## **FISPAN Integration with NetSuite’s Bank Feeds API**

**Background:**

In the NetSuite 2020.1 update, NetSuite released APIs allowing for bank statement data to be auto-imported on a scheduled basis to the NetSuite reconciliation module. FISPAN has leveraged these APIs with a recent update to the FISPAN platform that connects to these APIs and transmits the end-user’s bank data on behalf of the bank.

A pilot was established with several FISPAN and NetSuite end-users to test the FISPAN’s connectivity to these new NetSuite APIs. Bank transaction information is requested from the end-user’s bank and made available for overnight syncing with the NetSuite Bank Feeds APIs.

**Pilot Client Feedback:**

Clients who set up the NetSuite bank statement parser feature to allow their BAI transactions to be synced from FISPAN cited concerns with the data being imported and the NetSuite functionality post data import. A collective decision made between the pilot end-users, their bank and FISPAN was to pause the pilot until the below feedback was raised with NetSuite and the bank.

**Pilot Client Concerns:**

1. Concerns about the bank data

Trial clients experienced issues related to syncing bank statement data on a nightly basis. With syncing this frequently, there is a client need for oversight to ensure that transactions that change after the 1-day window to be captured correctly. E.g. check amount corrections, uncleared check, chargebacks, other changes to transactions or amounts after one day.

The bank services ultimately connected to the NetSuite Bank Feeds are intended for multiple use cases, not just reconciliation. This data often needs massaging, filtering and editing before it is ready for ERP ingestion and reconcilement.

Requirements:

1. Pre-process the statement before use it in NetSuite
2. Get a copy of the statement for future process beside the reconciliation process

Actionable:

* It is possible to programmatically process the info in the plugin implementation
  + At the same time saving a copy of each transaction in a custom record it is possible
* Manually process the info for general proposes it is possible from the custom record

Manually pre-process the statements before getting into the reconciliation process it is not possible. Please clarify if it is needed a manual pre-process of the statements and why.

1. Duplicate detection was not as expected

Detection of duplicate transactions within the BAI data imported by NetSuite did not meet end-users expectations. The NetSuite import functionality has built-in duplicate detection, but the end-users piloting this experience felt that duplicate items previously imported were detected only 20% of the time (anecdotal).

Please clarify if this is happening for the first import or for the standard daily process. Does the process behave different from the plugin implementation than the standard manual import process of a .BAI file?

Please collect test data to be able to replicate the case.

~~Note: It is possible to reimplement the .BAI parser function and rewrite the duplicated function to work as expected.~~

1. Inability to modify or delete imported line items

Both 1. and 2. above lead to the following key concern. There is no ability to remove or delete duplicate or incorrect transaction items from the reconciliation tool. The full overnight import for that bank account needs to be deleted. Then the end-user is manually editing a BAI2 or csv file to remove duplicates, delete non-relevant BAI line items (e.g. five-day float) and manually import that file.

When multiplying this across many bank accounts, the process becomes arduous.

Duplicate and non-relevant: The reconciliation tool does allow to ignore transactions.

1. Only one bank account set up on a Bank Profile syncs overnight

When setting up a new bank profile on the bank statement parser, the UI allows the user to set up and match multiple bank accounts on one bank profile. However, only the data from one of the bank accounts, regardless of how many are set up, actually sync overnight.

FISPAN raised this issue with the NetSuite support teams. The workaround is to create multiple bank profiles, with one bank account set up in each. This was a negative user experience for the clients piloting and created additional setup steps in the case of 50+ bank accounts.

Case:

3888857 Financial Institution Format Profile only imports a single bank account 46298 FI.SPAN Services, Inc.

<https://nlcorp.app.netsuite.com/app/crm/support/supportcase.nl?id=87492849>

Description:

The user has one format profile that they want to be configured in a way that it can import multiple bank accounts at a time. Example: If BAI2 has more than 2 Account when it is linked to NetSuite GL Account. It only brought in transaction from 1 account.

Status Closed

1. Supports only one bank account per BAI

BAI2 file format supports multiple bank accounts in one BAI file. Pilot clients are unable to import a multi-account BAI into the Bank Statement parser. Only one account per BAI import is supported.

FISPAN are currently verifying this is the case

Solution: Get just one BAI file with all the accounts and use the default partner

Text, letter

Description automatically generated

1. Non-NetSuite-Accepted BAI codes

Users are citing error messages with BAI codes that are not accepted by NetSuite e.g. BAI2 code 425.

Also problematic and related to item 1 on this list, are BAI codes greater than 799.

**Further analysis and client feedback required on the below-proposed workarounds:**

The following document provides information on specifying which of the client’s transaction line items get pulled from the FISPAN database on per BAI code basis. I.e. defining the BAI codes which get imported, ignoring the others.

Financial Institution Connectivity Plugin: [https://docs.oracle.com/cloud/latest/netsuitecs\_gs/NSFIC/NSFIC.pdf](https://urldefense.proofpoint.com/v2/url?u=https-3A__docs.oracle.com_cloud_latest_netsuitecs-5Fgs_NSFIC_NSFIC.pdf&d=DwMF-g&c=SFszdw3oxIkTvaP4xmzq_apLU3uL-3SxdAPNkldf__Q&r=UI4uzd3eODJ82N6Bv3qeSubX1Yn4WG62HM5VEMGfvaw&m=e1KIhyBkTqALICZNvYELkrQ0oCeCy5aHwnYDdB9zgMU&s=1af9kudRMd1CkZ6N6ZjvGPO2TV8bsTU5S2hhhtRgvs0&e=)

The below documents allows for common 799+ BAI codes to be mapped to NetSuite supported codes (e.g. commonly used 868 can be mapped below 799) to allow for those transactions to be imported by the Bank Feeds parser.

Financial Institution Parser Plugin: [https://docs.oracle.com/cloud/latest/netsuitecs\_gs/NSPPL/NSPPL.pdf](https://urldefense.proofpoint.com/v2/url?u=https-3A__docs.oracle.com_cloud_latest_netsuitecs-5Fgs_NSPPL_NSPPL.pdf&d=DwMF-g&c=SFszdw3oxIkTvaP4xmzq_apLU3uL-3SxdAPNkldf__Q&r=UI4uzd3eODJ82N6Bv3qeSubX1Yn4WG62HM5VEMGfvaw&m=e1KIhyBkTqALICZNvYELkrQ0oCeCy5aHwnYDdB9zgMU&s=BNTmmIiCG07ZqaWgALJFyPxd2Gwg3Gt1mbM2kx4d2h0&e=)

Bank Statement Parser SuiteApp (Configuring BAI2 bank statement parser section): [https://docs.oracle.com/cloud/latest/netsuitecs\_gs/NSBNK/NSBNK.pdf](https://urldefense.proofpoint.com/v2/url?u=https-3A__docs.oracle.com_cloud_latest_netsuitecs-5Fgs_NSBNK_NSBNK.pdf&d=DwMF-g&c=SFszdw3oxIkTvaP4xmzq_apLU3uL-3SxdAPNkldf__Q&r=UI4uzd3eODJ82N6Bv3qeSubX1Yn4WG62HM5VEMGfvaw&m=e1KIhyBkTqALICZNvYELkrQ0oCeCy5aHwnYDdB9zgMU&s=ls1oyudiZwUc_0JCmrESVbfYmpN2eNCK6VrVTrsgA6M&e=)

Please collect test data to be able to replicate the case.

Does the annual import of a .BAI file behave in the same way? Or it happened just with the plugin?

~~Note: It is possible to reimplement the .BAI parser function.~~

1. Setup Documentation was not as expected

NetSuite documentation was not clear on the above setup issues in items 4 & 5 above.

FISPAN SuiteApp documentation can help remediate this issue

1. Other concerns

The timing of when the import occurs is of slight concern for some clients. Occasionally some bank accounts are not yet imported by start of day Eastern Time.

This may also be a concern for non-North American clients

Please ask the client to fill a case so we can audit what happening.

Can you validate if this happened because the process was not triggered, the process did finish or the process finish with errors?

Error feedback - no error message inside the ERP/on import screen. Status screen just says the import has failed. You have to check the parser logs to see the actual error code

Why would a final user need to see logs for an integration process error?

Blocker to migrate from Bundler to SDF?

Case:

4222770 Enhancement #627219: SDF > Error: The script ID xx referenced by object field "sourcefrom" is missing. 46298 FI.SPAN Services, Inc.

<https://nlcorp.app.netsuite.com/app/crm/support/supportcase.nl?id=106422397>

Enhancement:

627219 SDF > CLI > project:adddependencies > Error > SDF [ACP] Validation failed.

<https://nlcorp.app.netsuite.com/app/crm/support/issuedb/issue.nl?id=105857455&l=T>

Description:

Purpose: The Customer is currently encountering an error (The Script ID "[scriptid=xxxxx]" referenced by object field "sourcefrom" is missing) when they are trying to package a project via SDF CLI Java which is due to the Local Validation that runs when the "package" command is used.  
Impact: With this error, the Customer has not been able to transition from SuiteBundler to SuiteApp Control Center which should have happened last May 13, 2021. According to the details of the Defect record, the current workaround is to manually zip the SuiteApp project folder on Windows Explorer to upload it to the marketplace.  
-- MJMA, May 20, 2021

Daniel and Terry enhancement requests:

Case:

4241519 Enhancement: 630245 Please allow individual transactions inside Bank Statement records to be editable by users 46298 FI.SPAN Services, Inc.

<https://nlcorp.app.netsuite.com/app/crm/support/supportcase.nl?id=107660511&whence=>

Enhancement:

630245 Please allow individual transactions inside Bank Statement records to be editable by users

<https://nlcorp.app.netsuite.com/app/crm/support/issuedb/issue.nl?id=107657891&l=T>

Description:

Currently, after a user imports bank statement data, he/she cannot edit individual transaction lines inside a bank statement record in the Match Bank Data page. Please allow users to edit them inside the NS product once the data is uploaded to NS via FI plugin or manual import. Without this feature, uses need to first manually make the changes in the bank files before importing them into NS. Thus allowing editing inside NS provides a better user experience.

Case:

4241528 Enhancement: 630248 Please provide a programmatic way to update Bank Statement records 46298 FI.SPAN Services, Inc.

<https://nlcorp.app.netsuite.com/app/crm/support/supportcase.nl?id=107660731&l=T>

Enhancement:

630248 Please provide a programmatic way to update Bank Statement records

<https://nlcorp.app.netsuite.com/app/crm/support/issuedb/issue.nl?id=107659066>

Description:

A SDN partner has developed a bank integration using the FI plugin. Bank transaction number 123 with amount $10 was made in day 1, and the data was imported into NS using the plugin. After 3 days the bank updated the amount of transaction 123 from $10 to $9. The plugin now needs to be able to propagate this change back to the Bank Statement record inside NS.

Enhancement:

592360 FI Connectivity Plugin - automated daily import schedule to be customizable

Description:

FI Connectivity Plugin - automated daily import schedule to be customizable

Cases:

Graphical user interface, table

Description automatically generated

Graphical user interface, text

Description automatically generated

~~Case: (Issue 630249 is duplicate of enhancement 592360)~~

~~4241534 Enhancement: 630249 Please allow users to configure the time and frequency of a FI plugin's execution 46298 FI.SPAN Services, Inc.~~

[~~https://nlcorp.app.netsuite.com/app/crm/support/supportcase.nl?id=107660928&whence=~~](https://nlcorp.app.netsuite.com/app/crm/support/supportcase.nl?id=107660928&whence=)

~~Enhancement~~

~~630249 Please allow users to configure the time and frequency of a FI plugin's execution~~

[~~https://nlcorp.app.netsuite.com/app/crm/support/issuedb/issue.nl?id=86150277&l=T~~](https://nlcorp.app.netsuite.com/app/crm/support/issuedb/issue.nl?id=86150277&l=T)

~~Description:~~

~~Currently, there are two ways to trigger the execution of a FI plugin: manual trigger by user, or rely on NS system to execute the plugin on a daily basis. The latter method is too unpredictable and opaque for a SDN partner to provide a reliable solution. It is not clear in what time during the day the plugin is triggered, therefore it is not easy for an external party to build infrastructure to service incoming calls from NS FI plugins.  
Ideally, an administrator and/or plugin developer can configure the time of each FI plugin's daily execution, and the frequency of execution.~~