utilizing account, department and class.
- Lookups defined in Coupa:

Expenses Requests Orders Invoices Inventory Sourcing Contracts Reports Setup			
Lookups			
Create Load from file Export to View NS Advanced			
Name	Description	Active	Actions
acct_US	acct_US	Yes	
acct_UK	acct_UK	Yes	
acct_CA	acct_CA	Yes	
dept_US	dept_US	Yes	
dept_UK	dept_UK	Yes	
dept_CA	dept_CA	Yes	
clas_US	clas_US	Yes	
clas_UK	clas_UK	Yes	
clas_CA	clas_CA	Yes	
subs_US	subs_US	Yes	
subs_UK	subs_UK	Yes	
subs_CA	subs_CA	Yes	
Per page 15 45 90			

Three Chart of Accounts would be created in Coupa. Here is the US example:

Chart Of Accounts Create

Name

US

Currency

USD

Account Segments and Defaults

I'll load all of the valid combinations

I'll define the values segment by segment

	Purchase request defaults	Expense report defaults
<div>Segment 1</div> <div>subs</div>	<div>default by</div> <div>None</div>	<div>default by</div> <div>None</div>
	<div>This segment is a simple list defined in</div> <div>subs_US</div>	
<div>Segment 2</div> <div>dept</div>	<div>default by</div> <div>None</div>	<div>default by</div> <div>None</div>
	<div>This segment is a simple list defined in</div> <div>dept_US</div>	
	<div>This segment depends on</div> <div></div>	
	<div>This segment is required</div> <div></div>	
<div>Segment 3</div> <div>class</div>	<div>default by</div> <div>None</div>	<div>default by</div> <div>None</div>
	<div>This segment is a simple list defined in</div> <div>clas_US</div>	
	<div>This segment depends on</div> <div></div>	
	<div>This segment is required</div> <div></div>	
<div>Segment 4</div> <div>acct</div>	<div>default by</div> <div>None</div>	<div>default by</div> <div>None</div>
	<div>This segment is a simple list defined in</div> <div>acct_US</div>	
	<div>This segment depends on</div> <div></div>	
	<div>This segment is required</div> <div></div>	

Review Parameter List

Review parameter list and decide which ones are relevant to your company's implementation of Coupa.

Please note the fields that are mandatory are noted with an asterisk *

Script parameters

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your Value
Coupa URL*	The Coupa instance the integration connects to	_coupa_accs_url	Free-Form Text		https://abc.coupahost.com	
Coupa API Key*	The API key for Coupa's REST based API	_coupa_accs_apikey	Free-Form Text	A Coupa Admin can generate the Coupa API Key in Coupa https://xxxx.coupahost.com/api_keys	4dtyr7657fgfh5757fghhg	
Email Addresses for Notif	Email addresses to which exception emails to be sent	_coupa_accs_notify_email	Free-Form Text		abc@abc.com , abcd@abcd.com	
Include in Coupa	Id of the custom field (on the record this deployment applies to) to allow/disallow the transmission of the record to Coupa	_coupa_accs_cstmfieldincludeonly	Free-Form Text	The Internal ID of an existing field or custom field. This field should be a checkbox, and True if you wish the record to integrate	custrecord_include_loc_coupa	
Subsid Include in Coupa	Id of the custom field (on SUBSIDIARY records) to allow/disallow the transmission of the subsidiary to Coupa	_coupa_accs_dynamic_sub_included	Free-Form Text	The internal ID of an existing field or custom field on the subsidiary record. This field should be a checkbox and set to True if the subsidiary should be/is included in coupa	custrecord_include_subs_coupa	
*Use External ID	Allow using external ID from NS	_coupa_accs_extended	Check Box		T/F	
*Custom Segment Name	Applicable for custom segments	_coupa_accs_namefield	FreeForm		Name	

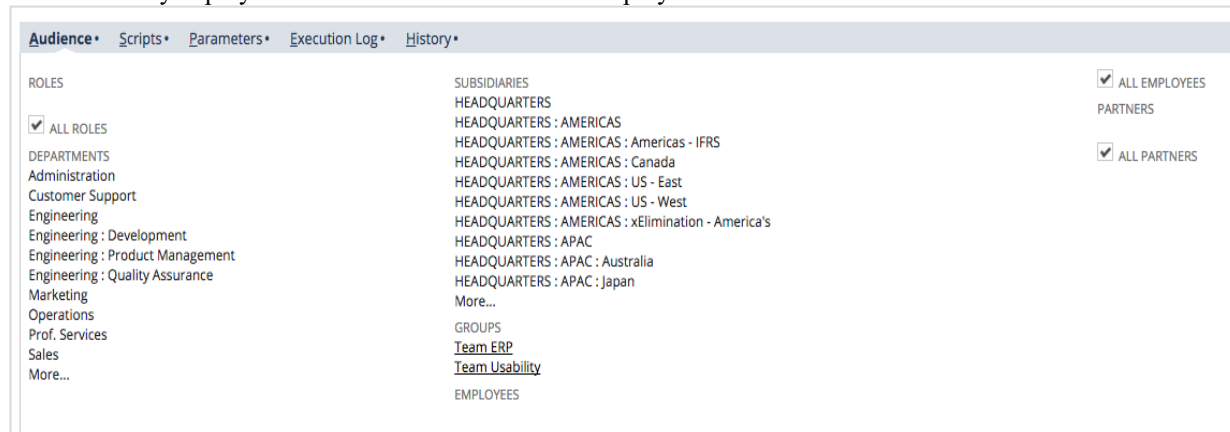
Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your Value
*Parent ID	Parent lookup values will be integrated	_parent_field_id	Freeform			
*Non One World System?	Support customer with non One World system	_coupa_accs_one_subsiidiary	Check Box		T/F	
*Custom Lookup name Prefix	Custom segment lookup name prefix	_lookup_name_prefix	Freeform			
*Custom Segment Subsidiary Field	Custom segment subsidiary mapping	_coupa_accs_subfield	FreeForm			

1

Review Scripts

The bundle creates deployments for each possible segment. You need to delete/remove deployments for records you do not want to integrate

Make sure they deployments are enabled for all of the employees and subsidiaries



Audience • Scripts • Parameters • Execution Log • History

ROLES

- ☒ ALL ROLES
- DEPARTMENTS
 - Administration
 - Customer Support
 - Engineering
 - Engineering : Development
 - Engineering : Product Management
 - Engineering : Quality Assurance
 - Marketing
 - Operations
 - Prof. Services
 - Sales
 - More...

SUBSIDIARIES

- HEADQUARTERS
- HEADQUARTERS : AMERICAS
- HEADQUARTERS : AMERICAS : Americas - IFRS
- HEADQUARTERS : AMERICAS : Canada
- HEADQUARTERS : AMERICAS : US - East
- HEADQUARTERS : AMERICAS : US - West
- HEADQUARTERS : AMERICAS : xElimination - America's
- HEADQUARTERS : APAC
- HEADQUARTERS : APAC : Australia
- HEADQUARTERS : APAC : Japan
- More...
- GROUPS
 - [Team ERP](#)
 - [Team Usability](#)
- EMPLOYEES

☒ ALL EMPLOYEES

☐ PARTNERS

☒ ALL PARTNERS

The bundle creates deployments for each possible segment. You need to delete/remove deployments for records you do not want to integrate

2

Find and Define Deployment Parameter Values

Navigation Path: Customization | Scripting | Scripts

Find the “Coupa Dynamic COA Integration” script

Coupa Netsuite Integration Guide - Last updated : 2022-12-06 – Bundle Release 11.0.0



Now you can click the Parameter menu item and start entering your specific values, please note your specific parameter values may vary based on your Coupa implementation.

Please refer back to the parameters table for guidance.

Script Deployment

Save

Cancel

Reset

Change ID

Actions ▾

SCRIPT

Coupa Dynamic COA Integration

APPLIES TO *

Account ▾

ID

customdeploy_coupa_accs_account

☒ DEPLOYED

STATUS *

Released ▾

EVENT TYPE

▾

LOG LEVEL

Debug ▾

EXECUTE AS ROLE

Current Role ▾

Audience •

Scripts •

Parameters •

Execution Log

History •

COUPA API KEY

COUPA URL

INCLUDE IN COUPA FIELD

SUBSIDIARY INCLUDE IN COUPA FIELD

ERROR NOTIFICATION EMAIL(S)

After you are done you can click the Save button and now the Script is successfully deployed



Coupa Invoice 1.0 & 2.0 to Netsuite Vendor Bill

Overview

This integration syncs up Coupa Invoices and Netsuite Vendor Bill via a scheduled SuiteScript. The Coupa Invoice once approved is set for export and now is ready to be imported into NetSuite as the Vendor Bill. The Scheduled SuiteScript is used to “create” the Vendor Bill which will call the Coupa HTTP GET API to retrieve the Coupa Invoice to generate the Vendor Bill in NetSuite

Coupa Invoice 1.0 Script is decommissioned since 7.0.0, please upgrade to Coupa Invoice 2.0 Script

Script Sequence Explained

This integration is on a schedule that you can set to meet your frequency and volume as Coupa Invoices are approved.

- The Coupa API call to create the Vendor Bill record is called on a predefined schedule
- The nlapiRequestURL function is used to perform a HTTP GET to retrieve the Coupa Invoice that is fully approved and not exported
- The nlapiCreateRecord (vendorbill) function is used to generate the NetSuite Vendor Bill

Activities

You will need to do the following to ensure a successful deployment of the Vendor Script

- 1 Review Parameter List**
- 2 Find and Define Deployment Parameter Values**

Tips to help you succeed

- Ensure your NetSuite Vendors are assigned appropriate Subsidiaries in NetSuite
- This integration requires the Netsuite internal id for the Vendor Record to be stored in the “Supplier Number” field in Coupa
- Validate the Vendor Bill “isBillable” field is populating as expected
- If you need to pass Amortization Schedules please see parameter definition
- Voids are supported from Coupa to Netsuite assuming the Vendor Bill is not yet paid in Netsuite. Change the status of the invoice from "Approved" to "Rejected". See parameter definition for more details.

Pre-Requisites

- Coupa P2P & Expenses Bundle deployed
- Coupa API key or OIDC Auth created
- Coupa Chart of Account SetupCoupa Suppliers with Netsuite Vendor ID

Review Parameter List – Invoice Script 1.0

Review parameter list and decide which ones are relevant to your company's implementation of Coupa.

Please note the fields that are mandatory are noted with an asterisk *

Script parameters

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your Value
Coupa URL*	The Coupa instance the integration connects to	_coupa_inv_url	Free-Form Text		https://abc.coupa.com	
Coupa API Key*	The API key for Coupa's REST based API	_coupa_inv_apikey	Free-Form Text	A Coupa Admin can generate the Coupa API Key in Coupa https://xxxx.coupa.com/api/keys	4dtyr7657fgfh5757fghhg	
GL Account Segment	The accounting segment in Coupa that holds the Netsuite GL account information	_coupa_inv_glactseg	Free-Form Text	Ensure that your GL Account # uses this format in your Coupa Chart of Accounts: NetsuiteGLAccountNumber:NetsuiteGLAccountName	segment-2	
Department Segment	The accounting segment in Coupa that holds the Netsuite department account information	_coupa_inv_deptseg	Free-Form Text		segment-3	
Subsidiary Account segment	The accounting segment in Coupa that holds the Netsuite subsidiary account information	_coupa_inv_subseg	Free-Form Text		segment-1	
Class Segment	The accounting segment in Coupa that holds the Netsuite classification account information	_coupa_inv_classseg	Free-Form Text		segment-4	
Location Segment	The accounting segment in Coupa that holds the Netsuite location information	_coupa_inv_locseg	Free-Form Text		segment-5	
Export From Invoice date	Add filter criteria to pull invoices from Coupa with Invoice date >= to this date	_coupa_inv_frominvdate	Date		12/14/2013	
Export To Invoice date	Add filter criteria to pull invoices from Coupa with Invoice date <= to this date	_coupa_inv_toinvdate	Date		12/27/2013	

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your Value
Use Updated at date	This will override the filter criteria of 'Invoice date' and use the 'Updated at' date filter criteria	_coupa_inv_use_updatedat_date	Check box		'T' i.e. checked	
Export From 'Updated at' date	Add filter criteria to pull invoices from Coupa with 'Updated at' date >= to this date	_coupa_inv_from_updatedat_date	Date		12/14/2013	
Export To 'Updated at' date	Add filter criteria to pull invoices from Coupa with 'Updated at' date <= to this date	_coupa_inv_to_updatedat_date	Date		12/27/2013	
Accounts Payable account number	The Netsuite AP account number	_coupa_inv_actpayablenum	Free-Form Text		210000	
notify this email address	email addresses alerts to be sent to	_coupa_inv_email_notifications	Free-Form Text	multiple addresses can be separated by comma ','	xyz1@abc.com , xyz2@abc.com	
Account Name	Used in the alert/notification email subject to identify which Netsuite instance the alerts are coming from.	_coupa_inv_accountname	Free-Form Text		ABC Production	
Send Tax Code	Option to enable tax code being sent from Coupa to Netsuite so that Netsuite calculates tax. When this option is selected tax amount will NOT be added to the vendor bill expense line amount. Also when this option is selected, there will be an additional line per expense line for header level charges.	_coupa_inv_send_taxcode	checkbox	When using this option, the taxcodes in Coupa need to be stored in the following format: netsuitename:net suiteinternalid. E.g. if you have a taxcode in Netsuite VAT21 with internal id of 46 then in Coupa the taxcode should be created as VAT21:46 Additionally, this will not support Multiple Tax Lines in Coupa as there is only one place to set a Tax Code in NetSuite. Tax codes are also only applicable for NON-US Subsidiaries	T' i.e. checked	
Invoice URL link	URL to Invoice	_coupa_inv_link_field	Free-Form Text		http://<invoice URL>.com	

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your Value
Invoice Image URL Link	URL to invoice image	_coupa_inv_image_link_field	Free-Form Text		http://<invoiceimage>.com	
Coupa Version 12.06+?	Version 12.06 introduces credit memos. Enable this checkbox if instance is on a release above 12.06	_coupa_inv_documenttype	checkbox		T' i.e. checked	
Dynamic Accounting ?	Allows for billing strings to be created dynamically.	_coupa_inv_dynamicaccts	checkbox		T' i.e. checked	
Limit exported records to	Configuration to limit the invoices per invoice run	_coupa_inv_limit	Integer Number		5	
Posting period cutoff	Configuration for posting period.	_coupa_inv_cutoff_date	Integer Number	If the integration routine date >= cutoff day then posting period is the month of the routine date. If the integration routine date < cutoff day then If invoice date is current month, posting period is current month and if invoice date is the prior month or earlier, posting period is the prior month.	5	
Custom Field (header) 1	Header level custom fields	_coupa_inv_custfieldheader1	Free-Form Text		See "Common Custom Field Examples" below	
Custom Field (header) 2	Header level custom fields	_coupa_inv_custfieldheader1	Free-Form Text		See "Common Custom Field Examples" below	
Custom Field (header) 3	Header level custom fields	_coupa_inv_custfieldheader1	Free-Form Text		See "Common Custom Field Examples" below	
Custom Field (header) 4	Header level custom fields	_coupa_inv_custfieldheader1	Free-Form Text		See "Common Custom Field Examples" below	
Custom Field (header) 5	Header level custom fields	_coupa_inv_custfieldheader1	Free-Form Text		See "Common Custom Field Examples" below	

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your Value
Custom Field (line) 1	Line level custom fields	_coupa_inv_custfieldline1	Free-Form Text		See “Common Custom Field Examples” below	
Custom Field (line) 2	Line level custom fields	_coupa_inv_custfieldline2	Free-Form Text		See “Common Custom Field Examples” below	
Custom Field (line) 3	Line level custom fields	_coupa_inv_custfieldline3	Free-Form Text		See “Common Custom Field Examples” below	
Custom Field (line) 4	Line level custom fields	_coupa_inv_custfieldline4	Free-Form Text		See “Common Custom Field Examples” below	
Custom Field (line) 5	Line level custom fields	_coupa_inv_custfieldline5	Free-Form Text		See “Common Custom Field Examples” below	
Credit Custom Field 1 (Header)	Header level custom field for credit notes	_custfield_crdt_header1	Free-Form Text		See “Common Custom Field Examples” below	
Credit Custom Field 1 (Line)	Line level custom field for credit notes	_custfield_crdt_line1	Free-Form Text		See “Common Custom Field Examples” below	
CustomField(header) count	Count of the number of header level custom fields configured	_coupa_inv_custfield_header_ct	Integer Number		5	
Custom field (line) count :	Count of the number of line level custom fields configured	_coupa_inv_custfield_line_cnt	Integer Number		3	
Credit Custom Field Count (Header)	Count of the number of header level custom fields configured for Credit Notes	_customfield_crdt_header_count	Integer Number		1	
Credit Custom Field Count (Line)	Count of the number of line level custom fields configured for Credit Notes	_customfield_crdt_line_count	Integer Number		1	
Support Void	Sets the Vendor Bill to void	_supportvoid	Integer Number	Set the value of the parameter to 1 to enable this feature Also in Coupa, when you void the invoice, mark the previously exported invoice as “not exported”	1	

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your Value
Coupa Pay Enabled?	Enables Payment hold for coupa pay invoices	<i>_coupa_inv_coupa_pay_enabled</i>	checkbox	Puts Coupa Pay invoice on Payment hold in Netsuite if set to true		
Skip Bill for VC n PO	Enables to skip Bill/Invoices to integrate to NS for Invoices where payment channel is "CoupaPay - Virtual Card on PO"	<i>ustscript_coupa_inv_skip_bill_for_vc_po</i>	checkbox			
Error Email Sender	Define error email sender	<i>custscript_coupa_inv_error_email_sender</i>	List/record	Added 6.1.2		
Is CoupaPay Supplier?	Used to filter suppliers whose Bill/CN will be synced to Coupa	<i>custentity_coupa_vendor_type</i>	CheckBox	Added 7.0.0		
*Custom Arguments	Add custom arguments to API query	<i>_coupa_invoice_field</i>	FreeForm		&id=256	
*Custom Form Mapping	Use custom Bill from NS on COA while creating the Bill	<i>_coupa_inv_vendor_billForm</i>	FreeForm		Coa_name1-form1:coa_name2-form2	
*User External IDs?	Support external IDs on COA segments	<i>_coupa_inv_extid</i>	Checkbox			

Common Custom Field Examples:

Script Definition					Script Deployment
Label	Description	Type	Comments	Example	Your Value
Amortization Schedule	Pass the amortization schedule defined on the Coupa Invoice	Lookup		amort-schedule:amortizationsched:Lookup:Value	
Amortization Start Date	The date when the amortization should start	Date		amort-start:amortizstartdate:Date:Value	
Amortization End Date	The date when the amortization should end	Date		amort-end:amortizationenddate:Date:Value	
isBillable	Sets the Billable field on the NS Vendor Bill	Text	In Coupa, you would need to create a custom field isbillable on the Requisition Line Item level of type Dropdown . Note that the values have to be 'T' (True) and 'F' (False)	isBillable:isBillable:Text:Value	
Coupa Pay Enabled?	Sets the script behavior for	checkbox	If set to True, Invoice paid in coupa via Coupa Pay will be		

Script Definition					Script Deployment
Label	Description	Type	Comments	Example	Your Value
	Invoice Paid in Coupa via Coupa Pay		put on "Payment Hold" to avoid double payments.		

Review Parameter List Invoice 2.0

Review parameter list and decide which ones are relevant to your company's implementation of Coupa.

Please note the fields that are mandatory are noted with an asterisk *

Script parameters

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your Values
Coupa URL*	URL of the Coupa instance the integratin connects to	custscript_coupa_inv_mr_host_url	Hyperlink			
Coupa API key	The API key for Coupa's REST based API	custscript_coupa_inv_mr_api_key	FreeForm			
Error Email Sender*	The internal id of the email sender employee record	custscript_coupa_inv_mr_email_sender	List/Recor d			
Error Email Receiver*	Comma separated email ids of the error email receiptents	custscript_coupa_inv_mr_email_receive r	eeForm			
Default Accounts Payable account number The	Netsuite AP account number to be set on Vendor bill/ Vendor credit transaction	custscript_coupa_inv_mr_act_payable_num	FreeForm		segment-2	
GL Account segment*	The accounting segment in Coupa that holds the Netsuite GL account information	custscript_coupa_inv_mr_acc_seg	FreeForm		segment-3	

Department segment*	The accounting segment in Coupa that holds the Netsuite department account information	custscript_coupa_inv_mr_dept_seg	FreeForm		segment-5	
Class segment*	Netsuite classification account information	custscript_coupa_inv_mr_class_seg	FreeForm		segment-4	
Subsidiary Segment* The accounting segment in Coupa that holds the	The accounting segment in Coupa that holds the Netsuite subsidiary account information	custscript_coupa_inv_mr_sub_seg	FreeForm		segment-1	
Export From Invoice Date	Add filter criteria to pull invoices from Coupa with Invoice date >= to this date	custscript_coupa_inv_mr_from_inv_date	Date			
Export To Invoice Date Add filter	criteria to pull invoices from Coupa with Invoice date <= to this date	custscript_coupa_inv_mr_to_inv_date	Date			
Export From 'Updated at' date	Add filter criteria to pull invoices from Coupa with 'Updated at' date >= to this date	custscript_coupa_inv_mr_from_upd_date	Date			
Export To 'Updated at' date	Add filter criteria to pull invoices from Coupa with 'Updated at' date <= to this date	custscript_coupa_inv_mr_to_upd_date	Date			
Custom Line Field Mappings	Line level custom field mapping	custscript_coupa_inv_mr_cust_line_fields	Text		customfields.line-textfield==custcol1; 8==TEXT; customfields.prepaidenddate==custcol19==DATE;	

Custom Body Field Mappings	Header level custom field mapping	custscript_coupa_inv_mr_cust_body_fields	TEXT		customfields.testmemo==custbody_coupa_memo==TEXT;customfields.testlookup==custbody_invoice_default==LOOK UP;	
Limit Exported Records To	Configuration to limit the invoices to be imported per invoice run	custscript_coupa_inv_mr_inv_limit	FreeForm		40	
Posting Period Cutoff Day	Configuration for posting period. The posting period can be positive or negative	custscript_coupa_inv_mr_cutoff_day	FreeForm		5 or -3	
Account Name	The account name to be added to error email's subject. If not provided default's from company name	custscript_coupa_inv_mr_account_name	FreeForm			
Invoice URL link	URL to coupa invoice	custscript_coupa_inv_mr_inv_url_field	FreeForm			
Invoice Image URL Link	URL to coupa invoice's image	custscript_coupa_inv_mr_image_url_field	FreeForm			
Custom API Filter	Custom API arguments can be added to filter the api result	custscript_coupa_inv_mr_inv_api_filter	FreeForm			
Use External ID	Use external id for searching record in NS instead of internal id	custscript_coupa_inv_mr_use_ext_id	CheckBox			
Support Void	If enabled marks the vendor bill in NS rejected when the Coupa invoice is voided	custscript_coupa_inv_mr_support_void	CheckBox			

Skip Bill for VC & PO	Enables to skip Bill/Invoices to integrate to NS for Invoices where payment channel is "CoupaPay - Virtual Card on PO"	custscript_coupa_inv_mr_skip_vc_po_bill	CheckBox			
Custom Form	The mapping if provided can be used to select custom forms per coupa COA ID	custscript_coupa_inv_mr_custom_form	FreeForm		38==221	
Use Coupa Charge	Distribution Use the invoice charge distribution logic provided by coupa platform	custscript_coupa_inv_mr_use_coupa_distribution	CheckBox			
Send Tax Code	Option to enable tax code being sent from Coupa to Netsuite so that Netsuite calculates tax. When this option is selected tax amount will NOT be added to the vendor bill expense line amount. Also when this option is selected, there will be an additional line per expense line for header level charges.	custscript_coupa_inv_mr_send_taxcode	CheckBox			
Skip Credit	Note Syncing Skip Credit Note transaction syncing from Coupa	custscript_coupa_inv_mr_skip_credit_note	CheckBox			
Custom Location Field	Source Location from coupa's custom field	custscript_coupa_inv_mr_custom_location	FreeForm			
Use Dynamic Accounting	Allows for billing strings to be created dynamically.	custscript_coupa_inv_mr_use_dynamic_acc	CheckBox			

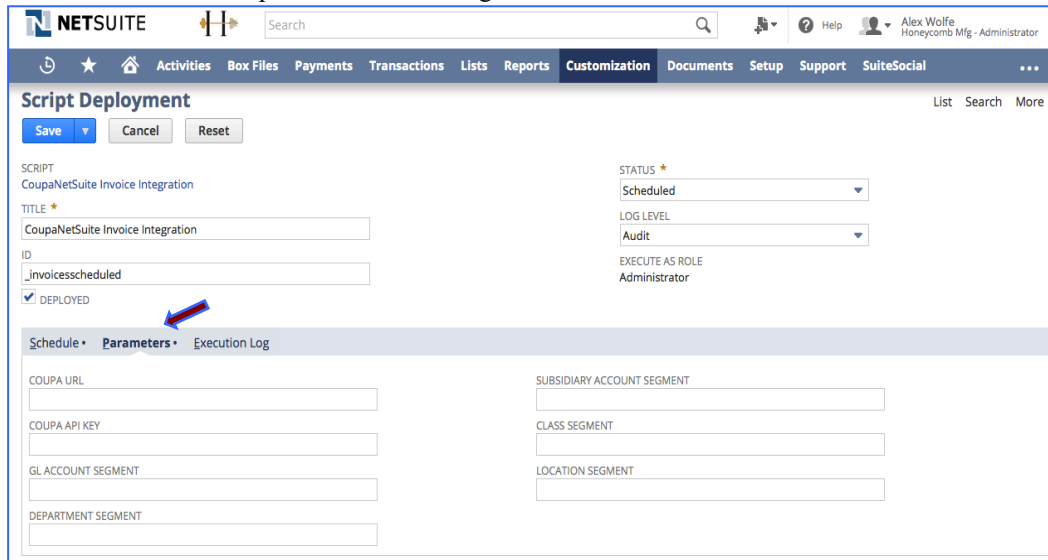
1

Find and Define Deployment Parameter Values

Navigation Path: Customization | Scripting | Scripts
Find the “Coupa invoice Integration” script

Now you can click the Parameter menu item and start entering your specific values, please note your specific parameter values may vary based on your Coupa implementation.

Please refer back to the parameters table for guidance.



The screenshot shows the Netsuite interface for script deployment. The 'Script Deployment' page is open, displaying the 'CoupaNetSuite Invoice Integration' script. The 'Parameters' tab is selected, showing various input fields for configuration. A red arrow points to the 'Parameters' tab.

Script	Status	Log Level	Execute As Role
CoupaNetSuite Invoice Integration	Scheduled	Audit	Administrator

DEPLOYED

Script Deployment Parameters:

Parameter	Value
COUPA URL	
COUPA API KEY	
GL ACCOUNT SEGMENT	
DEPARTMENT SEGMENT	
SUBSIDIARY ACCOUNT SEGMENT	
CLASS SEGMENT	
LOCATION SEGMENT	

After you are done you can click the Save button and now the Script is successfully deployed



Netsuite Payments to Coupa Invoice & ER Payments 2.0

Overview

This integration syncs up Vendor Payment created in Netsuite and sends it to Coupa. Netsuite is the Payment of record and creates payment information on the Coupa Invoice. The Scheduled SuiteScript is used to capture the “create” events that will call the Coupa POST API to create the new entry
From Bundle Release 6.0.1 Invoice & ER 2.0 is introduced. Please upgrade to use this new script.

Invoice & Expense Payment 2.0 script is available from 6.0.X. Please use Invoice & Expense Payment 2.0 script!

Invoice & Expense Payment script 1.0 will not be supported from 8.0.0.

Script Sequence Explained

This integration is on a schedule that you can set to meet your frequency and volume requirements as Payments are made in NetSuite.

- The Coupa API call to create the **payment** record is on a predefined schedule
- The `nlapiLoadRecord(vendorpayment)` function is used to read the NetSuite vendor payment record
- The `nlapiRequestURL` function is used to do a HTTP PUT to create the corresponding payment records in Coupa

Activities

You will need to do the following to ensure a successful deployment of the Vendor Script

1 Review Parameter List

2 Find and Define Deployment Parameter Values

Tips to help you succeed

- Ensure your currency codes in Coupa are enabled
- Understand volume and frequency of the schedule needed ahead of time to meet your business process
- The script will exclude invoice Payment created by Coupa in Netsuite for Coupa Pay Invoices.

Pre-Requisites

- Coupa P2P & Expenses Bundle deployed
- Coupa API key or OIDC Auth created

Review Parameter List

Review parameter list and decide which ones are relevant to your company’s implementation of Coupa.
Please note the fields that are mandatory are noted with an asterisk *

Script parameters

Script Definition

Script Deployment

Label	Description	Id	Type	Comments	Example	Your Values
Coupa URL*	The Coupa instance the integration connects to	_coupa_pay_url	Free-Form Text		https://abc.coupa host.com	
Coupa API Key*	The API key for Coupa's REST based API	_coupa_pay_apik ey	Free-Form Text	A Coupa Admin can generate the Coupa API Key in Coupa https://xxxx.cou pahost.com/api_ keys	4dtyr7657fgfh575 7fghhg	
Export From Payment Date	Add filter criteria to pull payments from Netsuite created >= to this date	_coupa_pay_from paydate	Date		12/14/2013	
Export To Payment Date	Add filter criteria to pull payments from Netsuite created <= to this date	_coupa_pay_topa ydate	Date		12/27/2013	
Error Email Notification Address	Email addresses to which exception emails to be sent	_coupa_pay_ema il_notifications	Free-Form Text		abc@abc.com , abcd@abcd.com	
Account Name*	As per standard, it will be company name with which environment its been setup (Test or Prod)	_coupa_pay_acco untname	Free-Form Text	<Company Name> <Test/Prod>	Coupa Test	
Process Records From #	Starting record to process. Passed as a parameter to a new scheduled script when usage limit reached or records to process more than "Process Records To #"	_coupa_pay_from records	Integer Number	Default is 0	0	
Process Records To #	Ending record to process. Passed as a parameter to a new scheduled script when usage limit reached or records to process more than "Process Records To #"	_coupa_pay_tore cords	Integer Number	Default is 250	250	

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your Values
Bring Payment to NS ASAP	when true, we import all statuses from payment initiated onward	custscript_coupa_pymt_status_filter	checkbox			
Custom API Arguments	lets you specify custom query parameters, does not let you change status	custscript_coupa_pmnt_cust_args	Text		return_object=sha1low, limit=5	
Static Department	specify a static department internal id for the payment header, or put 'retrieveValue' which will lookup the value from the first vendor bill	custscript_coupa_pymt_headdept	Text			
Static Classification	specify a static classification internal id for the payment header, or put 'retrieveValue' which will lookup the value from the first vendor bill	custscript_coupa_pymt_headclas	Text			
Static Location	specify a static location internal id for the payment header, or put 'retrieveValue' which will lookup the value from the first vendor bill	custscript_coupa_pymt_headlocln	Text			

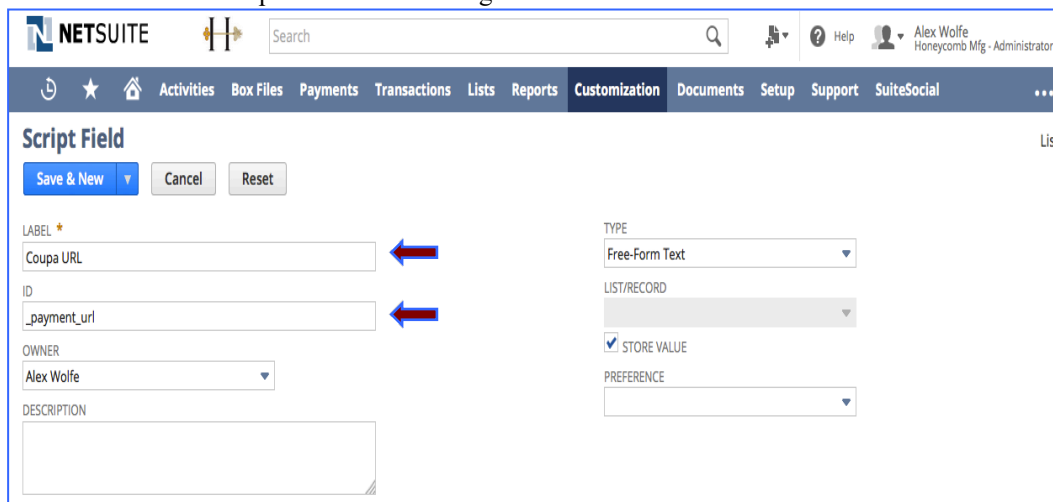
Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your Values
Skip 404 Error	Added script parameter to skip 404 error log and skip the Error email and an Audit log will be added to Execution logs	custscript_coupa_pay_skip_404_log	CheckBox			
Skip Cleared Filter	Clear status not used by some of the customers so it can be expluded	custscript_coupa_pay_skip_cleared_filter	CheckBox	Added 7.0.0		

Find and Define Deployment Parameter Values

Navigation Path: Customization | Scripting | Scripts
Find the “Coupa invoice Payment Integration” script

Now you can click the Parameter menu item and start entering your specific values, please note your specific parameter values may vary based on your Coupa implementation.

Please refer back to the parameters table for guidance.



The screenshot shows the 'Script Field' configuration interface in Netsuite. The top navigation bar includes 'NETSUITE' and a search bar. Below the navigation bar, the 'Customization' tab is selected. The 'Script Field' form has a 'Save & New' button and 'Cancel' and 'Reset' buttons. The form contains the following fields:

- LABEL ***: Coupa URL (indicated by a red arrow)
- ID**: _payment_url (indicated by a red arrow)
- OWNER**: Alex Wolfe
- DESCRIPTION**: (empty text area)
- TYPE**: Free-Form Text
- LIST/RECORD**: (empty dropdown)
- STORE VALUE**: ☒
- PREFERENCE**: (empty dropdown)

After you are done you can click the Save button and now the Script is successfully deployed

Coupa Expense Report to Netsuite Expense Report

Overview

The integration syncs up Coupa Expense Reports and NetSuite Expense Reports via a scheduled SuiteScript. The Coupa Expense Report once “approved for payment” and not exported are ready to be imported into NetSuite. When a new Expense Report is created with status=approved for payment, the NetSuite scheduled script will call the Coupa API and will create new Expense Report. After the pull the integration will set the exported flag to true

For Expense Reports paid via CoupaPay will not set Expense Report Approved in Netsuite to avoid double payments (parameter setup required)

Expense PreApproval (with Virtual Card) is supported from NS Bundle 11.0.0.

The new feature will integrate the Coupa virtual card Charges to Netsuite as Netsuite Credit Card transaction. Credit Card transaction will be booked against the holding account defined in Coupa.

Once Expense Report for this PreApproved spend is approved the final debit account will get updated on the Credit Card transaction in Netsuite. (Note NS Bundle will not create a Netsuite Expense Reports for these Coupa PreApproval Expense reports).

Netsuite Credit Card transaction and Coupa ER will have the reference to the Expense Report and User for transaction for auditability.

As Expense Reports are submitted and approved usually late or even in next financial period. Credit Card transaction must be in an open period, so NS Bundle can update account when Expense Report is approved.

To help simplify/automate moving these outstanding Credit Card transaction to the next period, we created a suitelet.

Navigate to Coupa Menu >> “Open T&E CC Transaction” menu.

Please check APPENDIX for more details.

Script Sequence Explained

This integration is on a schedule that you can set to meet your frequency and volume as Coupa Expense Reports are approved.

- The Coupa API call to create the Expense report is called on a predefined schedule.
- The `nlapiRequestURL` function is used to perform a HTTP GET to retrieve the Coupa Expense that is fully approved and not exported.
- The `nlapiCreateRecord(expensereport)` function is used to generate the Netsuite expense report.

Activities

You will need to do the following to ensure a successful deployment of the Expense script.

1 Review Parameter List

2 Find and Define Deployment Parameter Values

Tips to help you succeed

- The match record in NetSuite will be done on ‘Coupa ER Number’ and ‘External Id’ field. After the pull the exported flag will be set to true. .
- NetSuite Users should not update the ER in NetSuite as this could result in the updates being lost if the Coupa ER is updated after the fact.
- If NetSuite Expense Category name that maps to Coupa Expense Category is changed integration will fail to find the EC.. The change should be made in both Coupa and NetSuite at the same time.

Pre-Requisites

1. Coupa P2P + Expenses bundle deployed
2. Coupa API key or OIDC Auth created.
3. Coupa Chart of Account setup
4. Setup coupa Users with Netsuite employee id
5. Coupa Expense Category Setup
6. NetSuite Expense Category
7. Setup (names must match identically)
8. Coupa commodity and NetSuite mapping (names must match identically)

1

Review Parameter List

Review parameter list and decide which ones are relevant to your company's implementation of Coupa.

Please note the fields that are mandatory are noted with an asterisk *

Script parameters

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your values
Coupa URL *	The Coupa instance the integration will connect to	_coupa_er_url	freeform text		2e528d48825a8fe5a6a2a67972a2d8d21f5j3sdf4 8	
Coupa API Key*	The API Key for Coupa's REST based API	_coupa_er_apikey	freeform text		http://netsuiteintegration-dem.o.coupacloud.com/	
Error email notification addresses	List of emails to receive errors generated by the script execution	_coupa_er_email_addr_notify	freeform text	Separate the emails with a comma and no space	abc1@xyz.com , abc2@xyz.com	
Account Name*	As per standard, it will be company name with which environment its been setup (Test or Prod)	coupa_er_acccountname	freeform text	<company-name> <Test/Prod>		
Don't Import credit card expenses?	Checkbox that when enabled will force the script to skip expenses in Coupa that were created by the credit card integration.	_coupa_er_creditcardskip	checkbox	This will NOT mark them as exported currently, to allow them to be integrated through another point.		
Accounting Approval?	If true, then the expense report will have accounting approval set to true	_coupa_er_creditcardskip	checkbox			
Location Segment	The accounting segment in Coupa that holds the Netsuite location information	_coupa_er_classsegment	freeform text		segment-4	

Class Segment	The accounting segment in Coupa that holds the Netsuite classification account information	<code>_coupa_er_locseg</code>	freeform text		segment-3	
Subsidiary Segment	The accounting segment in Coupa that holds the Netsuite subsidiary account information	<code>_coupa_er_subseg</code>	freeform text		segment-1	
Employee Number Field	Field on the NetSuite Employee record which holds the value matching the Coupa Employee Number field on the User record.	<code>_coupa_er_employee_num</code>	freeform text	By default, the script will compare the Coupa employee number to the NetSuite Employee internalid. You usually would override this if the two do not correlate, or you require the internalID to be a specific format. This is used to match Expense reports to the correct employee.	<code>custentity_coupa_employee_number</code>	
Custom Body fields for expense report	Header custom field of coupa gets syncup with body custom field of netsuite. Note that no data transformation allowed.	<code>_coupa_er_body</code>	text area	Each custom field combination should be separated by a semicolon. The left side of the == is the field in the API response, and the right the internal id of the field in NetSuite that it maps to	<code>requestedby==custbody_couparequester;requested-by-email==custbody_couparequesteremail</code>	
Custom columns Fields	Line custom field of coupa gets syncup with column custom field of netsuite. Note that no data transformation allowed.	<code>_coupa_er_column</code>	text area	Each custom field combination should be separated by a semicolon. The left side of the == is the field in the API response, and the right the internal id of the field in NetSuite that it maps to	<code>sectionname==custcolumn_section_name</code>	

Send Tax code	Option to enable tax code being sent from Coupa to Netsuite so Netsuite calculates tax. When selected tax amount will NOT be added to the vendor bill expense line amount. Also when this option is selected, there will be an additional line per expense line for header level charges	_coupa_er_send_taxcode	checkbox	When using this option, the taxcodes in Coupa need to be stored in the following format: netsuiteid:netsuiteinternalid. E.g. if you have a taxcode in Netsuite VAT21 with internal id of 46 then in Coupa the taxcode should be created as VAT21:46 Additionally, this will not support Multiple Tax Lines in Coupa as there is only one place to set a Tax Code in NetSuite. Tax codes are also only applicable for NON-US Subsidiaries		
Coupa Pay	CoupaPay flag	custscript_coupa_er_coupapay	Check Box	CoupaPay is used to Pay ER		
Corp Card Enable	set Corporate Card Checkbox on Expense Report	custscript_coupa_er_corporate_card	Check Box	Added 6.1.2		
Reimbursable field	enable mapping Coupa's Reimburse to Employee field with NetSuite's Non-Reimbursable field	custscript_coupa_er_enable_reimbursable	Check Box	Added 6.1.2		
Use Approved at Date	Use approved at date for ER instead of created at date	custscript_coupa_er_use_approved_at_date	Check Box	Added 8.0.0 - NIB486		
*Use external IDs?	If customer is using external IDs for COA segments	_coupa_er_extid	Checkbox		T/F	
*Custom API argument	Add custom argument to API call	_er_filter	FreeForm		&limit=20	
*Number of Record Limit	Limit ERs on API call	_expts_limit	Freeform		10	

*Coupa Expense Report URL Link	Saves Coupa ER link to NS ER	_coupa_er_link_filed	FreeForm			
*Use Custom Form ID	Use custom ER Transaction form ID instead of preferred/standard form ID of 46	_coupa_er_customform_id	FreeForm			

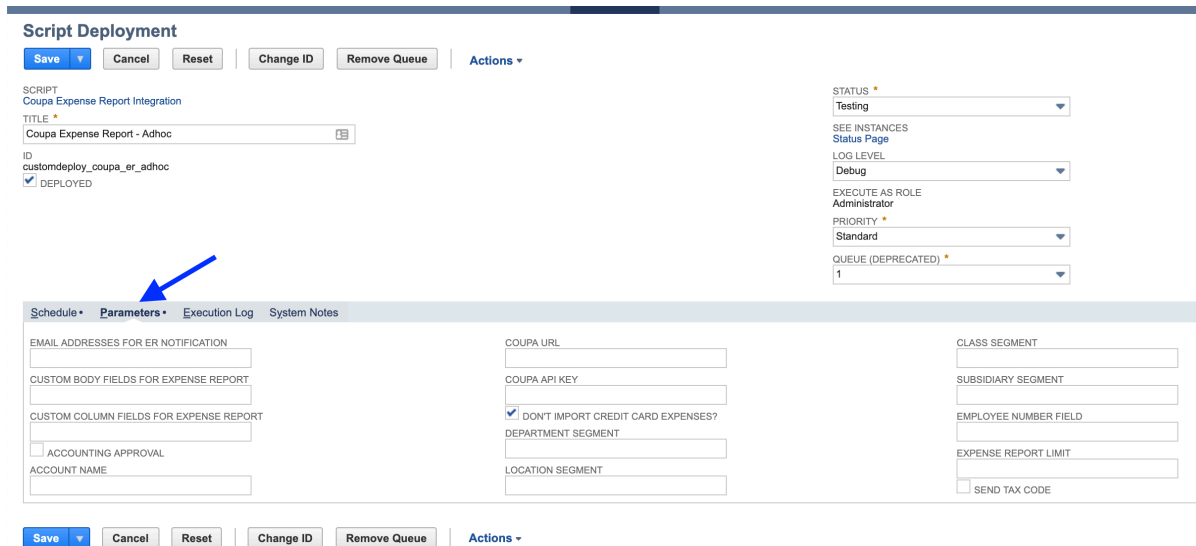
2 Find and Define Deployment Parameter Values

Navigation Path: Customization | Scripting | Scripts

Find the “Coupa Expense Integration” script

Now you can click the Parameter menu item and start entering your specific values, please note your specific parameter values may vary based on your Coupa implementation.

Please refer back to the parameters table for guidance.



Script Deployment

Save Cancel Reset Change ID Remove Queue Actions

SCRIPT
Coupa Expense Report Integration

TITLE *
Coupa Expense Report - Adhoc

ID
customdeploy_coupa_er_adhoc

☒ DEPLOYED

STATUS *
Testing

SEE INSTANCES
Status Page

LOG LEVEL
Debug

EXECUTE AS ROLE
Administrator

PRIORITY *
Standard

QUEUE (DEPRECATED) *
1

Schedule Parameters Execution Log System Notes

EMAIL ADDRESSES FOR ER NOTIFICATION

CUSTOM BODY FIELDS FOR EXPENSE REPORT

CUSTOM COLUMN FIELDS FOR EXPENSE REPORT

☐ ACCOUNTING APPROVAL

ACCOUNT NAME

COUPA URL

COUPA API KEY

☒ DON'T IMPORT CREDIT CARD EXPENSES?

DEPARTMENT SEGMENT

LOCATION SEGMENT

CLASS SEGMENT

SUBSIDIARY SEGMENT

EMPLOYEE NUMBER FIELD

EXPENSE REPORT LIMIT

☐ SEND TAX CODE

Save Cancel Reset Change ID Remove Queue Actions

After you are done you can click the Save button and now the script is successfully deployed. Once testing is complete, schedule the deployment called “Coupa Expense Report Scheduled”. The Adhoc deployment should be used for testing and one-off executions. The log level of the Scheduled deployment should be set to Audit.

Coupa Expense Report Payment to Netsuite ER Payment

Overview

The integration syncs up Coupa Expense Report Payments via CoupaPay to Netsuite via a scheduled SuiteScript. The Coupa Expense Report Payment will be integrated when the payment is moved to “completed_successfully” or “completed_with_errors” in. After the pull the integration will set the exported flag to true on the Expense Payments.

Script Sequence Explained

This integration is on a schedule that you can set to meet your frequency and volume for Coupa Expense Payments.

- The Coupa API call to create the Expense Payment is called on a predefined schedule.
- The nlapiRequestURL function is used to perform a HTTP GET to retrieve the Coupa Expense Payment that is fully “completed_successfully” or “completed_with_errors” and not exported.

Activities

You will need to do the following to ensure a successful deployment of the Expense script.

1 Review Parameter List

2 Find and Define Deployment Parameter Values

Pre-Requisites

1. Coupa P2P + Expenses bundle deployed
2. Coupa API key created.
3. Coupa Chart of Account setup
4. Coupa Company Payment Account Setup for Expense Payments
5. Coupa Users Payment Method set to
6. Netsuite EFT is disabled

Tips to help you succeed

- Map the Coupa CPA to NS Bank account correctly to avoid payment creation issues

Review Parameter List

Review parameter list and decide which ones are relevant to your company's implementation of Coupa.

Please note the fields that are mandatory are noted with an asterisk *

Script parameters

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your values
Coupa URL *	The Coupa instance the integration will connect to	_coupa_er_url	freeform text		2e528d48825a8fe5a6a2a67972a2d8d21f5j3sdf4 8	
Coupa API Key*	The API Key for Coupa's REST based API	_coupa_er_apikey	freeform text		http://netsuiteintegration-dem.o.coupacloud.com/	
Email Errors To	List of emails to receive errors generated by the script execution	_custscript_coupa_erpay_errorto	freeform text	Separate the emails with a comma and no space	abc1@xyz.com , abc2@xyz.com	
Email Errors From	NS Internal ID of the user the emails should be from	custscript_coupa_erpay_errorfrm	freeform text			
Custom Payment Status	Custom status to export CoupaPay Payments. If blank, defaults to completed_successfully.	custscript_coupa_erpay_status	Free-Form Text			
Force Save?	If true, then the expense report will have accounting approval set to true	customscript_coupa_erpay_forcesav	checkbox	If true, will pass a parameter when saving to skip validations		
Coupa Acct to NS Acct	Mapping from Coupa Pay From Account ID to the NetSuite bank account number location information	custscript_coupa_erpay_accmap	freeform text			
Custom Status Filter	Apply custom status filter	custscript_coupa_erpay_status_filter	freeform text			

Custom Query Arguments	Apply custom API filter	custscript_coupa_erpay_cust_args	freeform text		When the SUBS_CURR_ID from the mapping matches the subsidiary currency the account from mapping provided will be used and set on the payment transaction mapping(CPA_ID==NS_ACCOUNT==SUBS_ID)	
Default Dept	Default Department on ER Payment	custscript_coupa_erpay_headdept	freeform text	Added 6.1.2	3==2892;11==213	
Default Class	Default Class on ER Payment	custscript_coupa_erpay_headclass	freeform text	Added 6.1.2	3==2892;11==213	
Default Locn	Default Location on ER Payment	custscript_coupa_erpay_headlocn	freeform text	Added 6.1.2	3==2892;11==213	
Map custom field	Map custom field	custscript_coupa_erpay_header_cust_field	freeform text	Added 6.1.2	id==custbody_test_field;pay-from-account.id==custbody_custom_field	
Use updated_at date	Use updated_at date instead of released_at date override	custscript_coupa_erpay_use_updated_at	CheckBox	Added 7.0.0		
*Support Single Currency	In case of single currency used in NS, use this flag to set default currency	_coupa_erpay_default_currency	FreeForm			

2 Find and Define Deployment Parameter Values

Navigation Path: Customization | Scripting | Scripts

Find the “Coupa Expense Payment Integration” script

Now you can click the Parameter menu item and start entering your specific values, please note your specific parameter values may vary based on your Coupa implementation.

Please refer back to the parameters table for guidance.

After you are done you can click the Save button and now the script is successfully deployed. Once testing is complete, schedule the deployment called “Coupa Expense Payment Scheduled”. The Adhoc deployment should be used for testing and one-off executions. The log level of the Scheduled deployment should be set to Audit.



Coupa Expense Report to Netsuite Vendor Bill

Overview

The integration takes a Coupa Expense Report and turns it into a NetSuite Vendor bill. When a new Expense Report is created with status=approved for payment, the NetSuite scheduled script will call the Coupa API and will create Vendor Bill. After the pull the integration will set the exported flag to true.

when clients want to pay their employees in a currency other than the “functional currency” of their subsidiary. For example, if an employee resides in the Japan, and the functional subsidiary currency is set to Yen, the employee can only be reimbursed in Yen.

Script Sequence Explained

This integration is on a schedule that you can set to meet your frequency and volume as Coupa Expense Reports are approved.

- The Coupa API call to create the Vendor Bill is called on a predefined schedule.
- The HTTP GET function is used to retrieve the Coupa Expense that is fully approved and not exported.
- The createVendorBill(expenseReport, vendorId, reportId) is used to generate the Netsuite Vendor Bill.

Activities

You will need to do the following to ensure a successful deployment of the Expense script.

1 Review Parameter List

2 Find and Define Deployment Parameter Values

Tips to help you succeed

- When an existing Expense Report that was already exported to NetSuite is updated then Coupa will set the exported flag to false.
- The NetSuite scheduled script will call the Coupa API and update the Expense Report where the “exported” flag is false and “status” is approved for payment.
- *Note: Editing the Expense Report in NetSuite is restricted. Any changes will be overridden by Coupa ER dataER*

Pre-Requisites

1. Coupa P2P + Expenses bundle deployed
2. Coupa API key or OIDC Auth created.
3. Coupa Chart of Account setup
4. Setup coupa Users with Netsuite vendor id
5. Coupa Expense Category Setup
6. NetSuite Expense Category Setup (names must match identically)
7. Coupa commodity and NetSuite mapping (names must match identically)
8. If NetSuite Expense Category name that maps to Coupa Expense Category is changed, the integration will fail to find the Expense Category. The change should be made in both Coupa and NetSuite at the same time

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your values
Coupa URL *	The Coupa instance the integration will connect to	custscript_coupa_er2vb_url	freeform text		https://abc-demo.coupa-dev.com	
Coupa API Key*	The API Key for Coupa's REST based API	custscript_coupa_er2vb_apikey	Freeform text		j3jn1j3bmmdloq212313m1b1b231m1k23	
Custom Export Field	Used to denote a custom export field, rather than last_exported_at	custscript_coupa_er2vb_export	Freeform text		exported-to-netsuite	
Custom Status Field	Used to set a status other than approved_for_payment to query by	custscript_coupa_er2vb_status	Freeform text		accounting_review	
Report Execution Limit	In some cases, the Coupa API may take too long to return expense reports. So this can be used to limit the number of records returned	custscript_coupa_er2vb_limit	Freeform text		10	
Query by Updated Date?	If true, then the two updated fields will be checked and used to filter expense reports	custscript_coupa_er2vb_updatedate	checkbox		True	
Updated After	If Query by Updated Date is set, and this populated, then a parameter will be added to only retrieve reports updated after the provided date	custscript_coupa_er2vb_fm_updated	date		2020-1-30	
Updated Before	If Query by Updated Date is set, and this populated, then a parameter will be added to only retrieve reports updated before the provided date	custscript_coupa_er2vb_to_updated	Date		2020-1-15	
External Source Filter Value	Used to skip Corporate card files	custscript_coupa_er2vb_extsrcfilter	Freeform-text		corporate_credit_card_amex	
Custom	Used to denote a custom	custscript_c	Freeform-text		netsuite-	

Employee Number	field on the Expensed By user in Coupa that should be used instead of employee number to match entityid (Vendor ID)	oupa_er2vb_employee_field			vendor-id	
Account Segment	Xml tag for the GL Account segment	custscript_coupa_er2vb_glaccseg	Freeform-text		segment-2	
Line Custom Field Mappings	Line level custom field mappings in the format: coupa-xml-tag==netsuitefield;	custscript_coupa_er2vb_lfields	Text-area		coupa-xml-tag==netsuitetag1;coupa-xml-tag2==netsuitetag2;	
Header Custom Field Mappings	Header level custom field mappings in the format: coupa-xml-tag==netsuitefield;	custscript_coupa_er2vb_hdrfields	Text-area		coupa-xml-tag==netsuitetag1;coupa-xml-tag2==netsuitetag2;	
Account Payable number	The GL number for the account payable you wish to use	custscript_coupa_er2vb_payacc	Freeform-text		20001	
Cut-off Date	The calendar date after which you do not want to allow reports to post in the previous period	custscript_coupa_er2vb_cutoff			7	
Force Saving	If true, then save will ignore Mandatory fields	custscript_coupa_er2vb_forcesave	checkbox			
Department Segment	Xml tag for the department segment	custscript_coupa_er2vb_deptseg			Segment-3	
Class Segment	Xml tag for the class segment	custscript_coupa_er2vb_classseg			Segment-4	
Location Segment	Xml tag for the location segment	custscript_coupa_er2vb_locseg			Segment-5	
Subsidiary Segment	Xml tag for the subsidiary segment	custscript_coupa_er2vb_subsseg			Segment-1	
Error To Emails	List of emails that should receive the alerts	custscript_coupa_er2vb_errorto			<u>abc@example.com</u> , <u>xyz@example.com</u>	
Error From ID	Internal ID of the netsuite Employee that the emails should be from	custscript_coupa_er2vb_errorfrm			1	

Corp Card Vendor	The field can be a select field with a list of vendor OR a text field with the internal Id of the vendor. If the externalSrcName is not blank or web the Expense will be considered as Corp Card Expense and the selected Vendor in the parameter will be set as the Entity on the Vendor Bill instead of the actual expensed by user Vendor record.	custscript_coupa_er2vb_corp_card_vendor	List/Record			
Multi Corp Card vendor mapping	Mapping multiple Corporate Card Vendors.	custscript_coupa_er2vb_corpcard_mapping	text	Added 6.1.2	corpcc.master==18555;corpc.c.visa==3456;corpc.c.mastercard==3457;corpcc.amex==345	
Default currency	Use default currency in case only once currency used	custscript_coupa_er2vb_pay_def_currency				

2 Find and Define Deployment Parameter Values

Navigation Path: Customization | Scripting | Scripts
Find the “Coupa ExpenseReport to Vendor Bill ” script

Now you can click the Parameter menu item and start entering your specific values, please note your specific parameter values may vary based on your Coupa implementation.

Please refer back to the parameters table for guidance.

Script Deployment

[Save](#)
[Cancel](#)
[Reset](#)
[Change ID](#)
[Actions](#)

SCRIPT
Coupa ExpenseReport to Vendor Bill

TITLE
Coupa ExpenseReport to Vendor Bill

ID
customdeploy1
☒ DEPLOYED

STATUS
Not Scheduled

SEE INSTANCES
Status Page

LOG LEVEL
Debug

EXECUTE AS ROLE
Administrator

PRIORITY
Standard

CONCURRENCY LIMIT
1

☒ SUBMIT ALL STAGES AT ONCE

YIELD AFTER MINUTES
60

BUFFER SIZE
1

[Schedule](#)
[Parameters](#)
[Execution Log](#)
[System Notes](#)

COUPA URL

COUPA API KEY

CUSTOM EXPORT FIELD

CUSTOM STATUS FIELD

REPORT EXECUTION LIMIT

☐ QUERY BY UPDATED DATE?

UPDATED AFTER

UPDATED BEFORE

EXTERNAL SOURCE FILTER VA

CUSTOM EMPLOYEE NUMBER

LINE CUSTOM FIELD MAPS

ACCOUNT PAYABLE NUMBER

CUT-OFF DAY

HEADER CUSTOM FIELD MAPS

☐ FORCE SAVE

ACCOUNT SEGMENT

DEPARTMENT SEGMENT

CLASS SEGMENT

LOCATION SEGMENT

SUBSIDIARY SEGMENT

ERROR TO EMAILS

ERROR FROM ID

After you are done you can click the Save button and now the script is successfully deployed. Once testing is complete, schedule the deployment called “Coupa Expense Report Scheduled”. The Adhoc deployment should be used for testing and one-off executions. The log level of the Scheduled deployment should be set to Audit.

Coupa Charge to Netsuite Credit Card transaction

Overview

Customers with Coupa Pay PO Payments may use this integration. The integration takes a Coupa Charges for Purchase Orders and turns it into a NetSuite Credit Card Transactions. When a new Charge is created the NetSuite scheduled script will call the Coupa API and will create a Credit Card transaction. After the pull the integration will set the exported flag to true.

Script Sequence Explained

This integration is on a schedule that you can set to meet your Charge data load to Coupa.

- The Coupa API call to create the Vendor Bill is called on a predefined schedule.
- The HTTP GET function is used to retrieve the new Coupa Charges for Purchase Orders.
- Script is used to generate the Netsuite credit Card Transactions.

Activities

You will need to do the following to ensure a successful deployment of the Expense script.

- 1 Review Parameter List**
- 2 Find and Define Deployment Parameter Values**

Tips to help you succeed

- Charges will be loaded to Coupa on a daily basis, so schedule your integration daily. Align timing at implementation

Pre-Requisites

- Coupa P2P + Coupa Pay
- Coupa API key created.
- Coupa Chart of Account setup
- Coupa Payment Partner Setup
- Coupa Supplier Payment Method setup
- Account for credit card charges in Netsuite
- Issuer Bank as Supplier in Netsuite
- Budget Periods(this is used to find Accounting Periods in NS) have to be created for that COA in Coupa

Review Parameter List

Review parameter list and decide which ones are relevant to your company's implementation of Coupa. Please note the fields that are mandatory are noted with an asterisk *

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your values
Email Errors To	Comma separated list of emails to receive error notifications	custscript_coupa_chrg_errorto	freeform text		test@coupa.com	
Email Errors From	NS Internal ID of the user the emails should be from	custscript_coupa_chrg_errorfrom	Freeform text		test@coupa.com	
Coupa URL	URL of your coupa instance	custscript_coupa_chrg_url	Freeform text		https://netsuite-integrations.coupa.com	
Coupa API Key	custscript_coupa_chrg_apikey	API key from your Coupa instance	Freeform text		102983oilJ092831237772310	
Subsidiary Segment	XML tag for the subsidiary segment	custscript_coupa_chrg_subsseg	Freeform text		segment-1	
Account Segment	XML tag for the gl account segment	custscript_coupa_chrg_acct	checkbox		segment-2	
Department Segment	custscript_coupa_chrg_dept	XML tag for the department segment	date		segment-3	
Class Segment	XML tag for the Class segment	custscript_coupa_chrg_class	Date		segment-4	
Location Segment	custscript_coupa_chrg_locn	custscript_coupa_chrg_locn	Freeform-text		segment-5	
Charge Custom Fields	Custom field mapping for extra fields	custscript_coupa_chrg_linecf	Freeform-text		coupa-xml-path==netsuite_field_id	
Charge Custom Fields	Custom field mapping for extra fields	custscript_coupa_chrg_headcf	Freeform-text		coupa-xml-path==netsuite_field_id	
Force Save?	If true, will pass a parameter when saving to skip validations	custscript_coupa_chrg_forcesave	CheckBox			

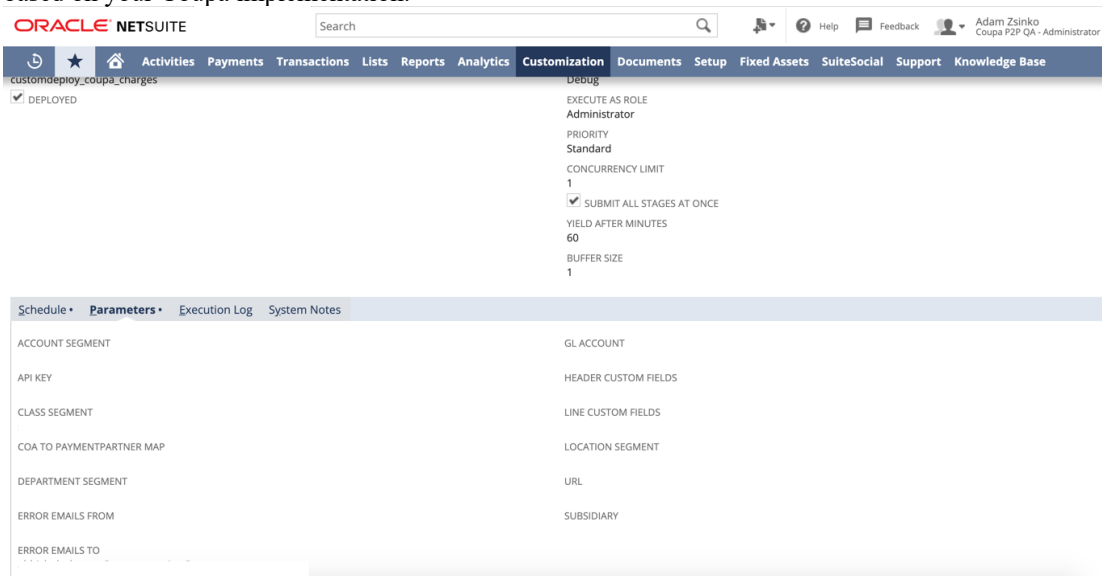
CoA ID to Vendor ID	map from the Coupa Chart of Accounts ID to the NetSuite Vendor ID representing the Payment Partner or Credit Card Vendor that these charges should be tied to. In Coupa, one CoA is tied to one Payment Partner, so the same applies here, one CoA ties to one Vendor	custscript_coupa_chrg_coatopp	Text-area		1==3412;2==4812;3==452	
Account Payable Number	The internal ID for the Account Payable that should be set on the transaction linked to the Coupa COA	custscript_coupa_chrg_glacc	Freeform-text		1==245	
Cut off days for Posting Period	Add +/- posting cutoff days	custscript_coupa_chrg_cutoff_day	Freeform-text			
Issuer Reconciliation ID	Issuers unique transaction ID	custbody_coupa_issue_r_recon_id	Freeform-text			
Holding Account Mapping	COA to Holding account mapping for Ghost Card	custscript_holding_acc_mapping			COA_ID1==HOLDING_ACCOUNT_NUMBER1; COA_ID2==HOLDING_ACCOUNT_NUMBER2	
Holding Department Mapping	COA to Holding Department mapping for Ghost Card	custscript_holding_dept_mapping			COA_ID1==HOLDING_DEPARTMENT_ID1; COA_ID2==HOLDING_DEPARTMENT_ID2	

Holding Location Mapping	COA to Holding Location mapping for Ghost Card	custscript_holding_location_mapping			COA_ID1==HOLDING_LOCATION_ID1;COA_ID2==HOLDING_LOCATION_ID2	
Holding Class Mapping	COA to Holding Class mapping for Ghost Card	custscript_holding_class_mapping			COA_ID1==HOLDING_CLASS_ID1;COA_ID2==HOLDING_CLASS_ID2	

2 Find and Define Deployment Parameter Values

Navigation Path: Customization | Scripting | Scripts
Find the “Coupa Charges Integration” script

Now you can click the Parameter menu item and start entering your specific values, please note your specific parameter values may vary based on your Coupa implementation.



The screenshot shows the Oracle NetSuite interface. The top navigation bar includes 'Activities', 'Payments', 'Transactions', 'Lists', 'Reports', 'Analytics', 'Customization', 'Documents', 'Setup', 'Fixed Assets', 'SuiteSocial', 'Support', and 'Knowledge Base'. The 'Customization' tab is active, and the 'Parameters' sub-tab is selected. The main area displays the 'customdeploy_coupa_charges' script with a 'DEPLOYED' status. Below the script details, there is a table of parameters:

Parameter Name	Value
ACCOUNT SEGMENT	GL ACCOUNT
API KEY	HEADER CUSTOM FIELDS
CLASS SEGMENT	LINE CUSTOM FIELDS
COA TO PAYMENTPARTNER MAP	LOCATION SEGMENT
DEPARTMENT SEGMENT	URL
ERROR EMAILS FROM	SUBSIDIARY
ERROR EMAILS TO	

Please refer back to the parameters table for guidance.

After you are done you can click the Save button and now the script is successfully deployed. Once testing is complete, schedule the deployment called “Coupa Charges Integration Scheduled”. The Adhoc deployment should be used for testing and one-off executions. The log level of the Scheduled deployment should be set to Audit.



Coupa PrePayment to Netsuite PrePayment

Overview

Pre-Payment (milestone) can be created on Coupa Purchase Orders for non CoupaPay Suppliers.

Once Purchase Order is approved Pre-Payments will be integrated to Netsuite.

PrePayments must be paid in Netsuite. Integration will bring Pre-Payment “paid” status back to Coupa which allows the Pre-Payment to be allocated to an Invoice.

Pre-Payment allocation in Coupa will be integrated to Netsuite creating a Pre-Payment Allocation.

When Coupa Invoice integrated to Netsuite, it will have the relevant Pre-Payment linked through the Pre-Payment Allocation which will reduce automatically the Invoice Payment due amount in Netsuite.

Script Sequence Explained

This integration is on a schedule that you can set to send Pre-Payment and Pre-Payment Allocation data to Netsuite from Coupa and send Pre-Payment payment status from Netsuite to Coupa

This Integration involves 3 scripts.

Activities

You will need to do the following to ensure a successful deployment of the Expense script.

1

Review Parameter List

2

Find and Define Deployment Parameter Values

Tips to help you succeed

- Make sure Pre-Payment approval is set in NS (or set to auto approve)
- Scripts are schedules correctly and running frequently

Pre-Requisites

- Coupa P2P + Coupa Pay
- Coupa API key or OIDC Auth created.
- Script deployment parameter is set
- Coupa/Netsuite Segment mapping is setup correctly
-

Review Parameter List

Review parameter list and decide which ones are relevant to your company's implementation of Coupa.
Please note the fields that are mandatory are noted with an asterisk *

Coupa Vendor PrePayment Integration Script

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your values
Coupa URL		custscript_coupa_prepay_host_url	Hyperlink			
Error Email Receiver(Comma separated)		custscript_coupa_prepay_error_to	Text Area			
Error Email Sender		custscript_coupa_prepay_error_from	List/Record	Employee		
Custom Body Field Mapping		custscript_coupa_prepay_cust_body_fields	Text Area			
Custom Arguments		custscript_coupa_prepay_custom_arguments	Text Area			
Department Segment Mapping		custscript_coupa_prepay_depart_mapping	Text Area			
Class Segment Mapping		custscript_coupa_prepay_class_mapping	Text Area			
Location Segment Mapping		custscript_coupa_prepay_location_mapping	Text Area			
Account Segment Mapping		custscript_coupa_prepay_account_mapping	Text Area			
Posting Period Cut Off		custscript_coupa_prepay_cutoff_day	Free-Form Text			
Auto Approve PrePayment?		custscript_coupa_prepay_auto_approve	Check Box			
EFT Field ID		custscript_coupa_prepay_eft_option	Free-Form Text			
Subsidiary Segment		custscript_coupa_prepay_sub_segment	Free-Form Text			

Coupa PrePaymentReconciliation Line Sync SCRIPT

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your values
Coupa URL		custscript_coupa_prepay_host_url	Hyperlink			
Error Email Receiver		custscript_coupa_prepay_sync_error_to	Text Area			
Error Email Sender		custscript_coupa_prepay_sync_error_from	List/Record			
Export From Prepayment Date		custscript_coupa_prepay_export_from_date	Date			
Export To PrePayment Date		custscript_coupa_prepay_export_to_date	Date			
Sync Voided PrePayments		custscript_coupa_prepay_sync_voided	Check Box			

Coupa Prepayment Application to VB Script is a User Event script, no Parameter required

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your values

2 Find and Define Deployment Parameter Values

Scripts used for this Integration are:

- Coupa Vendor PrePayment Integration Script – MAP/REDUCE
- Coupa PrePaymentReconciliation Line Sync SCRIPT – MAP/REDUCE
- Coupa Prepayment Application to VB Script is a User Event script, no Parameter required
-



Coupa Invoice Payment to Netsuite Vendor Bill Payment

Overview

Customers with Coupa Pay Invoice Payments may use this integration. The integration takes a Coupa Invoice Payments and turns it into a NetSuite Vendor Bill Payments. When an Invoice Payment is moved to “Completed Successfully” in Coupa the NetSuite scheduled script will call the Coupa API and will create a Bill Payment . The script will reverse Vendor Bill Payment and reset Payment Hold flag on invoice if the Payment status in Coupa changes to “Completed with Error”. After the pull the integration will set the exported flag to true on the Coupa Payment.

In NS Bundle release 3.0.0 we added option to

- Create one payment in Netsuite across all Invoices/Vendor bills in the Coupa Payment. This help with reconciliation and Credit Note handling in Payments. This feature requires a new parameter to be setup.
- Integrate payments as soon as they are released in Coupa to Netsuite, so no delay while the payment transition to “Completed Successfully” status. This feature requires a new parameter to be setup.

Script Sequence Explained

This integration is on a schedule that you can set to send Payment data to Netsuite from Coupa

- The Coupa API call to create the Vendor Bill is called on a predefined schedule.
- The HTTP GET function is used to retrieve the new Coupa Invoice Payments at “completed_successfully” status or at any status (from 3.0.0)
- Script is used to generate the Netsuite Vendor Bill Payment

Activities

You will need to do the following to ensure a successful deployment of the Expense script.

1

Review Parameter List

2

Find and Define Deployment Parameter Values

Tips to help you succeed

- Payments will be pulled when moves to completed_successfully status. Do not populate “status” parameter
- If you wish to see payment in netsuite asap, check parameters

Pre-Requisites

- Coupa P2P + Coupa Pay
- Coupa API key or OIDC Auth created.
- Coupa Chart of Account setup
- Coupa Payment Partner (if Virtual Card on Invoice Payment used)
- Coupa Company Payment Account(s)
- Coupa Supplier Payment Method setup
- Coupa Supplier Payment Account(s)

Review Parameter List

Review parameter list and decide which ones are relevant to your company's implementation of Coupa. Please note the fields that are mandatory are noted with an asterisk *

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your values
Email Errors To	Comma separated list of emails to receive error notifications	custscript_coupa_chrg_errorto	freeform text		test@coupa.com	
Email Errors From	NS Internal ID of the user the emails should be from	custscript_coupa_chrg_errorfrom	Freeform text		test@coupa.com	
Coupa URL	URL of your coupa instance	custscript_coupa_chrg_url	Freeform text		https://netsuite-integrations.coupa.com	
Coupa API Key	custscript_coupa_chrg_apikey	API key from your Coupa instance	Freeform text		102983oilJ092831237772310	
Custom Payment Status	Custom status to export Coupa Payments. If blank, defaults to completed_successfully	custscript_coupa_pymt_status	Freeform text	DO NOT populate, it will be set completed_successfully by default		
Force Save?	If true, will pass a parameter when saving to skip validations	customscript_coupa_pymt_forcesave	checkbox			
Coupa Account to NS Account	This is a mapping from the Coupa Pay From (Company) Account to the NetSuite Pay From account representing the bank account that	custscript_coupa_pymt_accmap		When the SUBS_CURR_ID from the mapping matches the subsidiary currency the account from mapping provided will be used and set on the payment	1==3412;2==4812;3==452	

Navigation Path: Customization | Scripting | Scripts
Find the “CoupaPay Payment Integration” script

ORACLE

NETSUITE

Help
 Feedback
 Adam Zsinko
Coupa P2P QA - Ad

[Activities](#)
[Payments](#)
[Transactions](#)
[Lists](#)
[Reports](#)
[Analytics](#)
[Customization](#)
[Documents](#)
[Setup](#)
[Fixed Assets](#)
[SuiteSocial](#)
[Support](#)
[Knowledge Base](#)

CoupaPay Payments Integration

ID
 customdeploy1

☒ DEPLOYED

Status Page

LOG LEVEL
 Debug

EXECUTE AS ROLE
 Administrator

PRIORITY
 Standard

CONCURRENCY LIMIT
 1

☒ SUBMIT ALL STAGES AT ONCE

YIELD AFTER MINUTES
 60

BUFFER SIZE
 1

[Schedule](#)
[Parameters](#)
[Execution Log](#)
[System Notes](#)

COUPA URL

PAY ACCOUNT MAPPING

COUPA API KEY

PAYMENT EXPORT STATUS

ERROR EMAILS FROM

ERROR EMAILS TO

After you are done you can click the Save button and now the script is successfully deployed. Once testing is complete, schedule the deployment called “CoupaPay Payment Integration Scheduled”. The Adhoc deployment should be used for testing and one-off executions. The log level of the Scheduled deployment should be set to Audit.



Coupa Expense Report Payment to Vendor Bill Payment

Overview

Customers with Coupa Pay Expense Payments may use this integration. The integration takes a Coupa Invoice Payments and turns it into a NetSuite Vendor Bill Payments. When an Expense Payment is moved to “Completed Successfully” in Coupa the NetSuite scheduled script will call the Coupa API and will create a Bill Payment . The script will reverse Vendor Bill Payment and reset Payment Hold flag on Vendor Bill if the Payment status in Coupa changes to “Completed with Error”. After the pull, the integration will set the exported flag to true on the Coupa Expense Payment.

- Create one payment in Netsuite across all Invoices/Vendor bills in the Coupa Expense Payment.
- Integrate Expense payments as soon as they are released in Coupa to Netsuite, so no delay while the payment transition to “Completed Successfully” status. This feature requires a new parameter to be setup.

Script Sequence Explained

This integration is on a schedule that you can set to send Payment data to Netsuite from Coupa .

- The Coupa API call to create the Vendor Bill is called on a predefined schedule.
- The HTTP GET function is used to retrieve the new Coupa Expense Payments at “completed_successfully” status or at any status
- Script is used to generate the Netsuite Vendor Bill Payment

Activities

You will need to do the following to ensure a successful deployment of the Expense script.

1

Review Parameter List

2

Find and Define Deployment Parameter Values

Tips to help you succeed

- Payments will be pulled when moves to completed_successfully status. Do not populate “status” parameter
- If you wish to see payment in netsuite asap, check parameters

Pre-Requisites

- Coupa Expense + Coupa Pay
- Coupa API key or OIDC Auth created.
- Coupa Chart of Account setup
- Coupa Company Payment Account(s)
- Coupa Employee Payment Method setup
- Coupa Employee Payment Account(s)

Review Parameter List

Review parameter list and decide which ones are relevant to your company's implementation of Coupa. Please note the fields that are mandatory are noted with an asterisk *

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your values
Email Errors To	Comma separated list of emails to receive error notifications	custscript_coupa_chrg_errorto	freeform text		test@coupa.com	
Email Errors From	NS Internal ID of the user the emails should be from	custscript_coupa_chrg_errorfrom	Freeform text		test@coupa.com	
Coupa URL	URL of your coupa instance	custscript_coupa_chrg_url	Freeform text		https://netsuite-integrations.coupa.com	
Coupa API Key	custscript_coupa_chrg_apikey	API key from your Coupa instance	Freeform text		102983oiIJ092831237772310	
Custom Payment Status Filter	CoupaPay Payments. If blank, defaults to completed_successfully otherwise all payments exported an any status	custscript_coupa_er2v_bpay_status_filter	checkbox	DO NOT populate, it will be set completed_successfully by default		
Custom payment status	Integrate only payments in custom status	custscript_coupa_er2v_bpay_status	Free-Form-text			
Coupa Account to NS Account	This is a mapping from the Coupa Pay From (Company) Account to the NetSuite Pay From account representing the bank account that these payments	custscript_coupa_er2v_bpay_accmap		When the SUBS_CURR_ID from the mapping matches the subsidiary currency the account from mapping provided will be used and set on the payment transaction mapping(CPA_ID	1==3412;2==4812;3==452	

	should be tied to.			==NS_ACCOUNT ==SUBS_ID)		
Custom Query Argument	Add any custom query argument to the API call	custscript_coupa_er2v bpay_cust_args	Free-Form-Text		&created-at[gt]=2020-07-01	
Header Department		custscript_coupa_er2v bpay_headdept	Free-Form-Text	If Header level Department is required, set to either default value or 'retrieveValue' to pull from the first line	23	
Header Class		custscript_coupa_er2v bpay_headclass	Free-Form-Text	If Header level Classification is required, set to either default value or 'retrieveValue' to pull from the first line	12	
Header Location		custscript_coupa_er2v bpay_headlocn	Free-Form-Text	If Header level Location is required, set to either default value or 'retrieveValue' to pull from the first line	11	
Custom field mapping	Map header custom field	custscript_coupa_er2v bpay_header_custfil	Free-Form-Text	Added 6.1.2	custbody_test_field;pay-from-account.id==memo	
Override released_at	Override usage of released_at date to use update_at date for payment	custscript_coupa_er2v bpay_use_updated_at	CheckBox	Added 7.0.0		

2 Find and Define Deployment Parameter Values

Navigation Path: Customization | Scripting | Scripts

Find the “CoupaPay Payment Integration” script

Now you can click the Parameter menu item and start entering your specific values, please note your specific parameter values may vary based on your Coupa implementation.

Please refer back to the parameters table for guidance.

After you are done you can click the Save button and now the script is successfully deployed. Once testing is complete, schedule the deployment called “CoupaPay ER2VB Payment Integration Scheduled”. The Adhoc deployment should be used for testing and one-off executions. The log level of the Scheduled deployment should be set to Audit.

Coupa Dynamic Discounting to Netsuite Invoice

Overview

Customers with CoupaPay Dynamic Discounting may use this integration. The integration checks CoupaPay Invoice status and makes the necessary updates to the Netsuite Bill to reflect the Early Payment Discount process.

When a Supplier invoice in Coupa is eligible for Working Capital Consideration the integration will put the Bill in Netsuite on “Hold Payment”, so it can not be included in the payment run while the Early Pay Discount (EPD) process is on. When the EPD is accepted or the EPD has expired, based on the Coupa PayInvoice, integration will remove NS Bill “Hold Payment” flag, and will apply the Payment Term applicable based on the EPD.

The script will create NS Payment term with prefix as
“_DD_Terms_Percentage%_DueDays”

- Supplier Invoice in Coupa is under Working Capital Consideration
- Integration sets NS Bill/invoice “Hold Payment” flag
- Integration removes NS Bill/Invoice. “Hold Payment” flag. And applies EPD Payment Term

Script Sequence Explained

This integration is on a schedule that you can set to send Dynamic Discounting data to Netsuite from Coupa .

- The Coupa API call to update the Vendor Bill is called on a predefined schedule.
- The HTTP GET function is used to retrieve the CoupaPay Invoice status

Activities

You will need to do the following to ensure a successful deployment of the Dynamic Discounting script.

1

Review Parameter List

2

Find and Define Deployment Parameter Values

Tips to help you succeed

- CoupaPay Invoice in. Working Capital Consideration
- New Payment Terms created automatically

Pre-Requisites

- Coupa API key or OIDC Auth created.
- Coupa Chart of Account setup
- Coupa Company Payment Account(s)
- Supplier Working Capital Whitelisted in Coupa

Review Parameter List

Review parameter list and decide which ones are relevant to your company's implementation of Coupa. Please note the fields that are mandatory are noted with an asterisk *

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your values
Email Errors Receiver	Comma separated list of emails to receive error notifications	custscript_coupa_dd_email_receiver	freeform text		test@coupa.com	
Email Errors Sender	NS Internal ID of the user the emails should be from	custscript_coupa_dd_email_sender	Freeform text		test@coupa.com	
Coupa URL	URL of your coupa instance	custscript_coupa_dd_url	Freeform text		https://netsuite-integrations.coupa.com	
Coupa API Key	custscript_coupa_chrg_apikey	custscript_coupa_dd_api_key	Freeform text		102983oiIJ092831237772310	

Find and Define Deployment Parameter Values

Navigation Path: Customization | Scripting | Scripts
Find the "CoupaPay Dynamic Discounting" script

Now you can click the Parameter menu item and start entering your specific values, please note your specific parameter values may vary based on your Coupa implementation.

Please refer back to the parameters table for guidance.

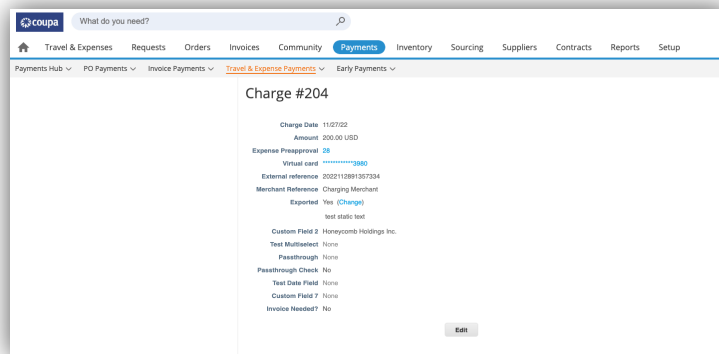
After you are done you can click the Save button and now the script is successfully deployed. Once testing is complete, schedule the deployment called "CoupaPay Dynamic Discounting". The Adhoc deployment should be used for testing and one-off executions. The log level of the Scheduled deployment should be set to Audit.

Expense PreApproval Process

Expense PreApproval (with Virtual Card) is supported from NS Bundle 11.0.0.

The process starts with an Expense PreApproval request. Once the Request is approved, the User will be provided with a Virtual Card. The Coupa Steps are not described here.

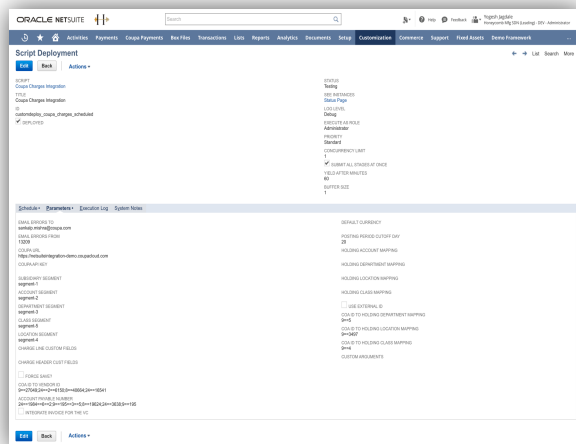
1. As soon the Virtual Card is used and charge has flown into Coupa from the card network, NS Bundle will integrate the Coupa virtual card **Charges** to Netsuite as Netsuite **Credit Card** transaction.



Charge #204

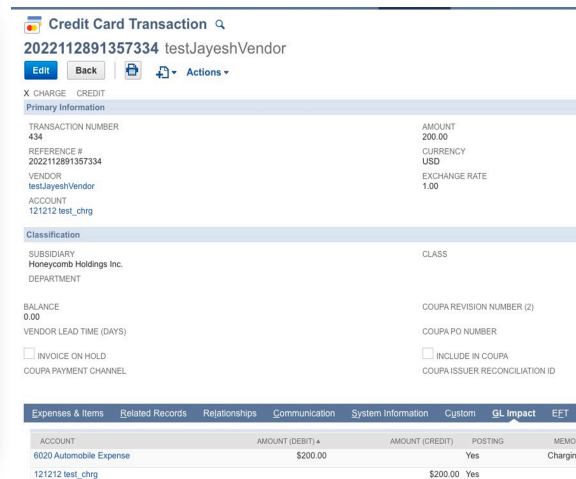
Charge Date: 11/27/22
Amount: 200.00 USD
Expense Preapproval: 28
Virtual card: *****(360)
External reference: 2022112891357334
Merchant Reference: Charging Merchant
Exported: Yes (Charge)
test static test
Custom Field 2: Honeycomb Holdings Inc.
Test Multiselect: None
Passthrough: None
Passthrough Check: No
Test Date Field: None
Custom Field 7: None
Invoice Needed?: No

2. At that point the Credit Card transaction will be booked against the holding account (in this example 6020) defined in Coupa as no final debit account yet available.



Script Deployment

Script: 6020 Automobile Expense
Script ID: 6020
Script Name: 6020 Automobile Expense
Script Type: User Defined Script
Script Location: /Script/6020 Automobile Expense
Script Status: Active
Script Version: 1.0
Script Created: 11/27/22
Script Modified: 11/27/22
Script Owner: Administrator
Script Permissions: Read, Write, Execute
Script Dependencies: None
Script Execution Log: View Log
Script Test Results: View Test Results



Credit Card Transaction

2022112891357334 testJayeshVendor

TRANSACTION NUMBER: 434
REFERENCE #: 2022112891357334
VENDOR: testJayeshVendor
ACCOUNT: 121212 test_chrg

AMOUNT: 200.00
CURRENCY: USD
EXCHANGE RATE: 1.00

CLASSIFICATION

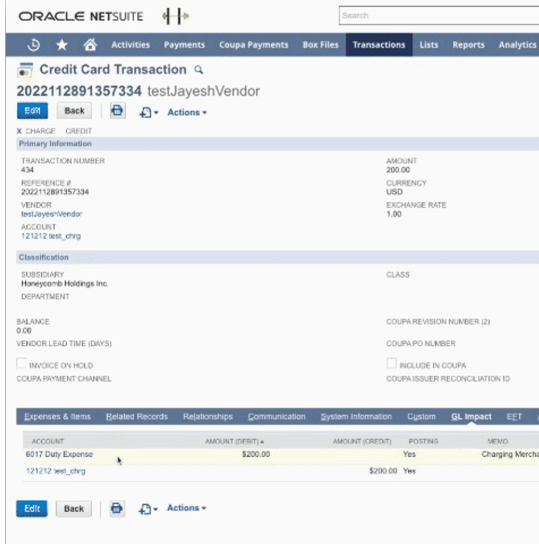
SUBSIDIARY: Honeycomb Holdings Inc.
DEPARTMENT: DEPARTMENT

BALANCE: 0.00
VENDOR LEAD TIME (DAYS): 0
INVOICE ON HOLD: ☐
COUPA PAYMENT CHANNEL: ☐

COUPA REVISION NUMBER (2):
COUPA PO NUMBER:
INCLUDE IN COUPA:
COUPA ISSUER RECONCILIATION ID:

ACCOUNT	AMOUNT (DEBIT) *	AMOUNT (CREDIT)	POSTING	MEMO
6020 Automobile Expense	\$200.00		Yes	Charging
121212 test_chrg		\$200.00	Yes	

3. Once Expense Report is approved for this PreApproved spend, the ER2ER integration will run and update the Credit Card transaction in Netsuite with the final Account (in this example 6017)
4. Netsuite Credit Card transaction will have the reference to the Coupa Expense Report and User. In Coupa there will be a link to the Netsuite Credit Card transaction for auditability.



Credit Card Transaction
2022112891357334 testJayeshVendor

Primary Information

TRANSACTION NUMBER	434	AMOUNT	200.00
REFERENCE #	2022112891357334	CURRENCY	USD
VENDOR	testJayeshVendor	EXCHANGE RATE	1.00
ACCOUNT	121212 test_chrg		

Classification

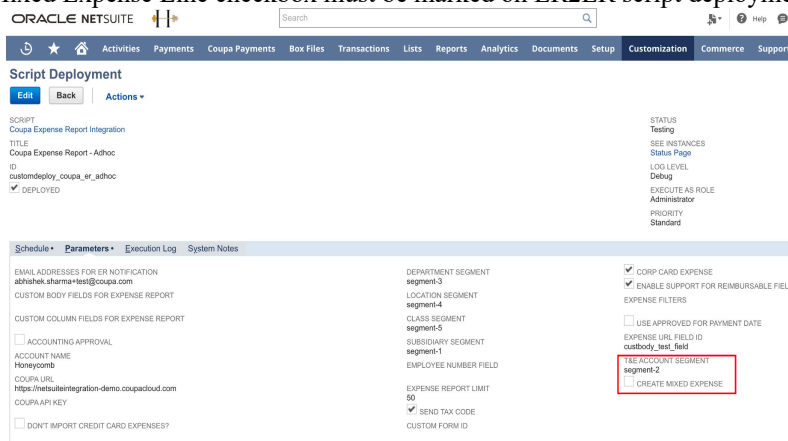
SUBSIDIARY	Honeycomb Holdings Inc.	CLASS	
DEPARTMENT			

Expenses & Items

ACCOUNT	AMOUNT (DEBIT)	AMOUNT (CREDIT)	POSTING	MEMO
6017 Duty Expense	\$200.00		Yes	Charging Merchant
121212 test_chrg		\$200.00	Yes	

NS Bundle Integration Options:

1. Expense Reports only contain PreApproval Expense Lines.
 - a. NS Bundle will not create a Netsuite Expense Reports for these Coupa PreApproval Expense reports.
 - b. Credit Card transaction will get updated with final debit account from approved Expense report
 - c. If Expense Report has mixed lines, NS bundle will throw an error.
2. Expense Report contains mixed lines (PreApproval, Corp card lines, Expense lines)
 - a. NS Bundle will not a Netsuite Expense Reports for all Expense Report Lines except the Coupa PreApproval Expense Line(s).
 - b. Create Mixed Expense Line checkbox must be marked on ER2ER script deployment



Script Deployment

SCRIPT: Coupa Expense Report Integration
TITLE: Coupa Expense Report - Adhoc
ID: customdeploy_coupa_er_adhoc
DEPLOYED: ☒

Parameters

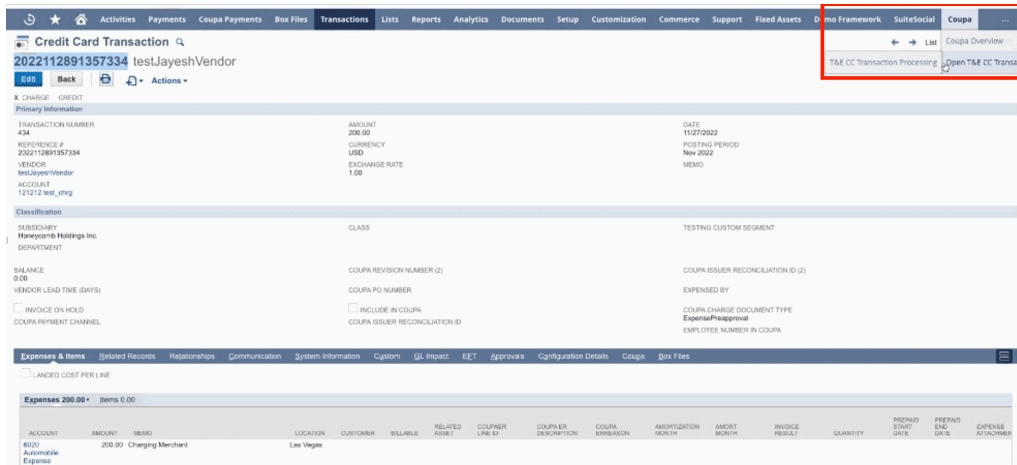
EMAIL ADDRESSES FOR ER NOTIFICATION	DEPARTMENT SEGMENT	<input checked="" type="checkbox"/> CORP CARD EXPENSE
abhishek.sharma@test@coupa.com	segment-3	<input checked="" type="checkbox"/> ENABLE SUPPORT FOR REIMBURSABLE FIELD
CUSTOM BODY FIELDS FOR EXPENSE REPORT	LOCATION SEGMENT	EXPENSE FILTERS
	segment-4	
CUSTOM COLUMN FIELDS FOR EXPENSE REPORT	CLASS SEGMENT	<input type="checkbox"/> USE APPROVED FOR PAYMENT DATE
	segment-5	EXPENSE URL FIELD ID
<input type="checkbox"/> ACCOUNTING APPROVAL	SUBSIDIARY SEGMENT	tbody_test_field
ACCOUNT NAME	segment-1	<input checked="" type="checkbox"/> THE ACCOUNT SEGMENT
Honeycomb	EMPLOYEE NUMBER FIELD	segment-2
COUPA URL	EXPENSE REPORT LIMIT	<input type="checkbox"/> CREATE MIXED EXPENSE
https://netsuiteintegration-demo.coupacloud.com	60	
COUPA API KEY	<input checked="" type="checkbox"/> SEND TAX CODE	
<input type="checkbox"/> DON'T IMPORT CREDIT CARD EXPENSES?	CUSTOM FORM ID	

Open T&E CC Transaction

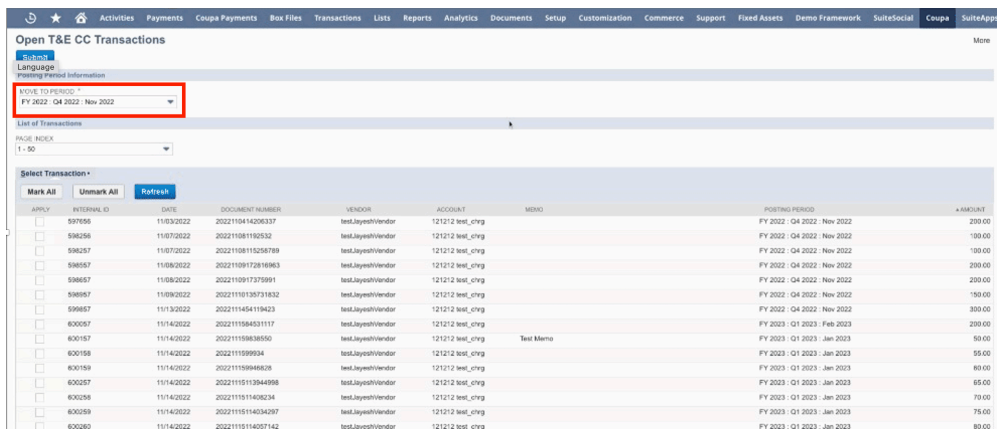
As Expense Reports are submitted and approved usually late or even in next financial period, Credit Card transaction must be in an open period, so NS Bundle can update final debit account when Expense Report is approved.

To help simplify/automate moving these outstanding Credit Card transaction to the next period, we created a suitelet.

1. Navigate to Coupa Menu >> Open T&E CC Transaction >> T&E CC transaction Processing
- 2.



3. The report will list all Credit Card transaction for Expense PreApproval which is still sitting on a Holding Account (Expense Report not yet been approved, and final account not yet set)



APPLY	INTERNAL ID	DATE	DOCUMENT NUMBER	VENDOR	ACCOUNT	MEMO	POSTING PERIOD	AMOUNT
<input type="checkbox"/>	597595	11/03/2022	2022110414206337	test.JayeshVendor	121212 test_cnyg		FY 2022 - Q4 2022 - Nov 2022	200.00
<input type="checkbox"/>	598256	11/07/2022	2022110811920532	test.JayeshVendor	121212 test_cnyg		FY 2022 - Q4 2022 - Nov 2022	100.00
<input type="checkbox"/>	598257	11/07/2022	20221108115258789	test.JayeshVendor	121212 test_cnyg		FY 2022 - Q4 2022 - Nov 2022	100.00
<input type="checkbox"/>	598557	11/08/2022	2022110917281963	test.JayeshVendor	121212 test_cnyg		FY 2022 - Q4 2022 - Nov 2022	200.00
<input type="checkbox"/>	598657	11/08/2022	2022110917375991	test.JayeshVendor	121212 test_cnyg		FY 2022 - Q4 2022 - Nov 2022	200.00
<input type="checkbox"/>	598657	11/09/2022	20221110135731832	test.JayeshVendor	121212 test_cnyg		FY 2022 - Q4 2022 - Nov 2022	100.00
<input type="checkbox"/>	598657	11/13/2022	2022111454119423	test.JayeshVendor	121212 test_cnyg		FY 2022 - Q4 2022 - Nov 2022	300.00
<input type="checkbox"/>	600057	11/14/2022	202211084531117	test.JayeshVendor	121212 test_cnyg		FY 2023 - Q1 2023 - Feb 2023	200.00
<input type="checkbox"/>	600187	11/14/2022	20221109836050	test.JayeshVendor	121212 test_cnyg	Test Memo	FY 2023 - Q1 2023 - Jan 2023	50.00
<input type="checkbox"/>	600199	11/14/2022	2022110999804	test.JayeshVendor	121212 test_cnyg		FY 2023 - Q1 2023 - Jan 2023	50.00
<input type="checkbox"/>	600199	11/14/2022	20221109998038	test.JayeshVendor	121212 test_cnyg		FY 2023 - Q1 2023 - Jan 2023	50.00
<input type="checkbox"/>	600257	11/14/2022	20221115113544956	test.JayeshVendor	121212 test_cnyg		FY 2023 - Q1 2023 - Jan 2023	65.00
<input type="checkbox"/>	600258	11/14/2022	2022111511409234	test.JayeshVendor	121212 test_cnyg		FY 2023 - Q1 2023 - Jan 2023	70.00
<input type="checkbox"/>	600259	11/14/2022	20221115114034297	test.JayeshVendor	121212 test_cnyg		FY 2023 - Q1 2023 - Jan 2023	75.00
<input type="checkbox"/>	600260	11/14/2022	20221115114057142	test.JayeshVendor	121212 test_cnyg		FY 2023 - Q1 2023 - Jan 2023	80.00

4. Select the period where you want to move the outstanding Credit Card transactions, then hit Submit

Note, this process can be automated fully as the background script can be set to run last day of the month and move all open Credit Card transactions for PreApproval to the next open period.