NetSuite Integration

Reference Guide

Document Details

Document #: 160_RG001_240712

System Requirements

- CES Release Version 10.30.00 or higher
- NetSuite Release Version 2021.2 or higher

Overview

This document describes the integration platform that allows customers to move data seamlessly between Crunchtime and NetSuite.

Download the latest version of this document here:
https://links.crunchtime.com/RG-NetSuiteIntegrationReference

… or scan the QR code to the left.

Related Documentation

- NetSuite Integration Setup Guide: https://links.crunchtime.com/SG-NetSuiteIntegrationSetup

© 2024 Crunchtime - All Rights Reserved

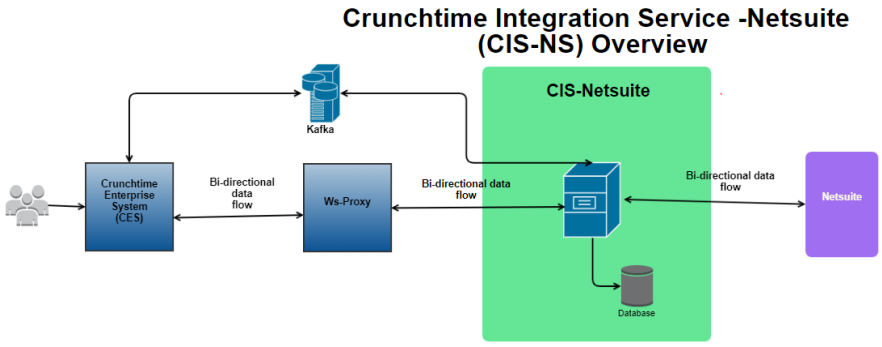
REFERENCE GUIDE

NetSuite Integration

Overview

Crunchtime's NetSuite integration platform allows customers to move data seamlessly between Crunchtime and NetSuite. This integration will push accounts payable (invoices/vendors), inventory changes and sales summary data into the NetSuite application. Behind the scenes, the integration uses Simple Object Access Protocol (SOAP) API's. The service which enables this seamless, bi-directional data transfer is known as Crunchtime Integration Services (CIS).

- CIS is a microservice running on a Docker container.
- Deployment of CIS is independent from the Crunchtime application.
- The Crunchtime application runs on Java11, Spring framework, using an Oracle 19c database, running on a Tomcat server.



Features

- Customers can configure the integration record and all relevant values.
- Customers can define the recipient email address for each integration.
- The application is pre-configured with the exports that can be scheduled (see the list below).
- After the data is transferred, customers will be able to review integration errors and reprocess the export if necessary.

Use Cases & Data Flow

- *Accounts Payable (Reconciled Invoices)* – Vendor Orders, Vendor Returns, and Purchase By Invoice (PBI) transactions - GL level. Unidirectional, from Crunchtime to NetSuite.
- *Sales (Summary level)* – Journal Entry to capture the reconciliation of Sales by Location/GL Account (Net Sales to Cash including Over/Shorts). Unidirectional, from Crunchtime to NetSuite.
- *Location Transfers (Inventory Changes)* – Journal Entry to capture transfers from one physical Location to another (weekly based on post period). Unidirectional, from Crunchtime to NetSuite.
- *Period Ending Inventory Values* – Inventory Delta. Unidirectional, from Crunchtime to NetSuite.
- *Commissary Orders* – Journal Entry to capture these transactions - GL level. Unidirectional, from Crunchtime to NetSuite.
- *PBI Sync* – Purchase Orders captured directly in NetSuite. Unidirectional, from NetSuite to Crunchtime.
- *Vendor Sync* – Bidirectional, from NetSuite to Crunchtime or vice versa. Depending on how one of the User Defined Attributes (UDAs) is configured in Crunchtime, this will flow in one direction or the other.

(Continued)

© 2024 Crunchtime - All Rights Reserved

NetSuite Integration

(Continued)

Customer Compatibility

- Currently these integrations are only available to Crunchtime cloud-hosted customers; they are not available for cruise or self-hosted customers.

Prerequisites

The following information must be configured in the NetSuite and Crunchtime systems before you begin integration setup:

- Accounting periods must be created in NetSuite prior to integration.
- At least one subsidiary must exist in NetSuite (for customers using NetSuite Oneworld).
- The same locations must exist in both Crunchtime and NetSuite.
- Vendors must be created and synced between the 2 systems. If you choose to run the Vendor Sync process, this must be done prior to other transactions.
  - Vendors should be maintained in either in NetSuite or Crunchtime (a single system of record). For example: If NetSuite is the system of record, the Vendor Sync process will sync from NetSuite to Crunchtime.



Tip: This is only recommended for customers who are new to either Crunchtime or NetSuite.

- The Crunchtime "Vendor Code" or "Export Code" must match the NetSuite "Vendor ID" for all Vendors.
- GL Codes must match between systems, or be configured to accomodate Classes.
- GL Codes in NetSuite used for vendor bills must be classified as 'Expenses'. This enables the transfer of vendor invoice information between the two systems.
- In Crunchtime, the relevant User Group permissions must be enabled for the user configuring the User Defined Attributes (UDAs).
  - Specific permission is required for defining certain "Secret" UDA values, such as 'NS_Tenant_URL', 'NS_Consumer_Key' etc, which are required for authentication and API connectivity. (*Please refer to attributes listed in the table on page 5.*)

(Continued)

© 2024 Crunchtime - All Rights Reserved

NetSuite Integration

(Continued)

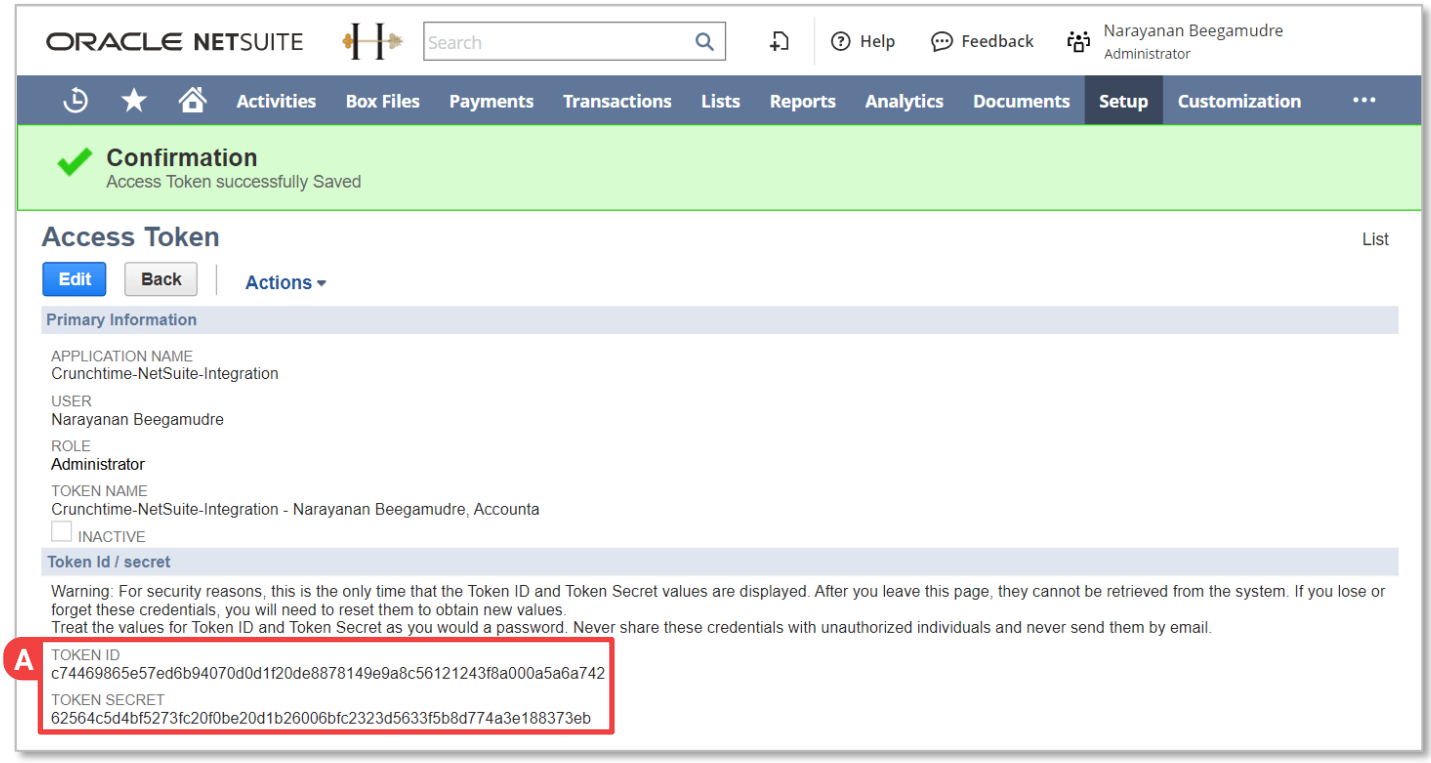
Integration Setup – Process Overview

1. Add the Crunchtime SuiteApp to your NetSuite system.
2. In NetSuite, create the Integration User and generate Access Tokens; see example below.
3. In Crunchtime, configure the company-level User Defined Attributes (UDAs); see the *User Defined Attributes* section on the next page for more details.
4. If necessary, define related vendor-specific UDAs.
5. If necessary, define related location-specific UDAs.

API Setup – Access Tokens

After installing the SuiteApp, you need to generate the *Token ID* and *Token Secret* values.

Oracle NetSuite: Setup > Users & Roles > Access Tokens > New



A This is the only time these tokens will be visible, so you **MUST** take note of them now.

Note: For detailed instructions to complete the steps above see the NetSuite Integration Setup Guide, found here: https://links.crunchtime.com/SG-NetSuiteIntegrationSetup

© 2024 Crunchtime - All Rights Reserved

NetSuite Integration

(Continued)

User Defined Attributes

- A User Defined Attribute (UDA) is a data element built in Crunchtime to support your data configuration. These UDA values are used when we do not have relevant mapping for the Company, Location, and/or Vendor fields in Crunchtime.
- The UDA values that start with "NS_" are those pertaining to NetSuite UDA configuration.
- You will need to configure the relevant UDAs with your specific data to facilitate the integration.
- Users need the appropriate permissions to modify UDA data in Crunchtime.

The following Company-level UDAs are required for NetSuite integration (listed alphabetically).

	Field	Description	Required for Integration
1	NS_ACCOUNT	Account number for NetSuite.	Yes
2	NS_COMPANY_SUPPLIER_PAYMENT_TERM	NetSuite Vendor field for "Payment Terms." Determines how long before payment is due. This is the global level for all Vendors. If NS_SUPPLIER_PAYMENT_TERM is configured for a Vendor, it will override this global value for that vendor. Use 'Net 30' as the default.	Yes
3	NS_CONSUMER_KEY	Consumer account key.	Yes
4	NS_CONSUMER_SECRET	Consumer secret key. Provided to the customer by Crunchtime.	Yes
5	NS_CREDIT_LOCATION_LINE	For Vendor Credits, Locations will be added either at the line or header level. Acceptable values are True or False. Default value is False. If False, the location will be used at both header and line level. If True, the location will only be at the line level.	Yes
6	NS_INTEGRATION_ENABLED	Allowed values are 'True' or 'False'. Set this value to 'True' to enable NetSuite integration.	Yes
7	NS_IS_SUPPLIER_SOR	Allowed values are 'True' or 'False'. Determines which system should be read as the system of record (SOR). 'True' indicates NetSuite is SOR, while 'False' indicates Crunchtime is SOR.	Yes
8	NS_SUBSIDIARY	Default subsidiary ID. This is the Internal ID from NetSuite.	Yes
9	NS_SUBSIDIARY_NAME	The subsidiary this company is attached to.	Yes
10	NS_TENANT_URL	The URL where we will send the data.	Yes
11	NS_TOKEN_ID	Integration account token ID. To be generated by the customer in NetSuite.	Yes
12	NS_TOKEN_SECRET	Secret token. To be generated by the customer in NetSuite.	Yes
13	NS_VENDOR_BILL_APPROVAL_STATUS	Determines if a Vendor Bill comes in as auto approved or waiting for manual approval. Can be "Pending Approval" or "Approved".	Yes

(Continued)

© 2024 Crunchtime - All Rights Reserved

NetSuite Integration

(Continued)

The following Company-level UDAs are available for NetSuite integration (listed alphabetically).

	Field	Description	Required for Integration
14	NS_AP_HEADER_MEMO	Sets a default value in the memo field of the AP header.	No
15	NC_CLASS_ID	Enter a NetSuite Class Internal ID associated with the Crunchtime GL Number.	No
16	NS_CLASS_LINE	This can be used to indicate which transaction(s) should us the GL Description as the Class. The associated Class description will be applied to all listed transactions when data is exported. Values are set with pipe delimiter based on transaction type – can include NS_AP|NS_ID|NS_LT|NS_S.	
17	NS_CO_CUSTOM_FIELD	Sends Journal field information for Commissary Order transactions. Accepted values are: Memo.	No
18	NS_IC_JOURNAL	Location Transfers will either be sent as an Advanced Intercompany Journal or Journal Entry. Accepts values are True/False/Both. Default value is False. If False, Transfers will be sent as Journal Entry. If True, Transfers will be sent as an Advanced Intercompany Journal. If Both, the system will determine the Journal Type depending on Subsidiary data of the location.	No
19	NS_ID_CUSTOM_FIELD	Sends Journal field information for Inventory Delta transactions. Accepted values are: Memo.	No
20	NS_INVENTORY_DELTA_FAILURE_EMAIL	After inventory deltas fail to send, notify email address(es). Separate multiple entries by comma. If you do not wish to be notified for failed push, leave the field blank.	No
21	NS_INVENTORY_DELTA_SUCCESS_EMAIL	After inventory deltas are sent successfully, notify email address(es). Separate multiple entries by comma. If you do not wish to be notified for successful push, leave field blank.	No
22	NS_INVOICE_FAILURE_EMAIL	If invoices fail to send, notify email address(es). Separate multiple entries by comma. If you do not wish to be notified for failed push, leave the field blank.	No
23	NS_INVOICE_LOCATION_LINE	For Vendor Bils, Locations will be added either at the line or header level. Acceptable values are True or False. Default value is False. If False, the location will be used at both header and line level. If True, the location will only be at the line level.	No
24	NS_INVOICE_SUCCESS_EMAIL	After invoices are sent successfully, notify email address(es). Separate multiple entries by comma. If you do not wish to be notified for successful push, leave the field blank.	No

(Continued)

© 2024 Crunchtime - All Rights Reserved

NetSuite Integration

(Continued)

	Field	Description	Required for Integration
25	NS_JOURNAL_CUSTOM_FORM	Determines which custom form is chosen when creating a Journal.	No
26	NS_LOCATION_ID	The internal ID associated with a location within NetSuite. Fill out to confirm that a location will be mapped to NetSuite using the "internal ID" instead of the "location name."	
27	NS_LOCATION_SUBSIDIARY	If entered, will use this value to override NS_SUBSIDIARY for the specified location.	No
28	NS_LOCATION_TRANSFER_FAILURE_EMAIL	If location transfers fail to send, notify email address(es). Separate multiple entries by comma. If you do not wish to be notified for failed push, leave the field blank.	No
29	NS_LOCATION_TRANSFER_SUCCESS_EMAIL	If location transfers are sent successfully, notify email address(es). Separate multiple entries by comma. If you do not wish to be notified for successful push, leave the field blank.	No
30	NS_LOC_SALES_GL_MAP	In a sales transaction, Override Sales account to remap to new Location Specific GL Account. Can be used to specify a Deposit account.	No
31	NS_LT_CUSTOM_FIELD	Sends Journal field information for location transfer transactions. Accepted values are: Memo.	No
32	NS_PULL_VENDOR_INVOICE_ATTACHMENTS	Send invoice attachments from Crunchtime to NetSuite. Can be either 'True' or 'False'.	No
33	NS_REMAP_SALES_GL_ACCOUNT	In a sales transaction file, use this account to override specified GL. When the system encounters a value in this UDA, the specified GL would be remapped to the one defined in the location UDA named NS_LOC_SALES_GL_MAP.	No
34	NS_SALES_CUSTOM_FIELD	Sends Journal field information for sales transactions. Accepted values are: Memo.	No
35	NS_SALES_FAILURE_EMAIL	If sales fail to send, notify email address(es). Separate multiple entries by comma. If you do not wish to be notified for failed push, leave the field blank.	No
36	NS_SALES_SUCCESS_EMAIL	If sales are sent successfully, notify email address(es). Separate multiple entries by comma. If you do not wish to be notified for successful push, leave the field blank.	No

(Continued)

© 2024 Crunchtime - All Rights Reserved

NetSuite Integration

(Continued)

	Field	Description	Required for Integration
37	NS_STATISTICAL_ACCOUNT_EXCLUDED	Excludes “Statistical” accounts from Sales transactions. If ‘True’, then these accounts will be excluded from sales transactions. *If set to ‘True’, then work with Crunchtime Support to determine potential valid values.*	No
38	NS_SUPPLIER_PAYMENT_TERM	NetSuite Vendor field for "Payment Terms." Determines how long before payment is due. For the specified vendor, will override the entry from NS_COMPANY_SUPPLIER_PAYMENT_TERM.	No
39	NS_VENDOR_BILL_CUSTOM_FORM	Determines which custom form is chosen when creating a Vendor Bill.	No
40	NS_VENDOR_CUSTOM_FORM	Determines which custom form is chosen when creating a Vendor.	No
41	NS_VENDOR_FILTER_EXCLUDE	Vendor Sync will exclude Categories listed (based on Category Internal ID). Groups will be separated by pipe character (|). Only ONE of these UDAs (either Include or Exclude) can be used.	No
42	NS_VENDOR_FILTER_INCLUDE	Vendor Sync will include Categories listed (based on Category Internal ID). Groups will be separated by pipe character (|).Only ONE of these UDAs (either Include or Exclude) can be used.	No
43	NS_VENDOR_SYNC_CONFIG	This attribute is applicable only when NetSuite is the system of record. If so, set to 'NC_VENDOR_CODE|NS_INTERNAL_ID'. Vendor Sync will then set NC_VENDOR_CODE to NS_INTERNAL_ID and NC_VENDOR_EXPORT_CODE to NS_VENDOR_CODE.	No
44	NS_VENDOR_SYNC_EMAILS	After vendors are synced with NetSuite in either direction, notify email address(es). Email will be sent irrespective of success or failure. Separate multiple emails by comma. If you do not wish to be notified of a successful push, leave the field blank.	No

© 2024 Crunchtime - All Rights Reserved

NetSuite Integration

(Continued)

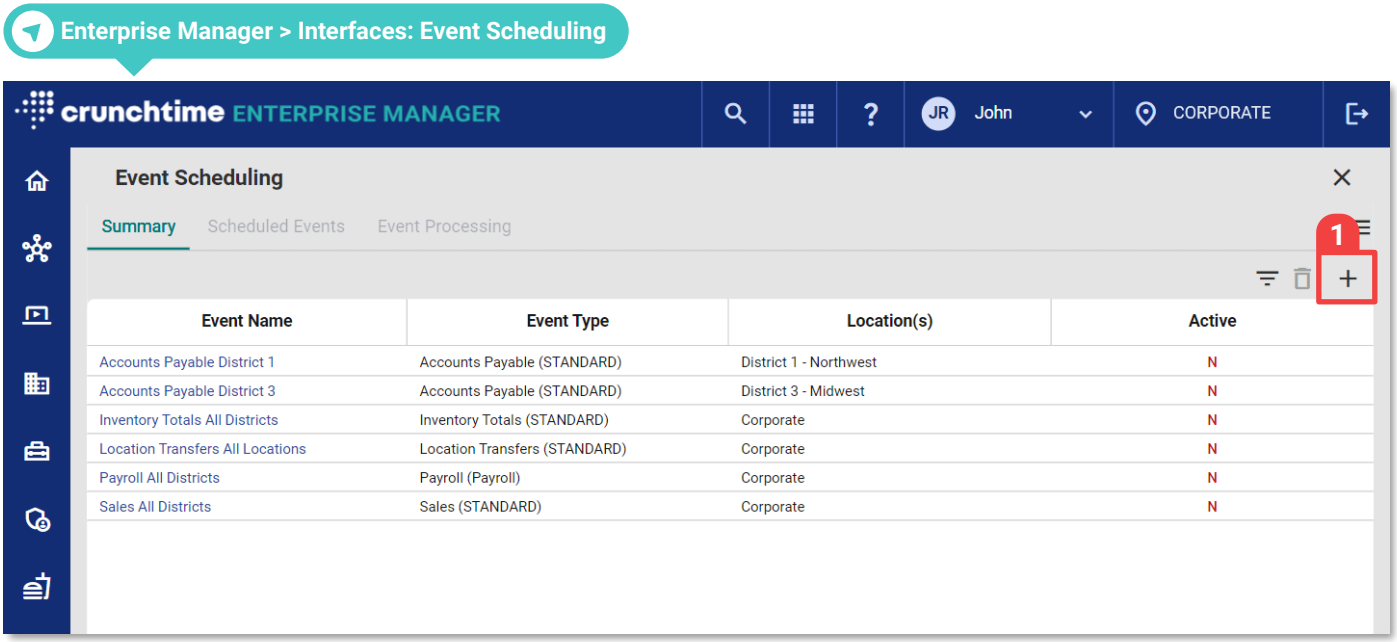
Launching and Scheduling Integrations

Prerequisites

- These integrations can only be scheduled and run if the “NS_INTEGRATION ENABLED” UDA = ‘True’.
- The relevant User Defined Attributes must be defined prior to scheduling exports.
- CIS will pull this UDA data from Crunchtime before it pushes the data to NetSuite.

Process Overview

In Crunchtime, depending on their business needs customers can schedule NetSuite integration exports to run on a daily, weekly, or monthly basis in the *Event Scheduling* screen.



1. In the Event Scheduling screen, select the *Plus* (Add) icon to create a new scheduled export.

Note: For detailed instructions on scheduling an export, see the related QuickStart Guide, found here: https://links.crunchtime.com/EMQS-ScheduleAnExport

2. When a scheduled export runs, the integration will pull data from Crunchtime and send it to NetSuite.
3. If there are any errors, they will appear in the Outbound Integrations screen in Crunchtime.
4. You can use the error logs to troubleshoot the issue.
5. Once the issue is resolved, you can reschedule the export.

© 2024 Crunchtime - All Rights Reserved

NetSuite Integration

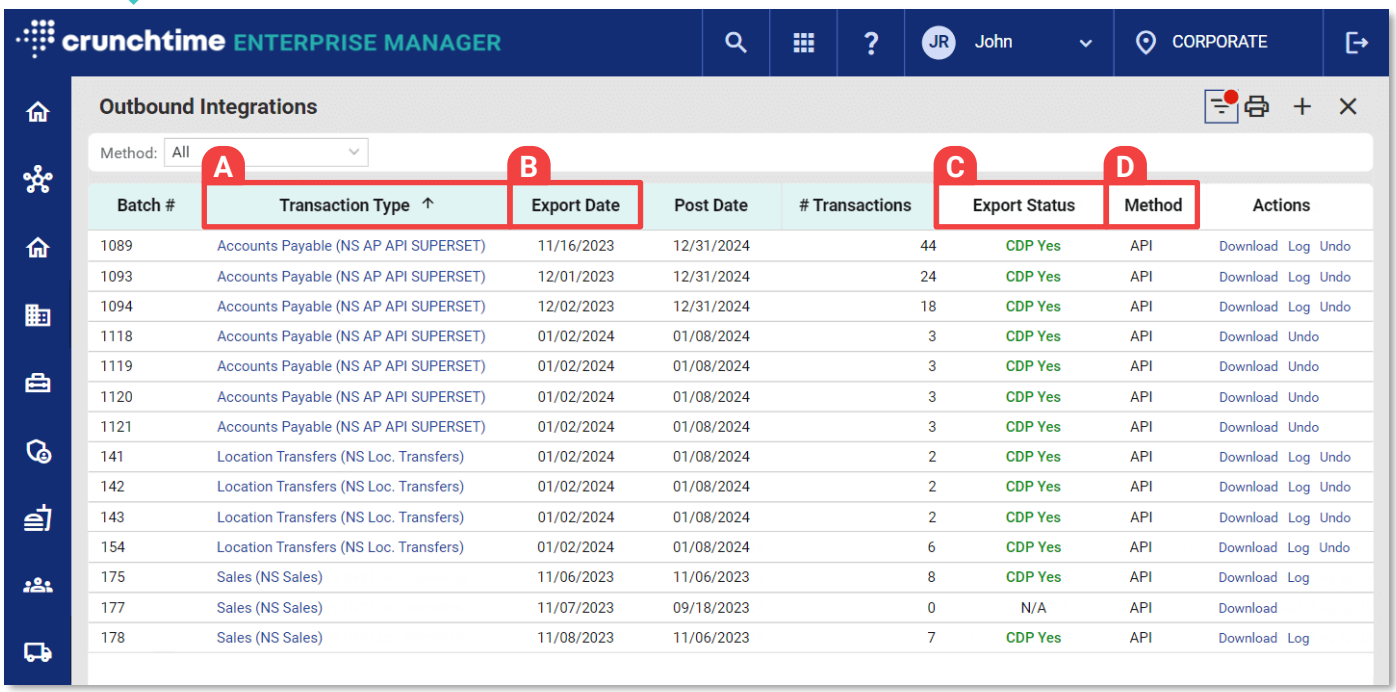
(Continued)

Error Remediation and Reprocessing

In Crunchtime, the *Outbound Integrations* screen is available to help you troubleshoot and resolve the issues, then reprocess the files.

Heads-Up: Currently, Vendor Sync process errors will *not* show up in this screen; you will only be able to track those via email.

Enterprise Manager > Interfaces: Outbound Integrations



Batch #	Transaction Type	Export Date	Post Date	# Transactions	Export Status	Method	Actions
1089	Accounts Payable (NS AP API SUPERSET)	11/16/2023	12/31/2024	44	CDP Yes	API	Download Log Undo
1093	Accounts Payable (NS AP API SUPERSET)	12/01/2023	12/31/2024	24	CDP Yes	API	Download Log Undo
1094	Accounts Payable (NS AP API SUPERSET)	12/02/2023	12/31/2024	18	CDP Yes	API	Download Log Undo
1118	Accounts Payable (NS AP API SUPERSET)	01/02/2024	01/08/2024	3	CDP Yes	API	Download Undo
1119	Accounts Payable (NS AP API SUPERSET)	01/02/2024	01/08/2024	3	CDP Yes	API	Download Undo
1120	Accounts Payable (NS AP API SUPERSET)	01/02/2024	01/08/2024	3	CDP Yes	API	Download Undo
1121	Accounts Payable (NS AP API SUPERSET)	01/02/2024	01/08/2024	3	CDP Yes	API	Download Undo
141	Location Transfers (NS Loc. Transfers)	01/02/2024	01/08/2024	2	CDP Yes	API	Download Log Undo
142	Location Transfers (NS Loc. Transfers)	01/02/2024	01/08/2024	2	CDP Yes	API	Download Log Undo
143	Location Transfers (NS Loc. Transfers)	01/02/2024	01/08/2024	2	CDP Yes	API	Download Log Undo
154	Location Transfers (NS Loc. Transfers)	01/02/2024	01/08/2024	6	CDP Yes	API	Download Log Undo
175	Sales (NS Sales)	11/06/2023	11/06/2023	8	CDP Yes	API	Download Log
177	Sales (NS Sales)	11/07/2023	09/18/2023	0	N/A	API	Download
178	Sales (NS Sales)	11/08/2023	11/06/2023	7	CDP Yes	API	Download Log

Recommended Filter/Sort Options:

- **A** *Transaction Type* – Filter for Accounts Payable, Inventory Delta, Location Transfers, Sales, etc.
- **B** *Export Date* (default)– Filter by this column to show the most recent exports sent.
- **C** *Export Status* – Filter for 'CDP Yes' (Export successful), 'CDP No' (Not yet sent), or 'N/A' (Export failed).
- **D** *Method* – Sort by this column to group the 'API' integrations together.

Actions:

- *Download* – This downloads the original file that was submitted.
- *Log* (Only present for API integrations) – Opens the Error Log screen (shown on next page), in which you can check the status details (success/failure) of the selected transaction.
- *Undo* – This will flag the data as "unexported" so you can resolve the issue and re-export the data.

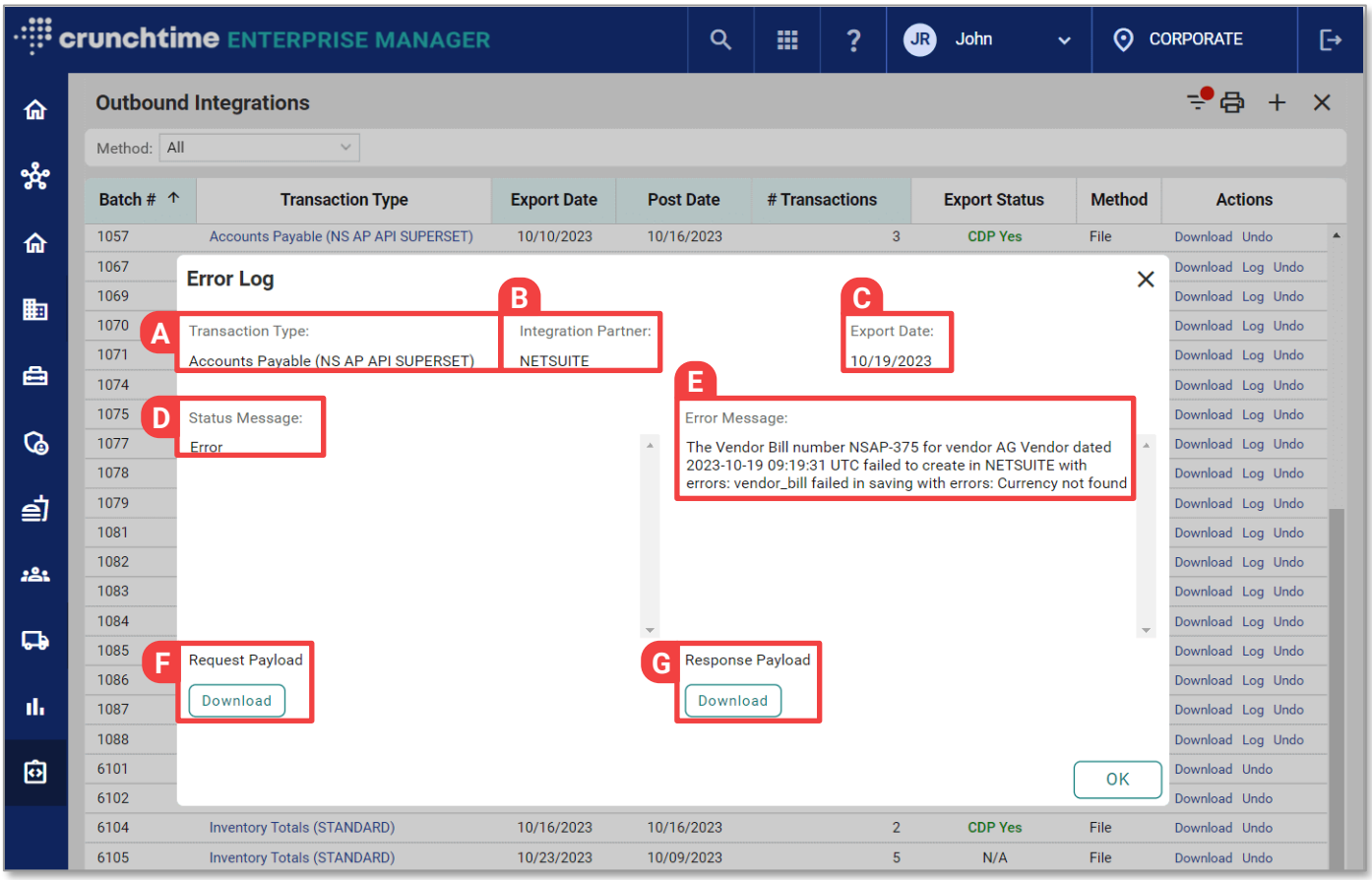
(Continued)

© 2024 Crunchtime - All Rights Reserved

NetSuite Integration

(Continued)

Log File Details



- **A** *Transaction Type* – The type of transaction that was submitted.
- **B** *Integration Partner* – Accounting system to which the transaction was submitted.
- **C** *Export Date* – Date the export was generated.
- **D** *Status Message* – Shows the Success or Error status.
- **E** *Error Message* – Error details for the failed transaction.
- **F** *Request Payload – Download* details of the Submitted transaction.
- **G** *Response Payload – Download* details of the NetSuite response message.

(Continued)

© 2024 Crunchtime - All Rights Reserved

NetSuite Integration

(Continued)

Remediation Steps for Failed Transactions:

1. Validate the status of the transaction based on triggered emails. Emails can be sent for both success and failure of transactions submitted to NetSuite.

Tip: Once your integration is running successfully, you can remove the addresses from the "SUCCESS_EMAILS" attributes so you'll only receive notifications for failed transactions.

2. For the failed transactions, download the log file and review the error details.
3. Validate the data and review the issues mentioned in the email.
4. Undo the export, correct the data and re-export the file. This action creates a new export event.

Resubmit Limitation: Per Step 4 above, vendor invoices can only be resubmitted once. If the new (re-exported) file fails, then repeat the process above for *that* export file – do not attempt to undo and resubmit the original file.

© 2024 Crunchtime - All Rights Reserved

REFERENCE GUIDE

NetSuite Integration

Process Flows & Related Details

AP – Invoices (Vendor Bills) Integration

This integration will push reconciled Vendor Orders, Vendor Returns, and Purchase By Invoice (PBI) transactions from Crunchtime to NetSuite as Accounts Payable invoices. The information is captured at a summary level (by GL account). This will be a unidirectional push from Crunchtime to NetSuite.

Key Related UDAs:

- NS_AP_HEADER_MEMO
- NS_CLASS_LINE (if NS_AP is defined)
- NS_CREDIT_LOCATION_LINE
- NS_INVOICE_FAILURE_EMAIL
- NS_INVOICE_LOCATION_LINE
- NS_INVOICE_SUCCESS_EMAIL
- NS_PULL_VENDOR_INVOICE_ATTACHMENTS
- NS_VENDOR_BILL_APPROVAL_STATUS
- NS_VENDOR_BILL_CUSTOM_FORM

Accounts Payable – Vendor Bills, Vendor Credits, & Purchases By Invoice

- In order for an AP transaction to be picked up in the NS integration, it must meet the following criteria:
  - Must be created with valid supplier. (*Must be an active supplier in both systems and the Crunchtime Vendor Code or Vendor Export Code must match the NetSuite "Vendor ID".*)
  - Must be contain 1 or more line items.
  - Must be reconciled.
- The sign of the Invoice Total will determine if the transaction is sent as a Bill or Credit (regardless of positive or negative line items).
  - When the Invoice Total is positive, the file is sent to NetSuite as a Vendor Bill.
  - When the Invoice Total is negative, the file is sent to NetSuite as a Vendor Credit.

Note: In NetSuite, the Invoice Total for a Vendor Credit is displayed as a positive value.

- Transactions with attachments (like .pdf, .doc, .jpg, or .png files) can be exported if the Company UDA "NS_PULL_VENDOR_INVOICE_ATTACHMENTS" is set to 'True'.
  - To send attachments to NetSuite, the CRUNCHTIME _INTEGRATION folder must be created in NetSuite in the "File Cabinet" section. This folder can only be accessed with permissions by the CRUNCHTIME_INTEGRATION user.
  - File Path: *Documents > File Cabinet > New*
- When a NetSuite Vendor Credit form has the Location required at the line level, the Company UDA for "NS_CREDIT_LOCATION_LINE" must be set to 'True'. Otherwise, this must be set to 'False'.

(Continued)

© 2024 Crunchtime - All Rights Reserved

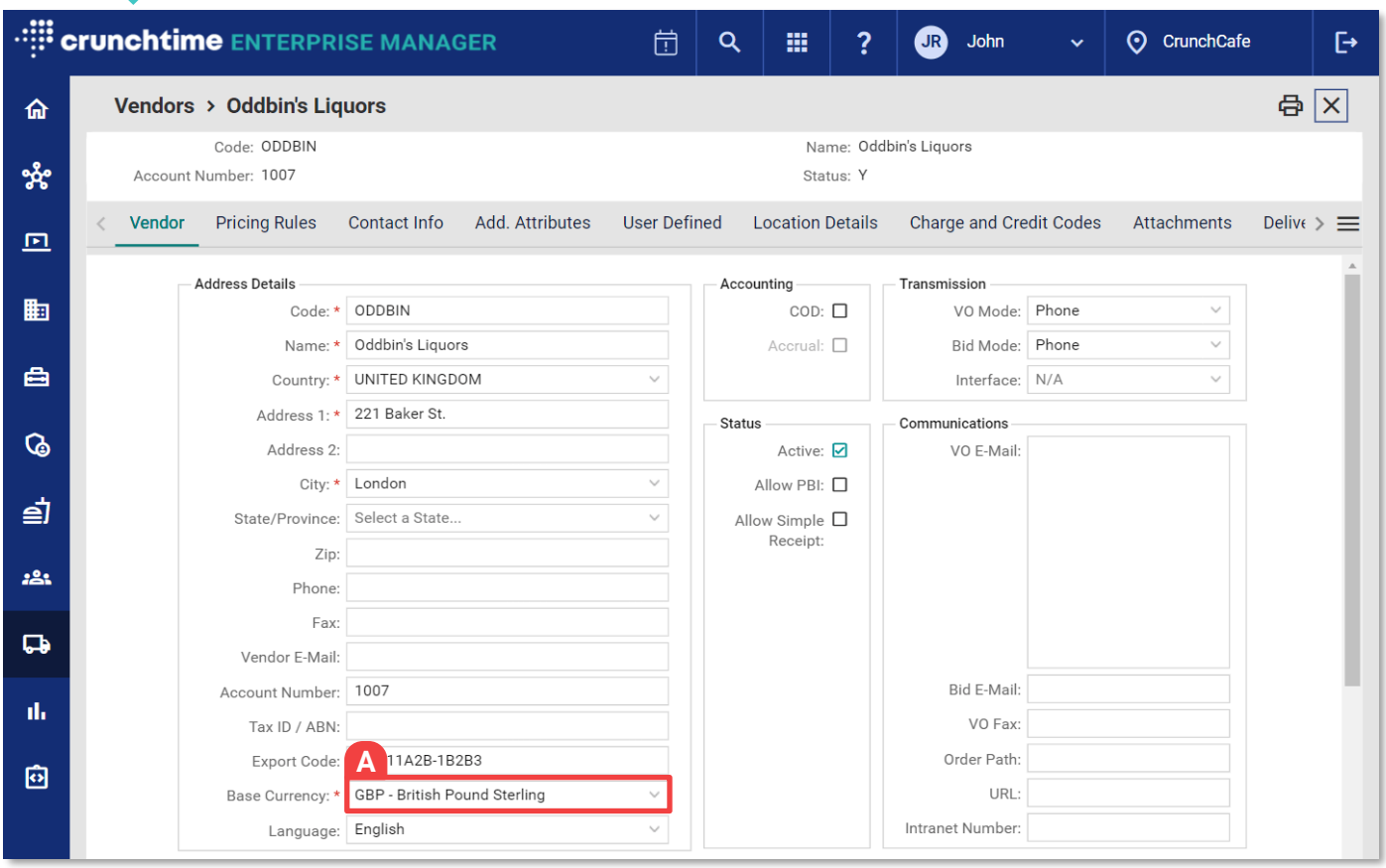
NetSuite Integration

Process Flows & Related Details (Continued)

Crunchtime Vendor Base Currency & NetSuite Vendor Bill Currency

When sending Accounts Payable information to Vendor Bills in NetSuite, the NetSuite currency needs to match the Crunchtime Vendor “Base Currency”.

Enterprise Manager > Purchasing: Vendors > {Edit} > Vendor tab



A When the Accounts Payable export is scheduled to run with Vendor Orders/PBIs, the Vendor “Base Currency” will determine the Vendor “Credit Currency” used in NetSuite.

Note: The AP record will fail to push if the Base Currency does not match a currency in NetSuite.

(Continued)

NetSuite Integration

Process Flows & Related Details (Continued)

Inventory Location Transfers

This integration will push reconciled Location Transfers from Crunchtime to NetSuite as Journal Entries. The information is captured at a summary level (by GL account). This will be a unidirectional push from Crunchtime to NetSuite.

Key Related UDAs:

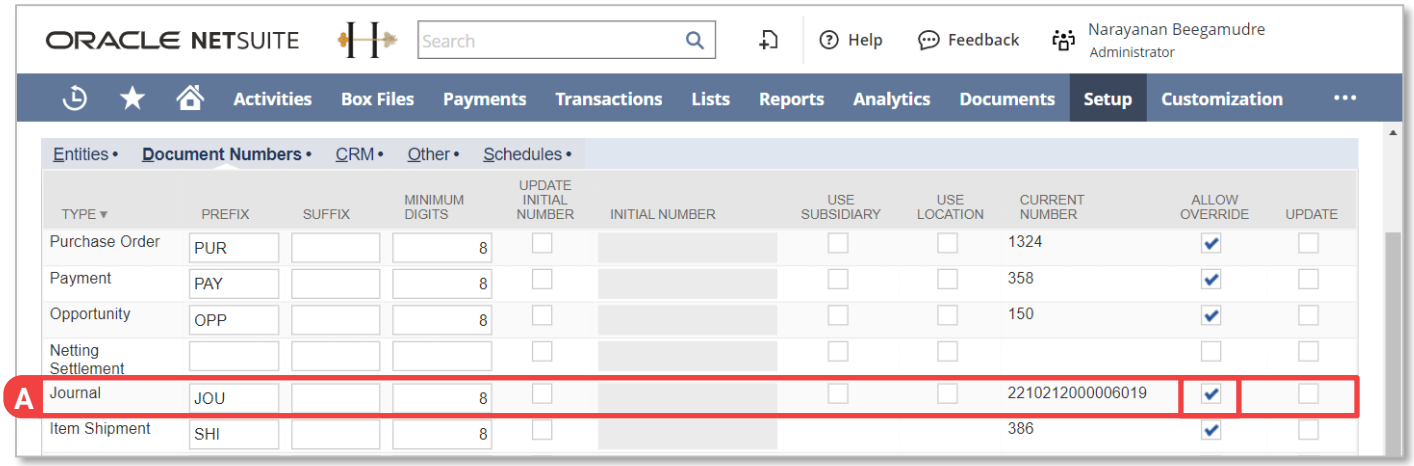
- NS_CLASS_LINE (if NS_LT is defined)
- NS_IC_JOURNAL
- NS_LOCATION_SUBSIDIARY
- NS_LOCATION_TRANSFER_FAILURE_EMAIL
- NS_LOCATION_TRANSFER_SUCCESS_EMAIL

Inter-Company Journals (IC Journals):

- This is for customers using multiple subsidiaries within the same Location Transfer transaction. The standard Journal entry form cannot handle these transactions, so the Inter-Company Journal forms will be used for this.

  Scenario:
  - Location A tied to Subsidiary 1
  - Location B tied to Subsidiary 2
- The Company UDA for "NS_IC_JOURNAL" must be set to 'True'.
- In NetSuite, overrides must be enabled for Auto-Generated Numbers for Journals; see example below. (*This is necessary because the transfer transaction numbers are assgined in Net-Chef*.)

Oracle NetSuite: Setup > Company > Auto-Generated Numbers



A For the "Journal" Type, the *Allow Override* option needs to be checked.

(Continued)

© 2024 Crunchtime - All Rights Reserved

NetSuite Integration

Process Flows & Related Details (Continued)

Inventory Delta – Period Ending Values

This integration will push Period Ending Inventory Values from Crunchtime to NetSuite. This will be a unidirectional push.

- In Crunchtime, the GL Account should include a valid "GL for CGS" entry. (*These are the offset accounts used in these transactions and must match between Crunchtime and NetSuite*.)

Key Related UDAs:

- NS_CLASS_LINE (if NS_ID is defined)
- NS_INVENTORY_DELTA_FAILURE_EMAIL
- NS_INVENTORY_DELTA_SUCCESS_EMAIL
- NS_LOCATION_SUBSIDIARY

Sales Report

This integration will push Sales data from Crunchtime to NetSuite as Journal Entries. The information is captured at a summary level (by GL account). This will be a unidirectional push from Crunchtime to NetSuite.

Key Related UDAs:

- NS_CLASS_LINE (if NS_S is defined)
- NS_LOCATION_SUBSIDIARY
- NS_STATISTICAL_ACCOUNT_EXCLUDED
- NS_SALES_FAILURE_EMAIL
- NS_SALES_SUCCESS_EMAIL

Statistical Accounting:

- Crunchtime does not have the ability to divide GL accounts, Financial OR Statistical. These accounts are configured separately within Crunchtime.
- Any statistical account codes must be filtered out of the Sales export before the Sales transactions are pushed to NetSuite.
  - The accounts to be excluded will be stored in the T_GL_UDA table – customers must work with Crunchtime Support to determine which accounts need to be entered in the table.
  - When the Company UDA for "NS_STATISTICAL_ACCOUNT_EXCLUDED" is set to 'True', these accounts will be excluded from the Sales export before pushing them to NetSuite to create the Journal entries.



Tip: We recommend running Sales weekly, since NetSuite only accepts one sales import per day.

PBI Sync

For invoices created directly in NetSuite, the PBI Sync process will push these Vendor Bills from NetSuite into Crunchtime as PBIs. This will be a unidirectional push from NetSuite to Crunchtime.

(Continued)

© 2024 Crunchtime - All Rights Reserved

NetSuite Integration

Process Flows & Related Details (Continued)

Commissary Orders

This integration will push reconciled Commissary Orders from Crunchtime to NetSuite as Journal Entries. The information is captured at a summary level (by GL account). This will be a unidirectional push from Crunchtime to NetSuite.

- The order status must be “Final Reconciled” for these transactions to be eiligible for export.
- When scheduled and exported, the “NS Com. Order” export file will be transformed and uploaded as a NetSuite Journal Entry.

Key Related UDAs:

- NS_CLASS_LINE
- NS_JOURNAL_CUSTOM_FORM
- NS_LOCATION_SUBSIDIARY
- NS_LOCATION_ID
- NS_SUBSIDIARY

Vendors Sync Integration

This integration will synchronize Vendor records between Crunchtime and NetSuite. This will be a unidirectional push from whichever of them is the system of record to the other system. The UDA setting for NS_IS_SUPPLIER_SOR will determine which is the system of record.

Key Related UDAs:

- NS_IS_SUPPLIER_SOR
- NS_VENDOR_CUSTOM_FORM
- NS_VENDOR_FILTER_EXCLUDE
- NS_VENDOR_FILTER_INCLUDE
- NS_VENDOR_SYNC_CONFIG
- NS_VENDOR_SYNC_EMAILS

Crunchtime to NetSuite:

- When Crunchtime is the SOR for Suppliers, create a Vendor in Crunchtime and push to NetSuite.
  - The Vendor Sync process can be initiated via Event Scheduling.
  - The Outbound Integrations screen in Crunchtime does not display logs for this process. Success or Failure Emails will be triggered once the process has completed for each run.

NetSuite to Crunchtime:

- When NetSuite is the SOR for Suppliers, create a Vendor in NetSuite and push to Crunchtime.
  - NetSuite vendors should have Address fields configured for upload in to Crunchtime.
  - The NetSuite “Vendor ID” should not be more than 6 characters.
  - The Outbound Integrations screen in Crunchtime does not display logs for this process. Success or Failure Emails will be triggered once the process has completed for each run.

(Continued)

© 2024 Crunchtime - All Rights Reserved

NetSuite Integration

Process Flows & Related Details (Continued)

Endpoints/APIs:

NetSuite Endpoints	
End Point	**API Reference**
Journal Entry	urn:general.transactions.webservices.NetSuite.com
Vendor Bill	urn:purchases.transactions.webservices.NetSuite.com
Vendor	urn:relationships.lists.webservices.NetSuite.com

Mapping

If you need additional information regarding field to field mapping between the applications, please contact Crunchtime Support at support@crunchtime.com.

Error Handling and NetSuite Response Codes:

HTTP status codes are used to inform you about the success or failure of a request. The following status codes are used:

- 2xx status codes are used for successful requests.
- 4xx status codes are used for failures due to data or functionality error.
- 5xx status codes are used for failures due to system error, such as system unavailability.

© 2024 Crunchtime - All Rights Reserved