



## Coupa NetSuite Integration Guide

**Version 4.1.5**

**Last Updated: 2020-04-15 – Bundle 3.0.0**

## Contents

<b>Coupa NetSuite Integration Guide.....</b>	<b>1</b>
<b>Contents .....</b>	<b>2</b>
<b>P2P Dataflow and Integration points.....</b>	<b>4</b>
Overview.....	4
Overview with SIM.....	5
Integration Method.....	5
<b>Coupa P2P &amp; Expense Bundle .....</b>	<b>6</b>
Client Installation Overview .....	6
Activities.....	6
Locate the Coupa Bundle Script.....	7
Confirm contents of the Bundle .....	8
<b>Netsuite Vendor to Coupa Supplier .....</b>	<b>9</b>
Overview.....	9
Script Sequence Explained.....	9
Deployment Parameter Setup .....	9
Activities.....	9
Tips to help you succeed .....	9
Pre-Requisites .....	9
Script parameters.....	10
<b>Coupa SIM Supplier to NetSuite Vendor .....</b>	<b>15</b>
Overview.....	15
Script Sequence Explained.....	15
Deployment Parameter Setup .....	15
Activities.....	16
Tips to help you succeed .....	15
Pre-Requisites .....	15
Script parameters.....	16
Special Considerations with the SIM Bundle .....	19
<b>Netsuite to Coupa Exchange Rates.....</b>	<b>20</b>
Overview.....	20
Script Sequence Explained.....	20
Activities.....	20
Tips to help you succeed .....	20
Pre-Requisites .....	20
<b>Dynamic - Coupa Chart of Account .....</b>	<b>23</b>
Overview.....	23
Script Sequence Explained.....	23
Activities.....	23
Tips to help you succeed .....	23
Pre-Requisites .....	23
<b>Coupa Invoice to Netsuite Vendor Bill .....</b>	<b>29</b>
Overview.....	29
Script Sequence Explained.....	29
Activities.....	29



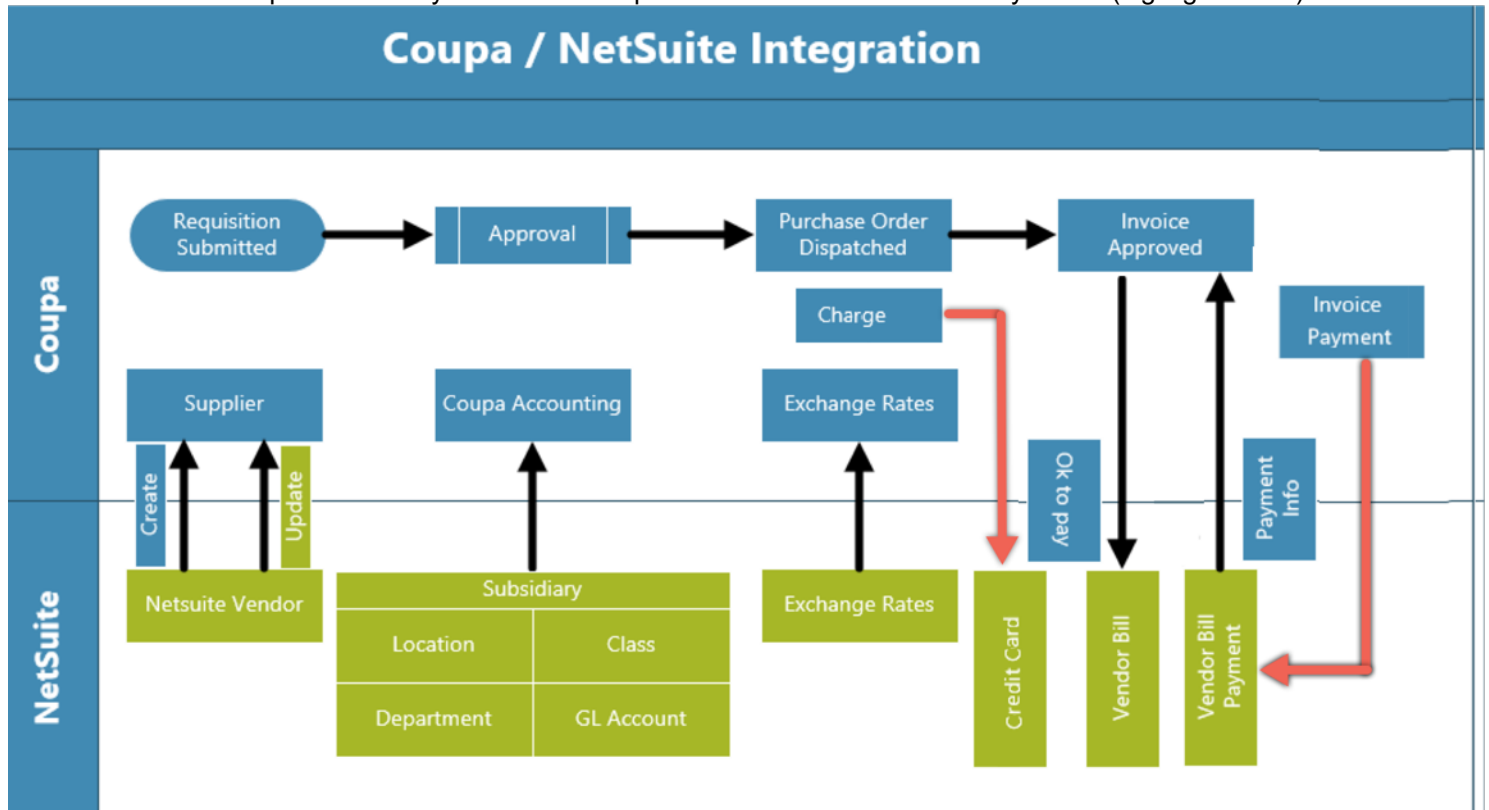
<i>Tips to help you succeed</i> .....	29
<i>Pre-Requisites</i> .....	29
See “Common Custom Field Examples” below .....	32
<b>Netsuite to Coupa Invoice Payments .....</b>	<b>36</b>
<i>Overview</i> .....	36
<i>Script Sequence Explained</i> .....	36
<i>Activities</i> .....	36
<i>Tips to help you succeed</i> .....	36
<i>Pre-Requisites</i> .....	36
<b>Coupa Expense Report to Netsuite Expense Report .....</b>	<b>40</b>
<i>Overview</i> .....	40
<i>Script Sequence Explained</i> .....	40
<i>Activities</i> .....	40
<i>Pre-Requisites</i> .....	40
<i>Tips to help you succeed</i> .....	40
<b>Coupa Expense Report Payment to Netsuite .....</b>	<b>45</b>
<i>Overview</i> .....	45
<i>Script Sequence Explained</i> .....	45
<i>Activities</i> .....	45
<i>Pre-Requisites</i> .....	45
<i>Tips to help you succeed</i> .....	45
<b>Coupa to Netsuite Expense Report to Vendor Bill .....</b>	<b>48</b>
<i>Overview</i> .....	48
<i>Script Sequence Explained</i> .....	48
<i>Activities</i> .....	48
<i>Tips to help you succeed</i> .....	48
<i>Pre-Requisites</i> .....	48
<b>Coupa to Netsuite Charge to Credit Card transaction .....</b>	<b>53</b>
<i>Overview</i> .....	53
<i>Script Sequence Explained</i> .....	53
<i>Activities</i> .....	53
<i>Tips to help you succeed</i> .....	53
<i>Pre-Requisites</i> .....	53
<b>Coupa to Netsuite Invoice Payment to Vendor Bill Payment .....</b>	<b>57</b>
<i>Overview</i> .....	57
<i>Script Sequence Explained</i> .....	57
<i>Activities</i> .....	57
<i>Tips to help you succeed</i> .....	57
<i>Pre-Requisites</i> .....	57

## P2P Dataflow and Integration points

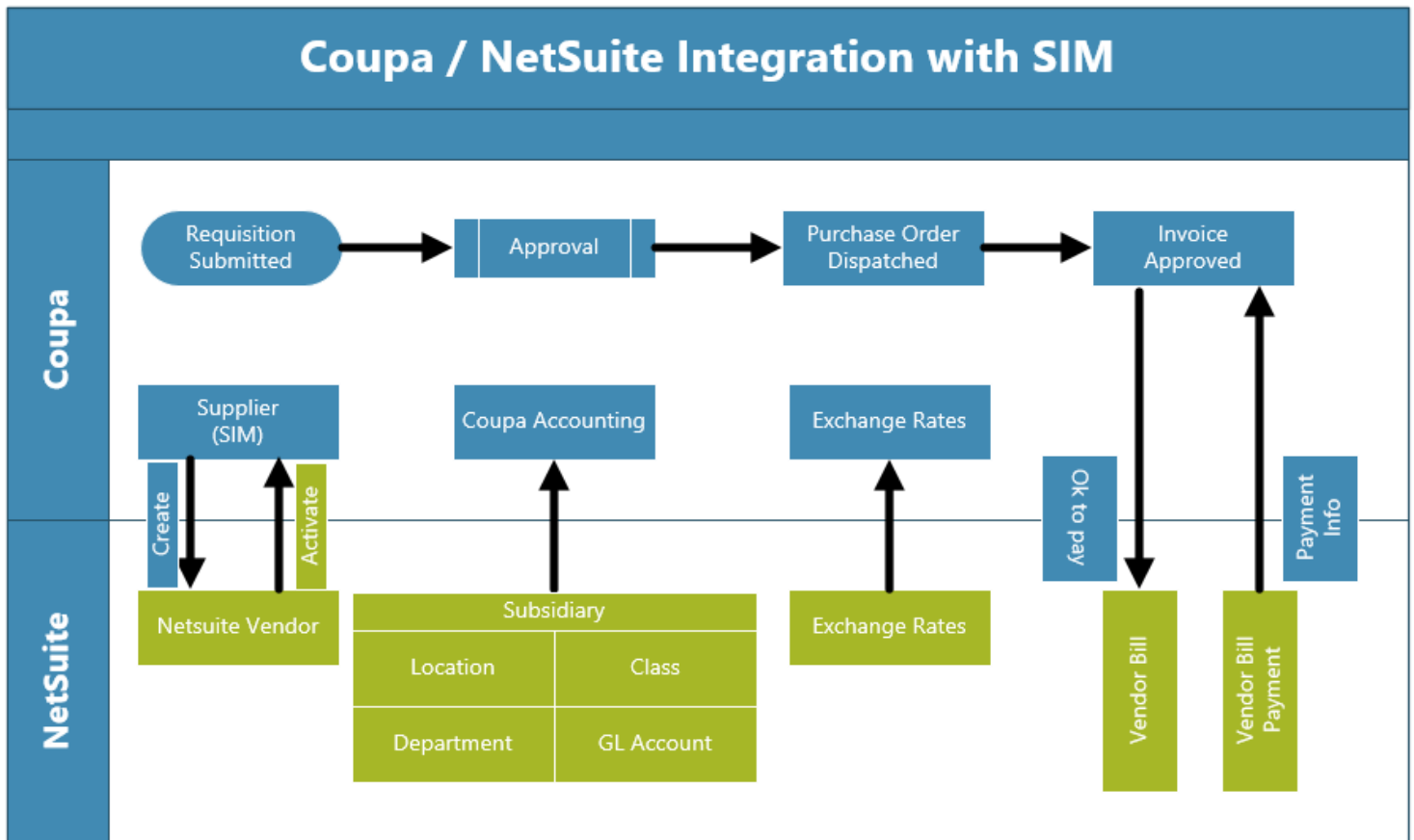
### Overview

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)

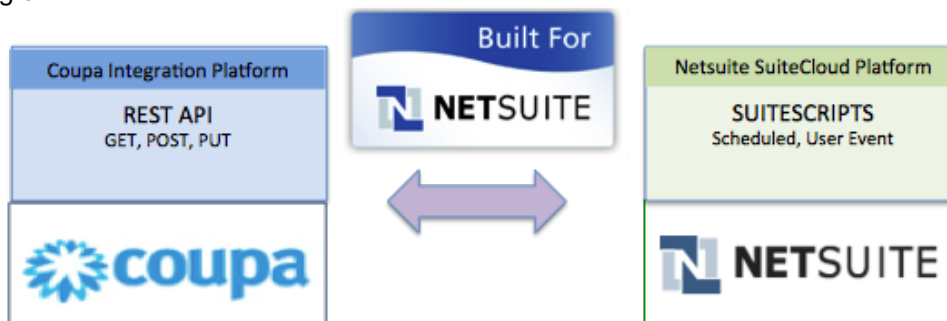


Now with SIM (Supplier Information Management), new supplier records can be initiated via a supplier information request from within Coupa. Once all the information is populated, the supplier record is created in NetSuite and simultaneously activated in Coupa. The diagram below indicates the additional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



### 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.



## Client Installation 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

# 1

## 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

[Installation Terms of Service](#)

NAME	BUNDLE ID
Coupa P2P + Expenses Bundle	84306

### Step 1.3

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

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





# 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 created
- You will need to baseline the Coupa Suppliers via the CSV UI Loader

# 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	Example	Id	Type	Comments	Example	Your Value
<b>Coupa URL*</b>	The Coupa instance the integration connects to	_vendor_url	Free-Form Text		https://abc.coupa host.com	
<b>Coupa API Key*</b>	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://xxxx.coupah ost.com/api_keys	4dtyr7657fgfh57 57fghhg	
<b>Error Email Notification Address</b>	Email addresses to which exception emails to be sent	_vendor_erroremailnotify	Free-Form Text		<a href="mailto:abc@abc.com">abc@abc.com</a> , <a href="mailto:abcd@abcd.com">abcd@abcd.com</a>	
<b>Account Name*</b>	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	
<b>Use Company Name as Display Name?</b>	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
<b>Phone/Fax Format</b>	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	
<b>Invoice Matching Level</b>	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.  If you want to deliberately default all records please use the following: none:custom_invoicelevel 2-way:custom_invoicelevel 3-way:custom_invoicelevel	customfield: custom_invoicelevel	
<b>Payment Method</b>	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. if there's only default, enter only default value	invoice:custom_paymentmethod	
<b>Invoice Emails</b>	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
<b>Send Invoice To Approvals</b>	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	
<b>Include in Coupa</b>	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	
<b>Allow Invoicing From CSN</b>	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:	
<b>Content Group Field</b>	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	
<b>PO Email Override</b>	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	
<b>Custom Fields Count*</b>	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
<b>Custom Field x (where x is an integer)</b>	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:NetSuitecustomfield:CoupaFieldType:NetsuiteFieldType:default	eligible-1099:is1099eligible:Boolean:Boolean	

## 2

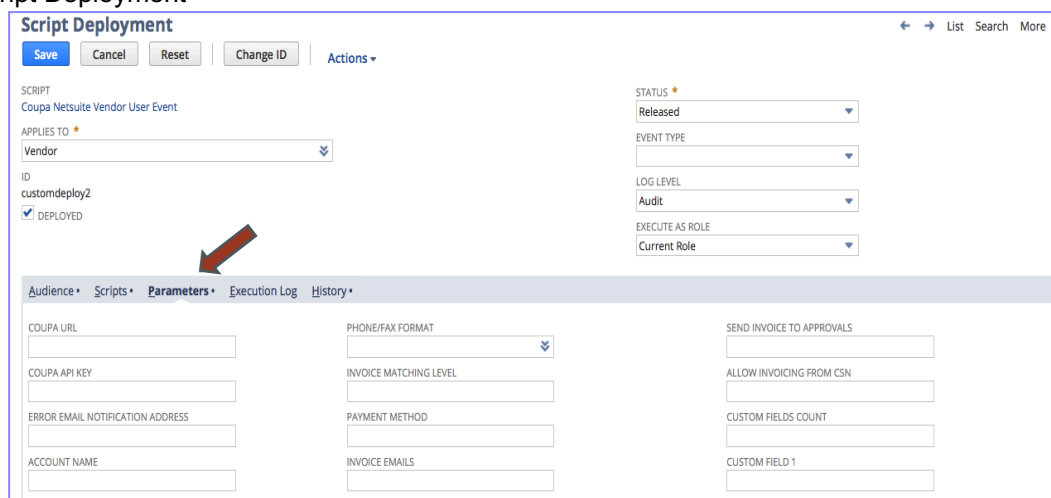
## Find and Define Deployment Parameter Values

**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



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.

## Deployment Parameter Setup

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

## 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 created with custom permissions



## 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
<b>Coupa URL*</b>	The Coupa instance the integration connects to	_coupa_sim_url	Free-Form Text	https://abc.coupahost.com
<b>Coupa API Key*</b>	The API key for Coupa's REST based API	_coupa_sim_apikey	Free-Form Text	4dtyr7657fgfh5757fghhg 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
<b>Vendor Mapping</b>	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
<b>Address Mapping</b>	Defines the mappings between coupa SIM custom fields and netsuite custom fields for Addresses	_coupa_sim_addresses_mapping	Free-Form Text	Coupacustomfield1==NScustomfield1;Coupacustomfield2/id==NScustomfield2
<b>Bank Rec Mapping</b>	Defines the mappings between coupa SIM custom fields and netsuite Bank record fields	_coupa_sim_bank_mapping	Free-Form Text	Coupacustomfield1==NScustomfield1;Coupacustomfield2/id==NScustomfield2
<b>Contact Rec Mapping</b>	Defines the mappings between coupa SIM custom fields and netsuite contact fields	_coupa_sim_contact_mapping	Free-Form Text	Coupacustomfield1==NScustomfield1;Coupacustomfield2/id==NScustomfield2
<b>Default Subsidiary ID</b>	Defines the default subsidiary ID if subsidiary is not mapped using custom below	_coupa_sim_default_sub	Free-Form Text	i.e. 5



Label	Description	Id	Type	Example
<b>Custom Subsidiary Coupa ID</b>	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
<b>“TO” Email Addresses for errors</b>	Where to email errors in this integration	_coupa_sim_error_to	Free-Form Text	tim@gogogo.com
<b>“FROM” Email address for errors</b>	Who will be displayed as the sender in those error emails	_coupa_sim_error_from	Free-Form Text	tim@gogogo.com
<b>Vendor “INCLUDE IN COUPA” Parameter ID</b>	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 inteegrations.	_coupa_sim_vendor_include	Free-Form Text	custentity_vend_include_in_coupa
<b>Payment File Format ID</b>	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
<b>Vendor Integration PO METHOD Parameter</b>	This is where to store the PO Method.	_coupa_po_method_parameter	Free-Form Text	Netsuitecustomfield1
<b>Use Display Name as Company Name</b>	This will override the Supplier Name with the whatever was specified in the “Display Name”	_coupa_sim_display_name	Free-Form Text	<Checkbox>

## 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*



## Script Deployment


← → List Search More

**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 ▼

**Schedule** • **Parameters** • [Execution Log](#) [System 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 to Coupa Exchange Rates

## 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.

## 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 created

# 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 Value
<b>Coupa URL*</b>	The Coupa instance the integration connects to	_coupa_fxrates_url	Free-Form Text		https://abc.coupahost.com	
<b>Coupa API Key*</b>	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	
<b>Error Email Notification Address</b>	Email addresses to which exception emails to be sent	_coupa_fxrates_erroremailnotify	Free-Form Text		<a href="mailto:abc@abc.com">abc@abc.com</a> , <a href="mailto:abcd@abcd.com">abcd@abcd.com</a>	
<b>Account Name*</b>	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	
<b>UTC Offset</b>	The <i>UTC offset</i> is the difference in hours and minutes from Coordinated Universal Time	_coupa_fxrates_utcoffset	Integer Number		10	

## 2

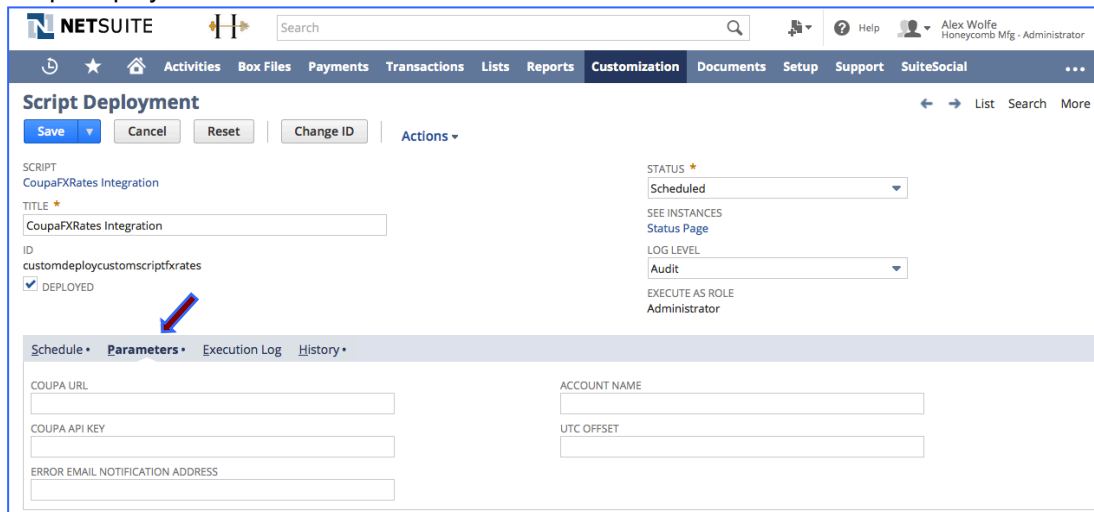
## Find and Define Deployment Parameter Values

**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

## 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 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> <span>Create</span> <span>Load from file</span> <span>Export to</span> <span>View</span> <span>All</span> <span>Advanced</span> <span>4000 sales</span> </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								
<div>Create Load from file Export to View All Advanced dept_americas</div>								
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

ViewNS

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 page15 | 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
<b>Coupa URL*</b>	The Coupa instance the integration connects to	_coupa_accs_url	Free-Form Text		https://abc.coupahost.com	
<b>Coupa API Key*</b>	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	
<b>Email Addresses for Notif</b>	Email addresses to which exception emails to be sent	_coupa_accs_notify_email	Free-Form Text		<a href="mailto:abc@abc.com">abc@abc.com</a> , <a href="mailto:abcd@abcd.com">abcd@abcd.com</a>	
<b>Include in Coupa</b>	Id of the custom field (on the record this deployment applies to) to allow/disallow the transmission of the record to Coupa	_coupa_accs_cstmfldincludeonly	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	
<b>Subsid Include in Coupa</b>	Id of the custom field (on SUBSIDIARY records) to allow/disallow the transmission of the subsidiary to Coupa	_coupa_accs_dynamic_sub_include	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	

## 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
<b>ROLES</b> <input checked="" type="checkbox"/> ALL ROLES <b>DEPARTMENTS</b> Administration Customer Support Engineering Engineering : Development Engineering : Product Management Engineering : Quality Assurance Marketing Operations Prof. Services Sales More...	<b>SUBSIDIARIES</b> 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... <b>GROUPS</b> <a href="#">Team ERP</a> <a href="#">Team Usability</a> <b>EMPLOYEES</b>	<input checked="" type="checkbox"/> ALL EMPLOYEES <b>PARTNERS</b> <input checked="" type="checkbox"/> ALL PARTNERS		

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

## 4

### Find and Define Deployment Parameter Values

**Navigation Path:** Customization | Scripting | Scripts

Find the “Coupa Dynamic COA 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 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 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

## 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 created
- Coupa Chart of Account SetupCoupa Suppliers with Netsuite Vendor ID

# 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 Value
<b>Coupa URL*</b>	The Coupa instance the integration connects to	_coupa_inv_url	Free-Form Text		https://abc.coupa.com	
<b>Coupa API Key*</b>	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	
<b>GL Account Segment</b>	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	
<b>Department Segment</b>	The accounting segment in Coupa that holds the Netsuite department account information	_coupa_inv_deptseg	Free-Form Text		segment-3	
<b>Subsidiary Account segment</b>	The accounting segment in Coupa that holds the Netsuite subsidiary account information	_coupa_inv_subseg	Free-Form Text		segment-1	
<b>Class Segment</b>	The accounting segment in Coupa that holds the Netsuite classification account information	_coupa_inv_classseg	Free-Form Text		segment-4	
<b>Location Segment</b>	The accounting segment in Coupa that holds the Netsuite location information	_coupa_inv_locseg	Free-Form Text		segment-5	
<b>Export From Invoice date</b>	Add filter criteria to pull invoices from Coupa with Invoice date >= to this date	_coupa_inv_frominvdate	Date		12/14/2013	
<b>Export To Invoice date</b>	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
<b>Use Updated at date</b>	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	
<b>Export From 'Updated at' date</b>	Add filter criteria to pull invoices from Coupa with 'Updated at' date >= to this date	_coupa_inv_from_updatedat_date	Date		12/14/2013	
<b>Export To 'Updated at' date</b>	Add filter criteria to pull invoices from Coupa with 'Updated at' date <= to this date	_coupa_inv_to_updatedat_date	Date		12/27/2013	
<b>Accounts Payable account number</b>	The Netsuite AP account number	_coupa_inv_actpayablenum	Free-Form Text		210000	
<b>notify this email address</b>	email addresses alerts to be sent to	_coupa_inv_email_notifications	Free-Form Text	multiple addresses can be separated by comma ','	<a href="mailto:xyz1@abc.com">xyz1@abc.com</a> , <a href="mailto:xyz2@abc.com">xyz2@abc.com</a>	
<b>Account Name</b>	Used in the alert/notification email subject to identify which Netsuite instance the alerts are coming from.	_coupa_inv_accounname	Free-Form Text		ABC Production	
<b>Send Tax Code</b>	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	
<b>Invoice URL link</b>	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
<b>Invoice Image URL Link</b>	URL to invoice image	_coupa_inv_image_link_field	Free-Form Text		http://<invoiceimage>.com	
<b>Coupa Version 12.06+?</b>	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	
<b>Dynamic Accounting ?</b>	Allows for billing strings to be created dynamically.	_coupa_inv_dynamicaccts	checkbox		T' i.e. checked	
<b>Limit exported records to</b>	Configuration to limit the invoices per invoice run	_coupa_inv_limit	Integer Number		5	
<b>Posting period cutoff</b>	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	
<b>Custom Field (header) 1</b>	Header level custom fields	_coupa_inv_custfieldheader1	Free-Form Text		<a href="#">See "Common Custom Field Examples" below</a>	
<b>Custom Field (header) 2</b>	Header level custom fields	_coupa_inv_custfieldheader1	Free-Form Text		<a href="#">See "Common Custom Field Examples" below</a>	
<b>Custom Field (header) 3</b>	Header level custom fields	_coupa_inv_custfieldheader1	Free-Form Text		<a href="#">See "Common Custom Field Examples" below</a>	
<b>Custom Field (header) 4</b>	Header level custom fields	_coupa_inv_custfieldheader1	Free-Form Text		<a href="#">See "Common Custom Field Examples" below</a>	
<b>Custom Field (header) 5</b>	Header level custom fields	_coupa_inv_custfieldheader1	Free-Form Text		<a href="#">See "Common Custom Field Examples" below</a>	



Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your Value
<b>Custom Field (line) 1</b>	Line level custom fields	_coupa_inv_custfieldline1	Free-Form Text		See “Common Custom Field Examples” below	
<b>Custom Field (line) 2</b>	Line level custom fields	_coupa_inv_custfieldline2	Free-Form Text		See “Common Custom Field Examples” below	
<b>Custom Field (line) 3</b>	Line level custom fields	_coupa_inv_custfieldline3	Free-Form Text		See “Common Custom Field Examples” below	
<b>Custom Field (line) 4</b>	Line level custom fields	_coupa_inv_custfieldline4	Free-Form Text		See “Common Custom Field Examples” below	
<b>Custom Field (line) 5</b>	Line level custom fields	_coupa_inv_custfieldline5	Free-Form Text		See “Common Custom Field Examples” below	
<b>Credit Custom Field 1 (Header)</b>	Header level custom field for credit notes	_custfield_crdt_header1	Free-Form Text		See “Common Custom Field Examples” below	
<b>Credit Custom Field 1 (Line)</b>	Line level custom field for credit notes	_custfield_crdt_line1	Free-Form Text		See “Common Custom Field Examples” below	
<b>CustomField(header) count</b>	Count of the number of header level custom fields configured	_coupa_inv_custfield_header_ct	Integer Number		5	
<b>Custom field (line) count :</b>	Count of the number of line level custom fields configured	_coupa_inv_custfield_line_cnt	Integer Number		3	
<b>Credit Custom Field Count (Header)</b>	Count of the number of header level custom fields configured for Credit Notes	_customfield_crdt_header_count	Integer Number		1	
<b>Credit Custom Field Count (Line)</b>	Count of the number of line level custom fields configured for Credit Notes	_customfield_crdt_line_count	Integer Number		1	
<b>Support Void</b>	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
<b>Coupa Pay Enabled?</b>	Enables Payment hold for coupa pay invoices	<code>_coupa_inv_coupa_pay_enabled</code>	checkbox	Puts Coupa Pay invoice on Payment hold in Netsuite if set to true		

Common Custom Field Examples:

Script Definition					Script Deployment
Label	Description	Type	Comments	Example	Your Value
<b>Amortization Schedule</b>	Pass the amortization schedule defined on the Coupa Invoice	Lookup		<code>amort-schedule:amortizationsched:Lookup:Value</code>	
<b>Amortization Start Date</b>	The date when the amortization should start	Date		<code>amort-start:amortizstartdate:Date:Value</code>	
<b>Amortization End Date</b>	The date when the amortization should end	Date		<code>amort-end:amortizationenddate:Date:Value</code>	
<b>isBillable</b>	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)	<code>isBillable:isBillable:Text:Value</code>	
<b>Coupa Pay Enabled?</b>	Sets the script behavior for Invoice Paid in Coupa via Coupa Pay	checkbox	If set to True, Invoice paid in coupa via Coupa Pay will be put on "Payment Hold" to avoid double payments.		

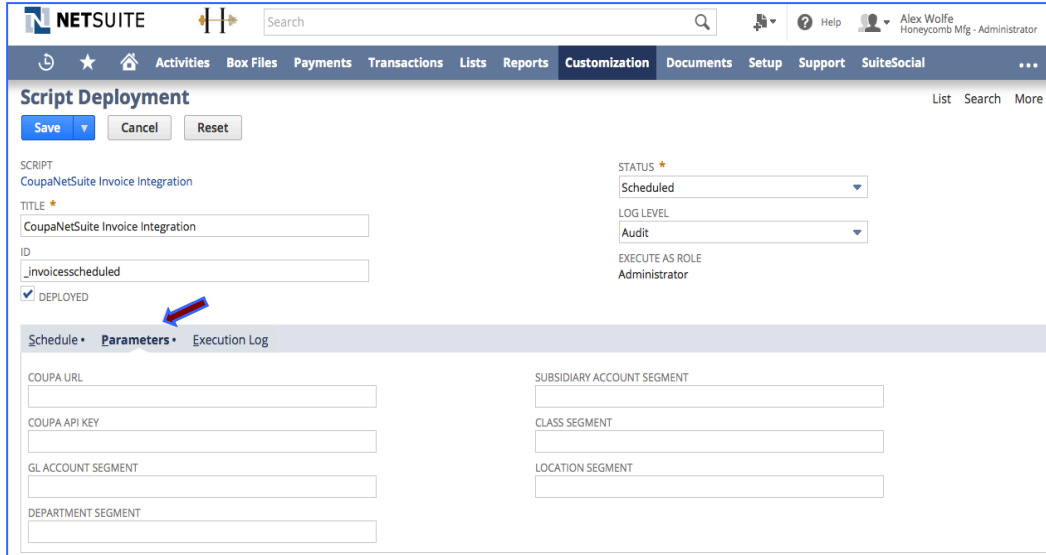
## 2

## 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 Script Deployment interface. The 'Script Deployment' page is open, displaying the 'CoupaNetSuite Invoice Integration' script. The 'Parameters' tab is selected, showing a list of parameters to be configured. A red arrow points to the 'Parameters' tab. The parameters are organized into two columns:

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

Other visible fields include:

- SCRIPT: CoupaNetSuite Invoice Integration
- TITLE: CoupaNetSuite Invoice Integration
- ID: \_invoicesscheduled
- DEPLOYED: ☒
- STATUS: Scheduled
- LOG LEVEL: Audit
- EXECUTE AS ROLE: Administrator

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

# Netsuite to Coupa Invoice Payments

## 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

## 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 created

# 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
<b>Coupa URL*</b>	The Coupa instance the integration connects to	_coupa_pay_url	Free-Form Text		https://abc.coupa host.com	
<b>Coupa API Key*</b>	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	
<b>Export From Payment Date</b>	Add filter criteria to pull payments from Netsuite created >= to this date	_coupa_pay_from paydate	Date		12/14/2013	
<b>Export To Payment Date</b>	Add filter criteria to pull payments from Netsuite created <= to this date	_coupa_pay_topa ydate	Date		12/27/2013	
<b>Error Email Notification Address</b>	Email addresses to which exception emails to be sent	_coupa_pay_ema il_notifications	Free-Form Text		<a href="mailto:abc@abc.com">abc@abc.com</a> , <a href="mailto:abcd@abcd.com">abcd@abcd.com</a>	
<b>Account Name*</b>	As per standard, it will be company name with which environment its been setup (Test or Prod)	_coupa_pay_accoun tname	Free-Form Text	<Company Name> <Test/Prod>	Coupa Test	
<b>Process Records From #</b>	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	

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your Values
<b>Process Records To #</b>	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_torecords	Integer Number	Default is 250	250	
<b>Bring Payment to NS ASAP</b>	when true, we import all statuses from payment initiated onward	custscript_coupa_pymt_status_filter	checkbox			
<b>Custom API Arguments</b>	lets you specify custom query parameters, does not let you change status	custscript_coupa_pmnt_cust_args	Text		return_object=shallow, limit=5	
<b>Static Department</b>	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			
<b>Static Classification</b>	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			

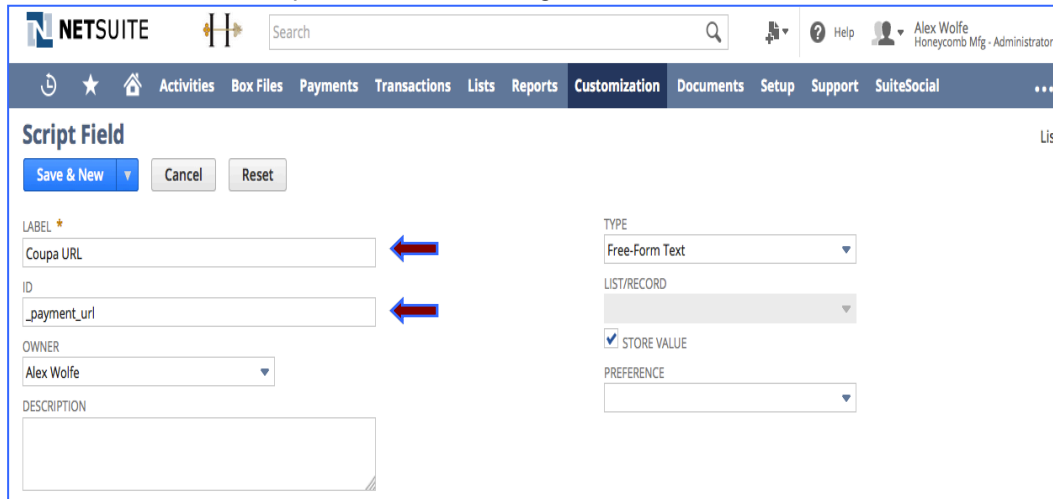
Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your Values
<b>Static Location</b>	specify a static location internal id for the payment header, or put 'retrieveValue' which will lookup the value from the first vendor bill	custscript_coupa_pymt_headlocn	Text			

## 2 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 various menu items like 'Activities', 'Box Files', 'Payments', 'Transactions', 'Lists', 'Reports', 'Customization', 'Documents', 'Setup', 'Support', and 'SuiteSocial'. The 'Customization' menu is active. The 'Script Field' form has a 'Save & New' button, 'Cancel', and 'Reset' buttons. The form fields are: LABEL (Coupa URL), ID (\_payment\_url), OWNER (Alex Wolfe), and DESCRIPTION. On the right side, there are dropdowns for TYPE (Free-Form Text), LIST/RECORD, and PREFERENCE, and a checked checkbox for STORE VALUE. Red arrows point to the LABEL and ID fields.

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)

## 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**

## Pre-Requisites

1. Coupa P2P + Expenses bundle deployed
2. Coupa API key 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)

## 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.
  - 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.
  - NetSuite Users should clear the Coupa ER Number field if they copy an existing ER that was brought in from Coupa. Subsequent updates to original ER may have an issue as it will find 2 ERs in NetSuite with same Coupa Number
  - Once the ER is ‘Paid’ in NetSuite subsequent changes to Coupa ER may fail to be brought over to NetSuite. Received ER will not allow the expense category to be reduced or cancel a line.
9. 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





8. Coupa commodity and NetSuite mapping (names must match identically)

same time.

# 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	<code>_coupa_er_url</code>	freeform text		2e528d48825a8fe5a6a2a67972a2d8d21f5j3sdf4 8	
Coupa API Key*	The API Key for Coupa's REST based API	<code>_coupa_er_apikey</code>	freeform text		http://netsuiteintegration-demo.coupacloud.com/	
Error email notification addresses	List of emails to receive errors generated by the script execution	<code>_coupa_er_email_addr_notify</code>	freeform text	Separate the emails with a comma and no space	<a href="mailto:abc1@xyz.com">abc1@xyz.com</a> , <a href="mailto:abc2@xyz.com">abc2@xyz.com</a>	
Account Name*	As per standard, it will be company name with which environment its been setup (Test or Prod)	<code>coupa_er_acccountname</code>	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.	<code>_coupa_er_creditcardskip</code>	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	<code>_coupa_er_creditcardskip</code>	checkbox			

Location Segment	The accounting segment in Coupa that holds the Netsuite location information	<code>_coupa_er_classseg</code>	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_couplerequester;requested by-email==custbody_couplerequesteremail</code>	

Custom column s 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==cust col_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	<code>_coupa_er_send_taxcode</code>	checkbox	When using this option, the taxcodes in Coupa need to be stored in the following format: netsuite name:netsuite internal id. 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	<code>custscript_coupa_er_coupapay</code>	Check Box	CoupaPay is used to Pay ER		

## 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	COUPA URL	CLASS SEGMENT
<input type="text"/>	<input type="text"/>	<input type="text"/>
CUSTOM BODY FIELDS FOR EXPENSE REPORT	COUPA API KEY	SUBSIDIARY SEGMENT
<input type="text"/>	<input type="text"/>	<input type="text"/>
CUSTOM COLUMN FIELDS FOR EXPENSE REPORT	<input checked="" type="checkbox"/> DONT IMPORT CREDIT CARD EXPENSES?	EMPLOYEE NUMBER FIELD
<input type="text"/>	DEPARTMENT SEGMENT	<input type="text"/>
<input type="checkbox"/> ACCOUNTING APPROVAL	LOCATION SEGMENT	EXPENSE REPORT LIMIT
ACCOUNT NAME	<input type="text"/>	<input type="text"/>
<input type="text"/>		<input type="checkbox"/> 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

## 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	<a href="mailto:abc1@xyz.com">abc1@xyz.com</a> , <a href="mailto:abc2@xyz.com">abc2@xyz.com</a>	
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			

## 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 to Netsuite Expense Report to 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 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



# 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 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		<a href="https://abc-demo.coupadev.com">https://abc-demo.coupadev.com</a>	
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	custscript_coupa_er2vb_updatedate	checkbox		True	

	filter expense reports					
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_er2v b_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_er2v b_to_updated	Date		2020-1-15	
External Source Filter Value	Used to skip Corporate card files	custscript_coupa_er2v b_extsrcfilter	Freeform-text		corporate_credit_card_amex	
Custom Employee Number	Used to denote a custom field on the Expensed By user in Coupa that should be used instead of employee number to match entityid (Vendor ID)	custscript_coupa_er2v b_employee_field	Freeform-text		netsuite-vendor-id	
Account Segment	Xml tag for the GL Account segment	custscript_coupa_er2v b_glaccseg	Freeform-text		segment-2	
Line Custom	Line level custom field	custscript_coupa_er2v b_infields	Text-area		coupa-xml-tag==netsuitet	

Field Mappings	mappings in the format: coupa-xml-tag==netsuitefield;				ag1;coupa-xml-tag2==netsuite tag2;	
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==netsuite ag1;coupa-xml-tag2==netsuite tag2;	
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_classeg			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			<a href="mailto:abc@example.com">abc@example.com</a> , <a href="mailto:xyz@example.com">xyz@example.com</a>	
Error From ID	Internal ID of the netsuite Employee that the emails	custscript_coupa_er2vb_errorfrm			1	

	should be from					
--	----------------	--	--	--	--	--

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 to Netsuite Charge to 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.

## 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

### 1 Review Parameter List

### 2 Find and Define Deployment Parameter Values

# 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 \*

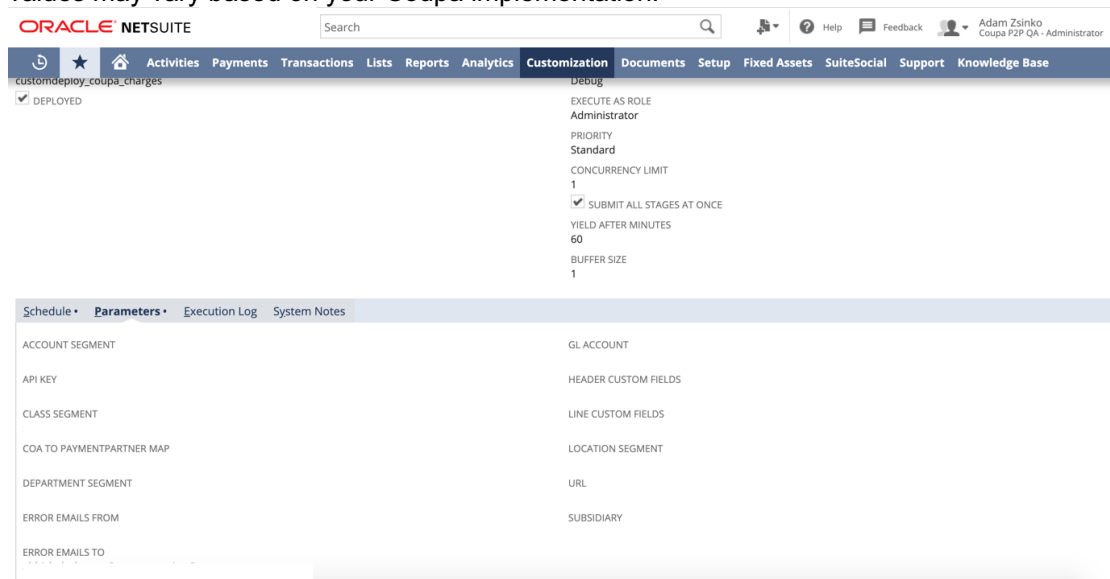
Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your values
Email Errors To	Comma separated list of emails to receive error notifications	custscript_coupa_chrg_errorto	freeform text		test@coupa.com	
Email Errors From	NS Internal ID of the user the emails should be 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	custscript_coupa_chrg_forcesave	CheckBox			

	to skip validations					
CoA ID to Vendor ID	This is a mapping 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 number for the Account Payable that should be set on the transaction linked to the Coupa COA	custscript_coupa_chrg_glacc	Freeform-text		1==245	

## 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.



Parameter	Value
EXECUTE AS ROLE	Administrator
PRIORITY	Standard
CONCURRENCY LIMIT	1
SUBMIT ALL STAGES AT ONCE	<input checked="" type="checkbox"/>
YIELD AFTER MINUTES	60
BUFFER SIZE	1
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 to Netsuite Invoice Payment to Vendor Bill Payment

## Overview

Customers with Coupa Pay PO 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 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 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 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)

# 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 \*

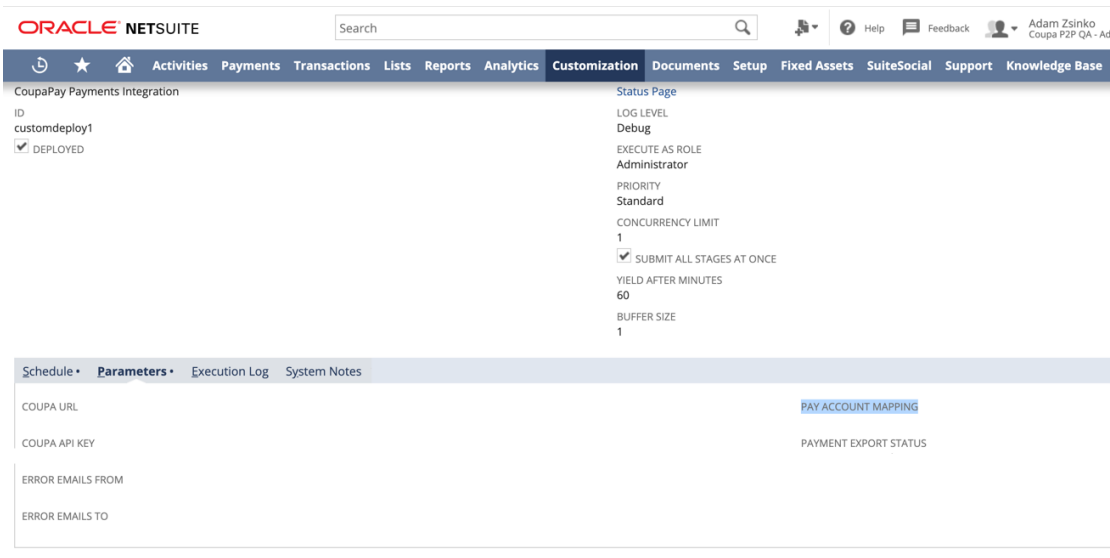
Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your values
Email Errors To	Comma separated list of emails to receive error notifications	custscript_coupa_chrg_errorto	freeform text		test@coupa.com	
Email Errors From	NS Internal ID of the user the emails should be 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	custscript_coupa_pymt_accmap			1==3412;2==4812;3==452	

	account representing the bank account that these payments should be tied to.					

## 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 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.