



## Coupa NetSuite **P2O** Bundle Integration Guide

**Doc Version 9.0.0 Last Updated: 2023-08-14**

## Table of Contents

<b><i>Doc Version 6.0.0 Last Updated: 2022-08-29</i></b> .....	<b>1</b>
<b><i>P2P Dataflow and Integration points</i></b> .....	<b>5</b>
<b>Overview</b> .....	<b>5</b>
<b>Overview with SIM</b> .....	<b>Error! Bookmark not defined.</b>
<b>Integration Method</b> .....	<b>6</b>
<b>Client Installation/Upgrade Overview</b> .....	<b>7</b>
<b>Activities</b> .....	<b>7</b>
<b>Locate the Coupa Bundle Script</b> .....	<b>7</b>
Step 1.1.....	7
<b>Confirm contents of the Bundle</b> .....	<b>8</b>
Step 1.3.....	8
3. Login to Netsuite and go to Setup > Company > General Preferences > Custom Preference.....	9
<b><i>Coupa SIM Supplier to NetSuite Vendor</i></b> .....	<b>10</b>
<b>Overview</b> .....	<b>10</b>
Script Sequence Explained .....	10
Deployment Parameter Setup.....	10
Activities.....	11
Tips to help you succeed .....	10
Pre-Requisites .....	10
Review Parameter List.....	11
<b>Script parameters</b> .....	<b>11</b>
Special Considerations with the SIM Bundle.....	15
<b><i>Netsuite Vendor to Coupa Supplier</i></b> .....	<b>Error! Bookmark not defined.</b>
<b>Overview</b> .....	<b>Error! Bookmark not defined.</b>
Script Sequence Explained .....	<b>Error! Bookmark not defined.</b>
Deployment Parameter Setup.....	<b>Error! Bookmark not defined.</b>
Activities.....	<b>Error! Bookmark not defined.</b>
Tips to help you succeed .....	<b>Error! Bookmark not defined.</b>
Pre-Requisites .....	<b>Error! Bookmark not defined.</b>
Review Parameter List.....	<b>Error! Bookmark not defined.</b>
<b>Script parameters</b> .....	<b>Error! Bookmark not defined.</b>
<b><i>Netsuite to Coupa Exchange Rates 2.0</i></b> .....	<b>30</b>
<b>Overview</b> .....	<b>30</b>
<b>Script Sequence Explained</b> .....	<b>30</b>
<b>Activities</b> .....	<b>30</b>



Tips to help you succeed .....	30
Pre-Requisites .....	30
Review parameter list .....	31
<b>Find and Define Deployment Parameter Values .....</b>	<b>32</b>
<b><i>Dynamic - Coupa Chart of Account .....</i></b>	<b>33</b>
<b>Overview .....</b>	<b>33</b>
Script Sequence Explained .....	33
Activities.....	33
Tips to help you succeed .....	33
Pre-Requisites .....	33
Define Coupa Lookups and Coupa Chart of Accounts .....	34
Review Parameter List .....	37
Review Scripts .....	38
Find and Define Deployment Parameter Values .....	39
<b><i>Coupa Purchase Order to Netsuite Purchase Order.....</i></b>	<b>40</b>
<b>Overview .....</b>	<b>40</b>
<b>Script Sequence Explained .....</b>	<b>40</b>
<b>Activities .....</b>	<b>40</b>
Tips to help you succeed .....	40
Pre-Requisites .....	40
<b>Review parameter list.....</b>	<b>41</b>
<b>Find and Define Deployment Parameter Values .....</b>	<b>42</b>
<b><i>Coupa Receipt to Netsuite Item Receipt .....</i></b>	<b>44</b>
<b>Overview    Tips to help you succeed.....</b>	<b>44</b>
<b>Pre-Requisites .....</b>	<b>44</b>
<b>Activities .....</b>	<b>44</b>
<b>Review Parameter List .....</b>	<b>44</b>
<b>1 Review Parameter List .....</b>	<b>45</b>
<b>Find and Define Deployment Parameter Values .....</b>	<b>45</b>
Overview .....	47
Script Sequence Explained .....	47
Activities.....	47
<b>Review Parameter List .....</b>	<b>47</b>
<b>Find and Define Deployment Parameter Values .....</b>	<b>47</b>
Tips to help you succeed .....	47
Pre-Requisites .....	47
<b>1 Review Parameter List .....</b>	<b>47</b>



**Find and Define Deployment Parameter Values ..... 50**

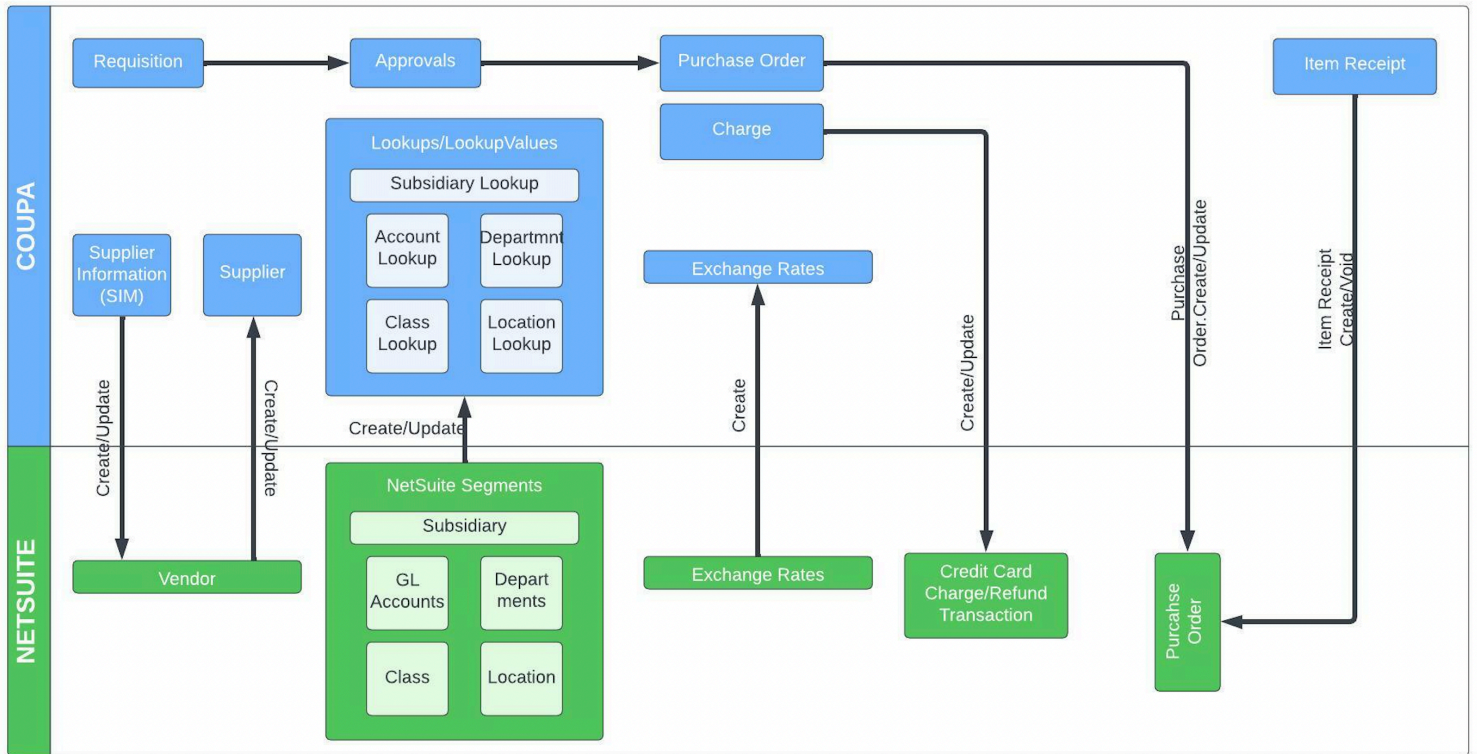
# P2O Dataflow and Integration points

## Overview

The diagram below indicates the integration points between Netsuite and Coupa for the P2O process. Typically we have master data - suppliers, Netsuite GL Account attributes, exchange rates and transaction data – PO and Receipt from Coupa to Netsuite.

Note: 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.

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.

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

## 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 Month
			Lists, Records, & Fields ▶				
			Forms ▶				
NAME		JOB	Scripting ▶			SUPERVISOR	
cott Beveridge		Pres	Plug-ins ▶			Mark Jones	
rad M Sparling		Engi	Centers and Tabs ▶			Miles Grey	
Mary Redding		Sale	SuiteBundler ▶			Create Bundle ▶	
Alex Wolfe		VP S				Search & Install Bundles ▶	
am Smith		Purchasing				Bundle Audit Trail	
ave Herrington		Production Manager					
lark Koozer		Machine Operator					

### Step 1.2

Search for “Coupa P2O + Expenses” and click on Bundle

### Confirm contents of the Bundle

### Step 1.3

Switch to the Components View and verify the contents of the P2O bundle **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**

The screenshot shows the Oracle NetSuite interface. The top navigation bar includes 'Setup', 'Commerce', 'Support', and 'SuiteSocial'. The 'Setup' menu is expanded, showing a list of options. The 'General Preferences' option is highlighted with a red rectangular box. Other visible options include 'Company Information', 'Enable Features', 'Subsidiary Settings Manager', 'Rename Records/Transactions', 'Auto-Generated Numbers', 'States/Provinces/Countries', 'Countries', 'Business Activity Statement', 'Application Publishers', and 'Users/Roles'.

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

The screenshot shows the 'Custom Preferences' page in Oracle NetSuite. The 'Custom Preferences' tab is selected and highlighted with a red box. The page is divided into several sections. The 'Coupa P2P + Expenses Bundle' section is highlighted with a red box and contains the following configuration options:

- INTEGRATE INVOICE FOR THE VC
- COUPA OIDC CLIENT IDENTIFIER: [Redacted]
- COUPA OIDC CLIENT SECRET: 5c58ea8a90930ca5efe2343a79b2f8f11b7
- COUPA INSTANCE URL: https://netsuiteintegration-

Other sections visible include 'Check Inventory', 'KB Rating & Case Customizations', and 'General'.

## 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.

### 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

### 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.

### 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
<b>Coupa URL*</b>	The Coupa instance the integration connects to	_coupa_sim_url	Free-	https://abc.coupahost.com	
<b>Coupa API Key*</b>	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	
<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	_coupa_sim_vendor_mapping	Free-Form Text	po-transmission-method/external-ref-num==custentity_coupa_po_method_rec;default-commodity/id==custentity_coupa_default_commodity	

Label	Description	Id	Type	Example	Comment
	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.				
<b>Address Mapping</b>	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	
<b>Bank Rec Mapping</b>	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	
<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	
<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 integrations.	_coupa_sim_vendor_include	Free-Form Text	custentity_vendor_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	_coupa_sim_payment_format	Free-Form Text	netsuite-payment-file-format/external-ref-num	

Label	Description	Id	Type	Example	Comment
	vendor. Newly created bank details will be marked as Primary, and the existing primary record will be demoted.				
<b>Vendor Integration PO METHOD Parameter</b>	This is where to store the PO Method.	<code>_coupa_po_method_parameter</code>	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"	<code>_coupa_sim_display_name</code>	Free-Form Text	<Checkbox>	
<b>Multiselect Subsidiary</b>	Add support for Multi subsidiary on vendor recording in NS	<code>custscript_coupa_sim_subselect</code>	Checkbox		Added 6.0.1
<b>*Skip bank Account Details</b>	Skip integrating bank account details	<code>_coupa_sim_bank_syncs</code>	Free Form Text	T/F	Updated 10.0.0
<b>*Custom Bank Module</b>	AvidXBank support	<code>_sim_custom_bank</code>	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

|  | [Actions](#) ▾

SCRIPT  
Coupa SIM Integration

TITLE \*

ID  
customdeploy\_coupa\_sim\_scheduled

DEPLOYED

STATUS \*

SEE INSTANCES  
[Status Page](#)

LOG LEVEL

EXECUTE AS ROLE  
Administrator

PRIORITY \*

CONCURRENCY LIMIT

SUBMIT ALL STAGES AT ONCE

YIELD AFTER MINUTES \*

BUFFER SIZE \*

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

COUPA URL <input type="text"/>	CUSTOM SUBSIDIARY COUPA ID <input type="text"/>
COUPA API KEY <input type="text"/>	"TO" EMAIL ADDRESSES FOR ERRORS <input type="text"/>
VENDOR MAPPING <input type="text"/>	"FROM" EMAIL ADDRESS FOR ERRORS <input type="text"/>
ADDRESS MAPPING <input type="text"/>	VENDOR "INCLUDE IN COUPA" PARAMETER ID <input type="text"/>
BANK REC MAPPING <input type="text"/>	PAYMENT FILE FORMAT ID <input type="text"/>
CONTACT REC MAPPING <input type="text"/>	VENDOR INTEGRATION PO METHOD PARAMETER <input type="text"/>
DEFAULT SUBSIDIARY ID <input type="text"/>	<input checked="" type="checkbox"/> 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 Vendor 2.0 to Coupa Supplier

## Overview

**PLEASE UPGRADE YOUR VENDOR SCRIPT TO VENDOR 2.0 SCRIPT. Vendor 1.0 script will be deprecated and will NOT be maintained from release 15.0.0 onwards.**

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”/”delete” events. Based on the event type either a new Supplier is created in Coupa or an existing Supplier in Coupa is updated.

On the Vendor record in Netsuite you can set if you want to integrate the record to Coupa.

## 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

## 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



**1** Review Parameter List

**2** Find and Define Deployment Parameter Values

## Review Parameter List Vendor 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	Explanation	Id	Type	Comments	Example	Your Value
<b>Coupa Instance URL*</b>	The Coupa instance the integration connects to	custscript_coupa_vi_mr_host_url	hyperlink		https://abc.coupa.com	
<b>Error Email Receiver*</b>	Error recipient	_custscript_coupa_vi_mr_email_receiver	Free-Form Text		<a href="mailto:abc@abc.com">abc@abc.com</a> , <a href="mailto:abcd@abcd.com">abcd@abcd.com</a>	
<b>Error Email Sender</b>	Error email sender	custscript_coupa_vi_mr_email_sender	List/Record			
<b>Account Name*</b>	As per standard, it will be company name with which environment its been setup (Test or Prod)	custscript_coupa_vi_mr_account_name	Free-Form Text	<Test/Prod>	Coupa Test	
<b>Include in Coupa Field ID</b>	Configuration that allows/disallows a supplier record to be sent to Coupa	custscript_coupa_vi_mr_integration_flag	Free-Form Text	Internal id of an existing field or custom field.  The custom field on the vendor record should be a checkbox		
<b>Invoice Matching Level</b>	Invoice matching level that is defined for this supplier.	custscript_coupa_vi_mr_invoice_match_level	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	

Script Definition					Script Deployment	
Label	Explanation	Id	Type	Comments	Example	Your Value
<b>Payment Method</b>	Payment method	custscript_coupa_vi_mr_payment_method	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	
<b>Invoice Emails IDs</b>	Registered email addresses allowed to send invoices	custscript_coupa_vi_mr_invoice_emails	Free-Form Text			
<b>Send Invoice To Approvals</b>	If yes, then invoices will all be sent thru approvals, regardless of total amount.	custscript_coupa_vi_mr_route_inv_for_app	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>Allow Invoicing From CSP</b>	If yes, then the supplier can create invoices against their POs or Contracts in CSN	custscript_coupa_vi_mr_allow_csp_invoice	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 ID</b>	Allows the supplier record in Coupa to be associated with content group(s)	custscript_coupa_vi_mr_content_grp_value	Free-Form Text	the value should be the internal id of the field that holds the name of the content field. Multiple content	_coupa_content_group	

Script Definition					Script Deployment	
Label	Explanation	Id	Type	Comments	Example	Your Value
				groups are delimited by , (comma)		
<b>Use Company Name as Display Name</b>	Use Company Name as Display Name in Coupa	custscript_coupa_vi_mr_use_display_name	Check-box			
<b>Skip Syncing Phone Number to Coupa</b>	Skip phone number integration	custscript_coupa_vi_mr_skip_phone_number	Check-box			
<b>Override PO Method</b>	Allows overriding the standard mapping for the PO Email field	custscript_coupa_vi_mr_po_methd_override	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>Skip PO Method Update</b>	Skip Po Method integration on Vendor Update	custscript_coupa_vi_mr_skip_pomethod	Check-box			
<b>PO Email Override</b>		custscript_coupa_vi_mr_po_email_override	Free-Form Text			
<b>Custom Fields Count*</b>	Define the number of custom fields to be integrated.	_vendor_customfieldscount	Integer Number	0,1,2,...	0	
<b>Content Group Field ID</b>		custscript_coupa_vi_mr_content_grp_value	Free-Form Text			
<b>Content Group Mapping</b>	Map subsidiary to Coupa Content Group	custscript_coupa_vi_mr_content_grp_map	Free-Form Text		NetSuite_Subsidiary_Name1==Coupa_Content_Group1;NetSuite_Subsidiary_Name2==Coupa_Content_Group2	
<b>Custom Body Field Mappings</b>	Map custom	custscript_coupa_vi_mr_cus	Free-Form Text			



Script Definition					Script Deployment	
Label	Explanation	Id	Type	Comments	Example	Your Value
		tom_body_fiel d				
<b>Enable Chart of Accounts Sync</b>		custscript_cou pa_vi_mr_syn c_coa_values	Check-box		T or F	
<b>Skip Syncing Remit-To Addresses to Coupa</b>	Skipping the remitto mapping	custscript_cou pa_vi_mr_skip _rta_sync	CheckBox		T or F	
<b>Coupa Supplier ID Field's ID</b>		custscript_cou pa_vi_mr_ven dor_id_field	Free-Form Text			
<b>NetSuite Subsidiary to Coupa Custom Field</b>		custscript_cou pa_vi_mr_sub s_to_custfield	Free-Form Text			



## Review Parameter List User Events for Vendor 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 Definition					Script Deployment	
Label	Explanation	Id	Type	Comments	Example	Your Value
Include in Coupa Field ID	internID of the checkbox field on Vendor Record	custscript_coupa_vi_ue_integration_flag	FreeForm-text		internID of the checkbox field on Vendor Record	
Coupa Instance URL	Coupa host	custscript_coupa_vi_ue_host_url	Hyperlink			
Account Name		custscript_coupa_vi_ue_account_name	FreeForm-text			
Error Email Receiver	Receiver for Errors	custscript_coupa_vi_ue_error_email_to	Text			
Error Email Sender	Sender for Error emails	custscript_coupa_vi_ue_error_email_from	List/record			

## 2 Find and Define Deployment Parameter Values

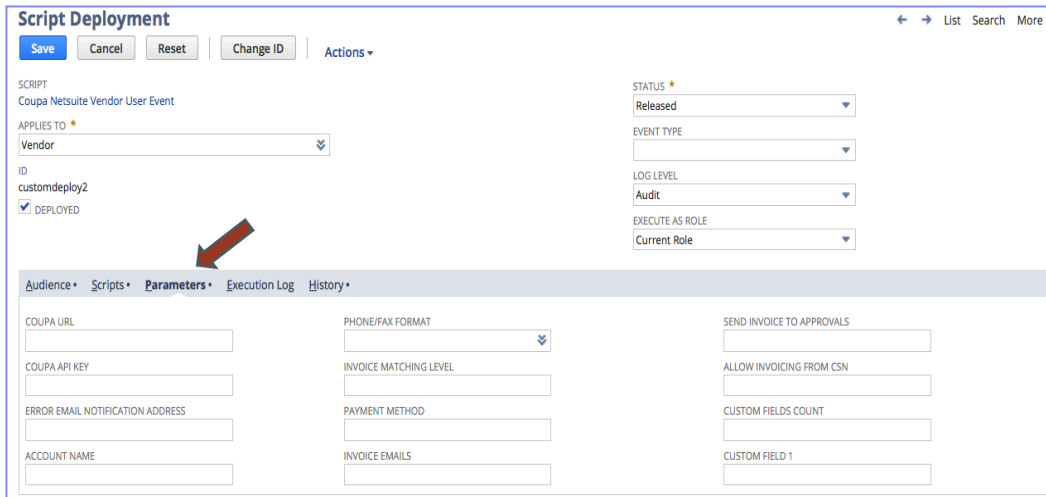
**Navigation Path:** Customization | Scripting | Scripts

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



The screenshot shows the 'Script Deployment' configuration interface. At the top, there are buttons for 'Save', 'Cancel', 'Reset', and 'Change ID', along with an 'Actions' dropdown. The script name is 'Coupa Netsuite Vendor User Event'. The 'APPLIES TO' dropdown is set to 'Vendor'. The ID is 'customdeploy2' and the 'DEPLOYED' checkbox is checked. On the right, there are dropdown menus for 'STATUS' (set to 'Released'), 'EVENT TYPE', 'LOG LEVEL' (set to 'Audit'), and 'EXECUTE AS ROLE' (set to 'Current Role'). A navigation bar at the bottom includes 'Audience', 'Scripts', 'Parameters' (highlighted with a red arrow), 'Execution Log', and 'History'. The main area contains a grid of input fields for various parameters: 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, and CUSTOM FIELD 1.

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

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	Explanation	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.coupa host.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	_vendor_use_display_name	Free-Form Text	<Company Name>	custentity_use_company_name	



Script Definition					Script Deployment	
Label	Explanation	Id	Type	Comments	Example	Your Value
	Coupa. Coupa sets this to the "Display Name" of new Supplier.					
<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.  2-way or 3-way or :NetSuite_field_id	2-way	
<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.	invoice	

Script Definition					Script Deployment	
Label	Explanation	Id	Type	Comments	Example	Your Value
				Value can be invoice or coupa_pay		
<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	
<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:	

Script Definition					Script Deployment	
Label	Explanation	Id	Type	Comments	Example	Your Value
<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	
<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	
<b>Content Group Mapping</b>	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	
<b>Vendor Subsidiary mapping</b>	Map all subsidiaries to ContentGroup or a custom field	custscript_vendor_substo_cust_field	Free-Form Text	Added 7.0.0		



Script Definition					Script Deployment	
Label	Explanation	Id	Type	Comments	Example	Your Value
*Skip Phone Number	Skipping the phone number on integration	_vendor_skipphonenum	CheckBox		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	Free Form		245	
*Subsidiaries	Sync subsidiaries to Coupa	_vendor_subs_to_cust_field	Free Form			
*Content Groups	Sync Content Groups	_vendor_contentgroup_field	Free Form			

## 2 Find and Define Deployment Parameter Values

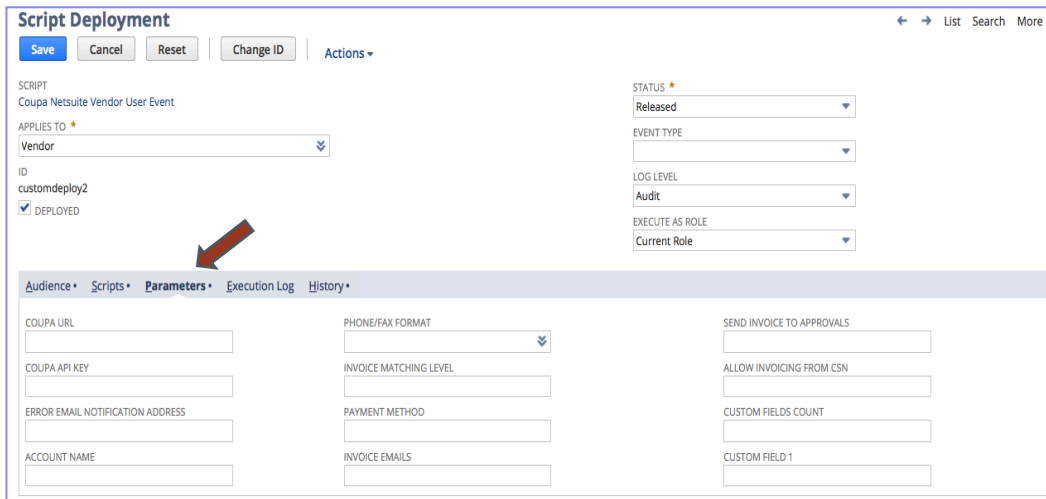
**Navigation Path:** Customization | Scripting | Scripts

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

PHONE/FAX FORMAT

SEND INVOICE TO APPROVALS

COUPA API KEY

INVOICE MATCHING LEVEL

ALLOW INVOICING FROM CSN

ERROR EMAIL NOTIFICATION ADDRESS

PAYMENT METHOD

CUSTOM FIELDS COUNT

ACCOUNT NAME

INVOICE EMAILS

CUSTOM FIELD 1

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

# Netsuite 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	FreeForm Text		https://abc.coupahost.com	
Coupa API Key	The API key for Coupa's REST based API	_coupa_fxrates_apikey	FreeForm 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	FreeForm Text		<a href="mailto:abc@abc.com">abc@abc.com</a> , <a href="mailto:abcd@abcd.com">abcd@abcd.com</a>	
Account Name*	As per standard, it will be company name with which environment its been setup (Test or Prod)	_coupa_fxrates_accountname	FreeForm 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	FreeForm Text			Added 2.0.0

Error Email to	Error Email Recipient	custscript_coupa_fxrate_mr_error_t o	FreeForm Text			Added 2.0.0
----------------	-----------------------	---	---------------	--	--	-------------

## 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

The screenshot shows the Netsuite interface for editing a script. The script is titled "CoupaFXRates Integration" and has the ID "customdeploycustomscriptfxrates". It is currently set to "DEPLOYED" status. The "Parameters" tab is selected, showing fields for "COUPA URL", "COUPA API KEY", "ERROR EMAIL NOTIFICATION ADDRESS", "ACCOUNT NAME", and "UTC OFFSET". A red arrow points to the "Parameters" tab in the navigation bar.

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 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

**1****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

[Create](#)
[Load from file](#)
[Export to](#)
View: All
[Advanced](#)

Showing results for 4000 sales

Lookup	Name	Description	External Ref Num	Default	Active	External Ref Code	Parent Name	Actions
acct_Square, Inc.	4000 Sales	None	4000	No	Yes	038493 4000	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](#)

Showing results for dept\_americas

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:

## Lookups

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

\*Currency

---

#### Account Segments and Defaults

I'll load all of the valid combinations  
 I'll define the values segment by segment

Segment	Purchase request defaults	Expense report defaults
*Segment 1 <input type="text" value="subs"/>	default by <input type="text" value="None"/>	default by <input type="text" value="None"/>
	<input checked="" type="radio"/> This segment is a simple list defined in <input type="text" value="subs_US"/>	
Segment 2 <input type="text" value="dept"/>	default by <input type="text" value="None"/>	default by <input type="text" value="None"/>
	<input checked="" type="radio"/> This segment is a simple list defined in <input type="text" value="dept_US"/>	
	<input type="radio"/> This segment depends on <input type="text"/>	
	<input type="checkbox"/> This segment is required	
Segment 3 <input type="text" value="class"/>	default by <input type="text" value="None"/>	default by <input type="text" value="None"/>
	<input checked="" type="radio"/> This segment is a simple list defined in <input type="text" value="clas_US"/>	
	<input type="radio"/> This segment depends on <input type="text"/>	
	<input type="checkbox"/> This segment is required	
Segment 4 <input type="text" value="acct"/>	default by <input type="text" value="None"/>	default by <input type="text" value="None"/>
	<input checked="" type="radio"/> This segment is a simple list defined in <input type="text" value="acct_US"/>	
	<input type="radio"/> This segment depends on <input type="text"/>	
	<input type="checkbox"/> This segment is required	



## 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	FreeForm Text		https://abc.coupahost.com	
<b>Coupa API Key</b>	The API key for Coupa's REST based API	_coupa_accs_api_key	FreeForm 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	FreeForm 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	FreeForm 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_includ	FreeForm 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	
<b>*Use External ID</b>	Allow using external ID from NS	_coupa_accs_extid	Check Box		T/F	

Script Definition					Script Deployment	
Label	Description	Id	Type	Comments	Example	Your Value
*Custom Segment Name	Applicable for custom segments	_coupa_accs_namefield	FreeForm		Name	
*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			

### 3 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 Purchase Order to Netsuite Purchase Order

## Overview

Customers running P2O process in Coupa need to move Purchase Orders from Coupa to Netsuite, so they can execute the Invoice matching (2-way or 3way) in Netsuite

- Purchase Order created, changed, soft-closed, closed, cancelled can be integrated
- Purchase Order approval workflow in Netsuite should be configured as best practice if soft-closed is integrated. In case of Purchase Order soft-closed in Coupa integration will put Purchase Order in Netsuite on “pending approval” to avoid changes.

## Script Sequence Explained

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

- The Coupa API call to fetch Purchase Order data is called on a predefined schedule.
- The HTTP GET function is used to retrieve the Coupa Purchase Order in relevant status

## Activities

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

## Tips to help you succeed

- Purchase Order process is defined
- Purchase Order status in scope is defined
- Subsidiary is defined

## Pre-Requisites

- Coupa
- Coupa API key created.
- Coupa Chart of Account setup
- Coupa Commodity – Netsuite Item setup
- Netsuite approval workflow setup



### 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	URL of coupa instance	custscript_coupapourl	freeform text			
Coupa API Key	API key setup on Coupa for integration	custscript_coupaapikey	Freeform text		102983oiIJ092831237772310	
Error Email Sender		custscript_coupa_error_email_sender	List/record			
Account Segment	Account	custscript_couapoglactseg	Freeform text			
Class Segment	Class	custscript_coupapoclassseg	Freeform text			
Subsidiary segment	Subsidiary	custscript_coupaposubseg	Freeform text			
Location Segment	Location	custscript_coupapolocseg	Freeform text			
Department Segment	Department	custscript_coupapodeptseg				
Account Name		custscript_coupaacctname	Freeform text			

Body Custom Fields	Header Custom field	custscript_coupapobody	Freeform text			
Line Custom Fields	Line custom fields	custscript_coupapoline	Freeform text			
Email Address Notification	Email address for notification	custscript_coupaemailaddr_notifications	Freeform text			
Use Coupa UOM (units of measure)	UoM	custscript_enablecoupaom	Checkbox			
Using Dynamic Accounting?	Use dynamic accounting segments?	custscript_dynamicaccnts	Checkbox			
UOM Enabled in NS?	Uom In NS	custscript_coupa_po_ns_uom	Checkbox			
Cancelled/Softclose/Closed Coupa POs?	Include cancelled, soft-closed, closed Pos?	custscript_po_support_close	Checkbox			
PO Default Status	Set Po to Approved by default in NS	custscript_coupa_po_default_status	Checkbox			
API Filter	Can be used to filter Pos pulled from COupa	custscript_coupa_po_filter	Text area	Pull POs just for 2 subsidiaries "accounttype[id][in]=9,8"		

## 2 Find and Define Deployment Parameter Values

Navigation Path: Customization | Scripting | Scripts  
Find the "Coupa PO Integration scheduled" 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 Purchase Order”. 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 Receipt to Netsuite Item Receipt

## Overview

Customers running P2O process in Coupa need to move Purchase Orders Receipt from Coupa to Netsuite, so they can execute the Invoice matching (3way) in Netsuite

- Item receipt created, voided can be integrated. Voided receipt in Coupa will result Item receipt deletion in Netsuite

## Script Sequence Explained

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

- The Coupa API call to fetch Item Receipts data is called on a predefined schedule.
- The HTTP GET function is used to retrieve the Coupa Item Receipt in relevant status

## Activities

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

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

## Tips to help you succeed

- Purchase Order process is defined
- Subsidiary is defined

## Pre-Requisites

- Coupa
- Coupa API key created.
- Coupa Chart of Account setup
- Coupa Commodity – Netsuite Item setup

## 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	URL of coupa instance	custscript_coupapourl	freeform text			
Coupa API Key	API key setup on Coupa for integration	custscript_coupaapikey	Freeform text		102983oiIJ092831237772310	
Error Email Sender	Sender	custscript_coupa_error_email_sender	List/record			
Email Addresses for Notifications		custscript_coupaemailaddr_notification	Freeform text			
Body Custom Fields	Header Custom field	custscript_coupapobody	Freeform text			
API Filter	Can be used to filter Pos pulled from Coupa	custscript_coupa_ir_filter	Text area	Pull POs just for 2 subsidiaries “orderline[accounttype][id][in]=9,8”		

## 2 Find and Define Deployment Parameter Values



Navigation Path: Customization | Scripting | Scripts  
Find the “Coupa Item Receipt Integration scheduled” 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 Purchase Order”. 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

### 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 \*

## 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



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://netsuiteintegrations.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_locon	custscript_coupa_chrg_locn	Freeform-text		segment-5	
Charge Custom Fields	Custom field mapping for extra fields	custscript_coupa_chrg_linecf	Freeform-text		coupa-xmlpath==netsuite_field_id	
Charge Custom Fields	Custom field mapping for extra fields	custscript_coupa_chrg_headcf	Freeform-text		coupa-xmlpath==netsuite_field_id	



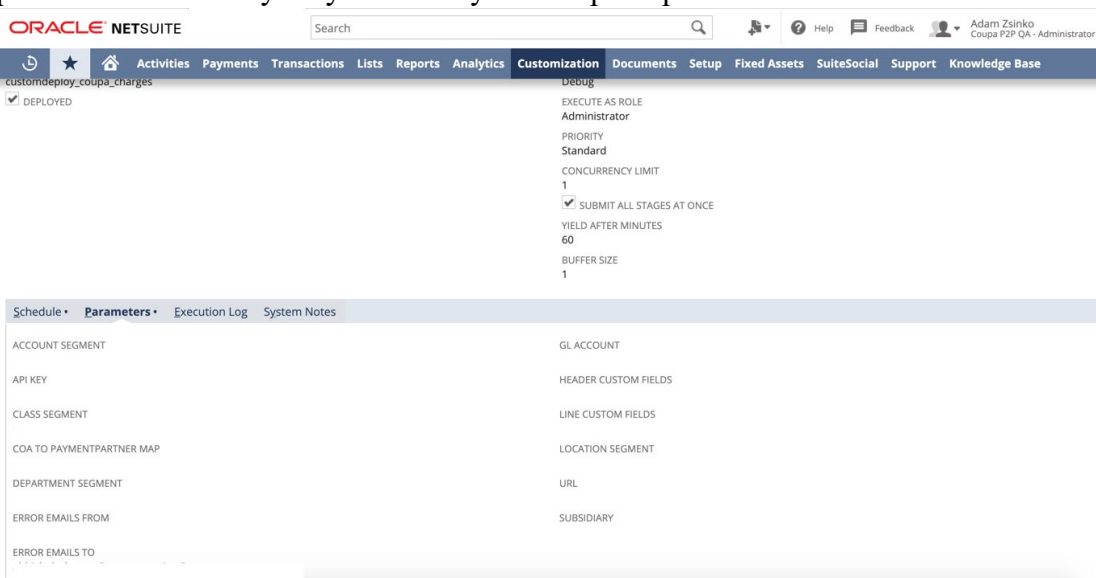
Force Save?	If true, will pass a parameter	customscript_coupa_chrg_forcesave	CheckBox			
	when saving to skip validations					
CoA ID + Payment Partner to Vendor ID	This is a mapping from the Coupa Chart of Accounts ID to the NetSuite Vendor ID representing the Payment Partner or Credit Card Vendor that 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	Textarea		1==3412;2==4812;3==452 With Payment Partner 1==3412; Example” COA_ID==Payment_Partner_ID==NS_VENDOR_ID	
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	
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			
--------------------------	-------------------------------	---------------------------------	---------------	--	--	--

## 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 for a script named 'customdeploy\_coupa\_charges'. The 'Parameters' tab is active, displaying a list of parameters for the script. The parameters are as follows:

Parameter Name	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.