



# Solution Guide

SuiteSolutions - CSV Integrator

**Solution Version: 2.2.0**



© 2017 NetSuite Inc.

Any reproduction or distribution of any part of this document without the prior written permission of NetSuite Inc. is strictly prohibited.

The information included in this document is confidential and proprietary information of NetSuite Inc.

**Document Version: 1.0, July 28, 2017**

NetSuite Inc.  
2955 Campus Drive, Suite 100  
San Mateo, CA. 99403-2511

## Table of Contents

1	Document Overview .....	4
1.1	Symbols and Conventions.....	4
1.2	Terms and Definitions .....	4
2	Solution Overview.....	5
2.1	Supported Features .....	5
3	Setup and Configuration .....	7
3.1	Bundle Installation.....	7
3.2	Feature Dependencies.....	7
3.2.1	SuiteCloud Features.....	7
3.2.2	Token Based Authentication.....	7
3.3	JAVA Environment.....	7
3.4	Create an Application for Token Based Authentication.....	8
3.5	Create Access Token for a User .....	9
3.6	Deploy the Script to Preferred Roles .....	9
3.7	Get the Broker Files .....	9
3.8	Broker Configuration .....	10
3.8.1	Create Folders.....	10
3.8.2	Define the Configuration File .....	10
3.8.3	Specify the Queue Number .....	12
3.8.4	Specify the CSV File Delimiter.....	13
3.8.5	Specify the CSV File Maximum Number of Rows .....	14
3.9	Create Saved CSV Import.....	14
3.10	Create an Integration Mapping Record.....	15
3.11	CSV File Format .....	16
3.12	Using the Integrator.....	16
3.12.1	Using the Integrator Broker User Interface - Local.....	16
3.12.2	Using the Integrator Broker User Interface - FTP.....	18
3.12.3	Using the Integrator Broker User Interface - SFTP .....	20
3.12.4	Using the Integrator Console Interface .....	22
3.12.5	Using the Integrator via Windows Scheduler .....	23
3.13	CSV File with More Than 24 999 Rows.....	23

3.14 Upload a JE CSV file with Debit and Credit rows NOT balanced.....	23
3.15 Upload a JE CSV file with Debit and Credit rows balanced .....	23
3.16 Error in Process .....	23
3.17 RUN VIA UI.....	23
3.18 Bundle Components.....	24
3.18.1 Custom Lists/Records .....	24
3.18.2 CSV Imports.....	24
3.18.3 Scripts .....	24
3.18.4 Roles and Permission.....	24
3.18.5 Windows Map Network Drive.....	26
3.19 Limitations.....	28
3.20 Solution Design.....	28
3.20.1 RESTlet SuiteScript Automation to Import Files.....	28
3.20.2 NetSuite Properties Configuration File.....	29

# 1 Document Overview

SuiteSolutions are pre-built offerings used to accelerate the delivery of custom functionality into Customer's account. These solutions were designed and developed by NetSuite Professional Services. SuiteSolutions are delivered as Non-Managed bundles i.e. the custom code/configurations delivered with the bundle is not locked and can be further customized on a customer account to meet unique business requirements. For the Terms of Service for SuiteSolutions, refer to <http://www.netsuite.com/tos>. Notwithstanding anything to the contrary in this Solutions Guide or on SuiteAnswers, in the event of any conflict between this Solution Guide and the Terms of Service for SuiteSolutions, the Terms of Service for SuiteSolutions shall supersede and control.

This is the Solution Guide for SuiteSolution - CSV Integrator. This documentation will guide the user how to setup, configure and work with the solution with the given use cases/scenarios as examples.

## 1.1 Symbols and Conventions

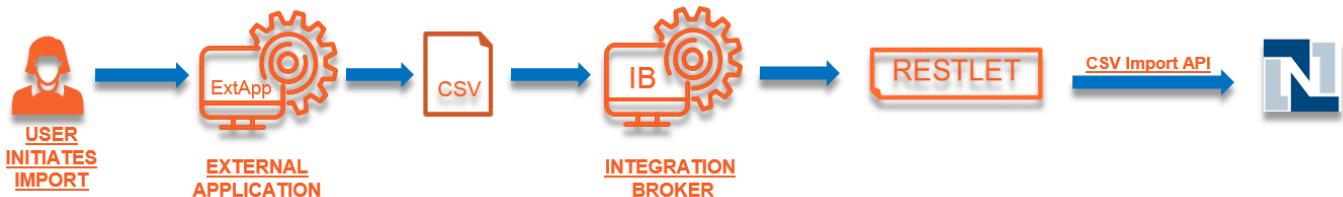
Symbol	Description
	Indicates situation the user should be specifically aware of when completing a task.
	Indicates helpful tips, shortcuts and suggestions. The 'bulb' icon explains general information around NetSuite while the 'record' icon gives important details related to records, fields, forms and validations.

Convention	Description
<b>Setup &gt; Custom &gt; CSV Integration Mapping</b>	The (>) symbol indicates a sequence of actions, such as selecting items from a menu or toolbar, or pressing buttons in a window. This example directs you to go to <b>Setup</b> tab and hover to <b>Custom</b> and select <b>CSV Integration Mapping</b>

## 1.2 Terms and Definitions

Term	Definition
<b>Integration Broker</b>	A client external module hosted outside the NetSuite application and developed using core JAVA technology as an executable file (.exe). This can be run using the UI or the backend batch process.
<b>RESTlet</b>	A form of web services integration. Each RESTlet is a server-side script that operates in a request-response model, and is invoked by an HTTP request to a system-generated URL.
<b>Mapping Record</b>	Custom record that stores the CSV Import Mapping IDs

## 2 Solution Overview



**Figure 1 – CSV Integrator Process Flow**

The integration is built using SuiteScript RESTlet and a broker in JAVA technologies to invoke the RESTlet. Below is an overview of the solution:

- The Integration Broker reads the CSV files generated by the external applications on SFTP Server, FTP Server or local folder and sends the raw data as JSON string to the RESTlet.
- The RESTlet will then convert the raw string into chunks of 25K (based on rows) and will invoke the CSV Import API to initiate the import process.
- \*The CSV Import will be added to the import queue and functions as any other NetSuite CSV Import jobs.
- Once the import has been processed, the user will be notified of the import status using standard NetSuite functionality. The JOB ID will be sent back to the broker.
  - Once a success response is received, the CSV file will be moved into a processed folder.
  - CSV files from the SFTP server or FTP server will also be moved on the processed folder on the server and will be deleted on the original folder.
  - In case of errors, the file will be moved to the error folder and an error log will be added.

\*For SuiteCloud PLUS Licensed Accounts, a specified queue number (2-5) can be set on the CSV import.

### 2.1 Supported Features

The following features are supported by the solution.

1. One Integration solution to import records supported by CSV Import functionality into NetSuite. Batch integration for periodic import of data from external applications.
  - On-demand via User Interface.
  - Only CSV File types are supported.
2. Allows to import files more than 25K in size, it can split up file into chunks of 25k. Also it makes sure that the transaction lines are not split across files.
3. Multiple file types can be imported via one source location using the default Broker.
4. Default Broker provided with the solution. The Broker retrieves the mapping id information of the Saved CSV Import based on the column headers.
5. Uses the standard CSV mapping feature in the CSV Import. Mapping should be created manually and added in the mapping definition record.

6. Supports both email/login based and Token-based Authentication for the authentication. Token-based authentication is recommended but the user can continue to use the standard method of authentication if they choose to.
7. Multiple CSV files can be imported via one source location using the default Broker but only one mapping is allowed for each record type. Also for each record, a primary field can be specified in the mapping record to allow the broker to split up the files.
8. The solution can download files from FTP and SFTP locations and import into NetSuite. The successful import processes are then moved to the processed folder on the FTP/SFTP site if write access is enabled.
9. Supports multi-queue execution of the import processes. The CSV Import mapping has a queue number and this solution allows to provide a queue number if there is a desire to run the import on specific queues.
10. Includes support for character encodings for standard NetSuite login and token-based authentication (See Table 1).

<b>Character Encoding</b>	<b>Standard NetSuite Login</b>	<b>Token-based Authentication</b>
Unicode (UTF-8)	Supported	Supported *
Western (Windows 1252)	Supported	Not Supported
Western (ISO-8859-1)	Supported	Supported *
Chinese Simplified (GB18030)	Supported	Not Supported
Chinese Simplified (GBK)	Supported	Not Supported
Traditional Chinese (Big5)	Supported	Not Supported
Japanese(Shift-JIS)	Supported	Not Supported
Western (Mac Roman)	Supported	Not Supported
Korean (ISO-2022-KR)	Not Supported	Not Supported
Korean (EUC-KR)	Supported	Not Supported
Traditional Chinese (Big5)	Supported	Not Supported

**Table 1 – Supported Character Encodings for Login Types**

\* - only Basic Latin Characters are supported

## 3 Setup and Configuration

### 3.1 Bundle Installation

To install the SuiteSolutions – CSV Integrator, a user has to have an Administrator or SuiteBundler permission in an account.

1. Go to **Customization > SuiteBundler > Search & Install Bundles**.
2. Search for the **SuiteSolutions – CSV Integrator** bundle and click the name.
3. Click the **Install** button to begin the installation process.

Please refer to these NetSuite Help links for more information about bundle installation: [Installing a Bundle](#), [Choosing a Bundle to Install](#), [Understanding Bundle Searches](#).



It is recommended that process is validated in a sandbox prior to installation in a production account.

### 3.2 Feature Dependencies

The following features needs to be enabled to the account:

#### 3.2.1 SuiteCloud Features

- Client SuiteScript
- Server SuiteScript
- Custom Records

#### 3.2.2 Token Based Authentication

Enable the Token-Based Authentication by going to **Setup > Company > Enable Features** and under SuiteCloud tab, check the **Token-Based Authentication** field on the Manage Authentication section.

[TOKEN-BASED AUTHENTICATION](#)

ENABLE TOKEN-BASED AUTHENTICATION AS AN ADDITIONAL AUTHENTICATION MECHANISM FOR YOUR USERS. BY ENABLING THIS FEATURE, YOU AGREE TO [SUITECLOUD TERMS OF SERVICE](#).

Figure 2 - Enable Token Based Authentication

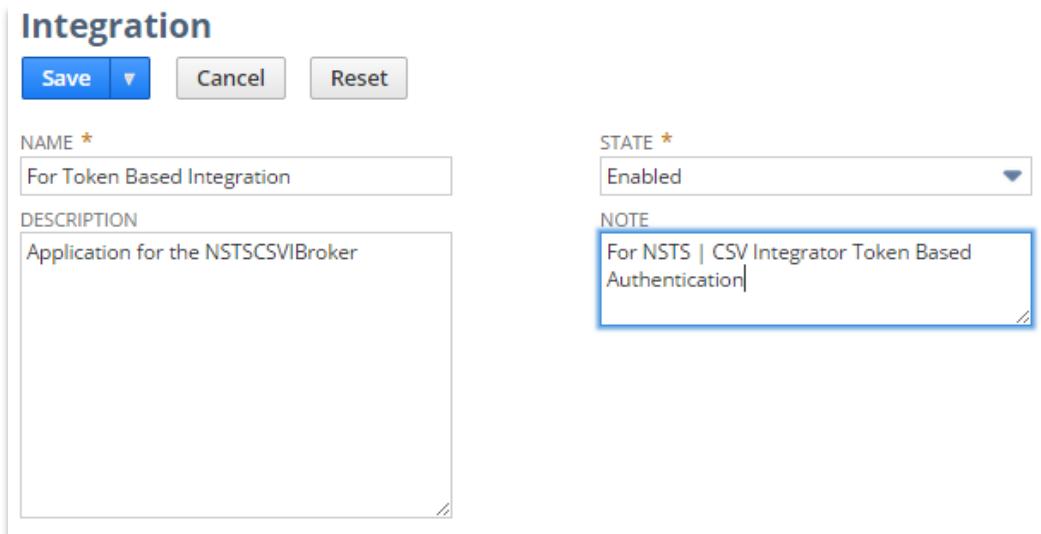
### 3.3 JAVA Environment

Requires installed JAVA environment. (E.g., JRE 1.7). The Java Software installed must be above version 7. The folder path where JAVA is located must be set on the Run Via Console.bat file or Run Via UI.bat file, see [Get Broker Files](#).

### 3.4 Create an Application for Token Based Authentication

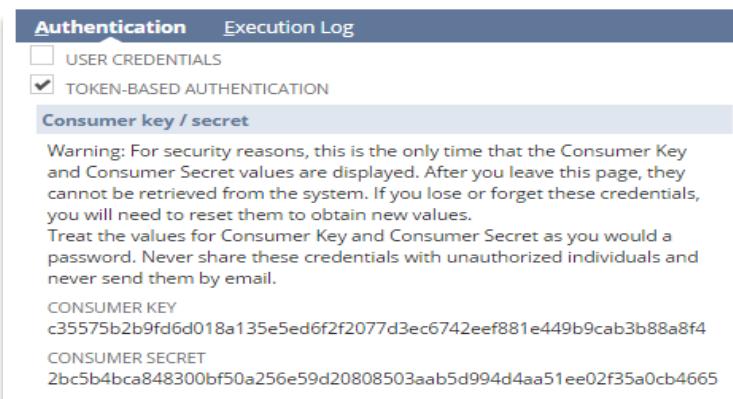
After enabling the Token-based Authentication, an application must be created for it. Below steps describes how to create an application for Token Based Authentication.

1. Go to **Setup > Integration > Manage Integrations > New**. The new Integration entry form will display.
2. Enter a name for the application.
3. The State is enabled by default.
4. On the authentication subtab:
  - a. Un-check the User Credentials checkbox.
  - b. Check the Token-Based Authentication checkbox.
5. Click **Save**.



**Figure 3 - Integration for Token-based Authentication**

6. After saving, the consumer key and consumer secret will display for the application under the Authentication subtab.

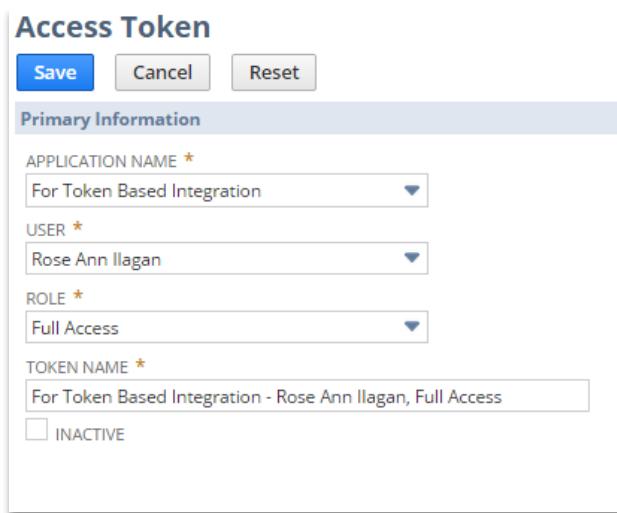


**Figure 4 - Generated Consumer Key/Secret**

7. The consumer key and consumer secret must be copied before leaving the page. These keys will only be displayed once and cannot be retrieved from the system.

### 3.5 Create Access Token for a User

1. Go to **Setup > User/Roles > Access Tokens > New**
2. On the Application Name, select the created [application](#).
3. On the User field, select the name of the user to grant access.
4. On the Role field, select the type of access for the selected user.
5. **Save.**



**Figure 5 - Access Token for User**

### 3.6 Deploy the Script to Preferred Roles

The RESTlet must be deployed only to roles allowed to do the importing. Below steps describes how to deployment to preferred roles.

1. Open the **Base CSV Integrator RS** RESTlet via [Customization> Scripting> Scripts](#)
2. On the Deployments tab, open and edit the **Base CSV Integrator RS** deployment.
3. From the Audience section, select the preferred roles by unchecking the Select All checkbox.
4. Press CTRL to multi-select the preferred roles.
5. Save the deployment.

The selected role must also be specified also in the [configuration file](#). It is not recommended to store an Administrator user/password in the properties file.

### 3.7 Get the Broker Files

Get the broker and console files from the bundle by going to *Documents > Files > File Cabinet* and find the bundle ID of the solution. Click the **NSTSCSVIBroker.zip** link to download the files.

## Contents:

1. netsuite.properties
2. lib
3. Sample CSV Files
4. README.txt
5. NSTSCSVIBroker.jar
6. Run Via Console.bat
7. Run Via UI.bat

## 3.8 Broker Configuration

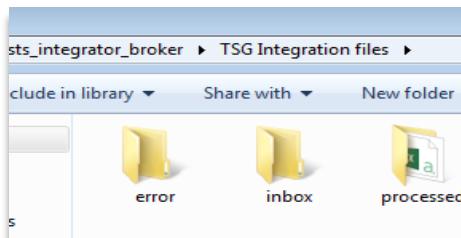
### 3.8.1 Create Folders

Create the Inbox, Processed and Error folders and get the folder directories.

<b>Inbox Folder</b>	This is the folder where the Broker User Interface via FTP or SFTP and Console Interface will fetch the csv file for processing.
<b>Processed Folder</b>	This is the folder where the Broker User Interface via FTP or SFTP and Console Interface will transfer the processed csv file.
<b>Error Folder</b>	This is the folder where the Broker User Interface via FTP or SFTP and Console Interface will transfer csv files if an import error is encountered.

Example:

.\\TSG Integration files\\inbox  
 .\\TSG Integration files\\processed  
 .\\TSG Integration files\\error



**For Linux OS:** The directory path for Linux OS should be entered with forward slash (/) and the folder should have read/write access upon creation. Another forward slash (/) should be added after the folder name.

**For Windows OS:** The directory path for Windows OS should be entered with double backslashes (\\\). Another double backslash (\\) should be added after the folder name.

### 3.8.2 Define the Configuration File

Create a properties file if no properties file exists yet and set the following details. Save it as **netsuite.properties**.

See [NetSuite Properties Configuration File](#) for a sample properties file.

NetSuite Connection	
<b>netsuite.ws.url</b>	Web Service URL. This varies on the account type (production, sandbox)
<b>netsuite.account</b>	NetSuite company account.
<b>netsuite.email</b>	Email address used to log in on NetSuite account.

<b>netsuite.password</b>	Password used to log in on NetSuite account.
<b>netsuite.role</b>	Role id to log in. Please see <a href="#">NetSuite Help</a> for the list of IDs and <a href="#">Deploy the Script to Preferred Roles</a> for more details.
<b>netsuite.login</b>	Login preference, <b>OAUTH</b> for token based authentication and <b>NLAUTH</b> for standard login.
<b>netsuite.csvStorage</b>	CSV files storage, <b>FTP</b> if the files are to be read on ftp server, <b>LOCAL</b> if files are in local directory, this is set if application is executed on console.
<b>netsuite.suiteCloudAcct</b>	Set to <b>true</b> ONLY IF account is SuiteCloud Licensed Account to support multiQueueing of CSV imports, this is set if application is executed on console, else set this to <b>false</b> .
<b>netsuite.csvImportQueue</b>	See <a href="#">Specify the Queue Number</a> .
<b>netsuite.dateFormat</b>	Set the date format to be used for the log file name.
<b>netsuite.datetimeformat</b>	Set the date-time format to be used for the logging entries on the log file.
<b>netsuite.csvFileDelimiter</b>	Define the CSV File Delimiter. This is only for the file with more than 25,000 records. Please refer to the Solution Guide for the details.
<b>netsuite.maxRow</b>	Define the maximum number of rows by which the splitting shall be based from
<b>Token Based Authentication</b>	
<b>netsuite.consumerKey</b>	Consumer id generated for the application.
<b>netsuite.consumerSecret</b>	Consumer secret generated for the application.
<b>netsuite tokenId</b>	Token id generated for the user/role.
<b>netsuite.tokenSecret</b>	Token secret generated for the user/role.
<b>FTP and SFTP</b>	
<b>netsuite.ftp.username</b>	FTP username.
<b>netsuite.ftp.password</b>	FTP password.
<b>netsuite.ftp.host</b>	FTP host is required.
<b>netsuite.ftp.path</b>	FTP path: location of CSV files, must be a directory.
<b>netsuite.ftp.port</b>	FTP port (optional).
<b>netsuite.ftp.removeProcessedFile</b>	This is enabled when CSV files processed will be removed on the current Directory and move to 'processed'/error file.
<b>netsuite.ftp.errorDir</b>	Directory for the processed CSV files that were not successfully imported.
<b>netsuite.ftp.processedDir</b>	Directory for the processed CSV files that were successfully imported.
<b>netsuite.sftp.knownhostpath</b>	Set the path that contains the file for known host. This is used to verify the server's legitimacy that is provided by SSH.
<b>netsuite.sftp.privatekeypath</b>	Set the path that contains the file for private key. This is to be paired with the server's public key.
<b>netsuite.sftp.passphrase</b>	Set the value for the passphrase. This is used for the encryption of the private key.
<b>netsuite.tsg.Integration.restlet.url</b>	Relative Path URL of the RESTLet from the deployment script.

<b>baselogdir</b>	.\\logs\\
<b>execution.log</b>	executionLog.txt
<b>success.log</b>	successLog.txt
<b>failure.log</b>	failureLog.txt
<b>files.path.inbox</b>	File input directory. See <a href="#">Create Folders</a> .
<b>files.path.processed</b>	File processed directory. See <a href="#">Create Folders</a> .
<b>files.path.error</b>	File error directory. See <a href="#">Create Folders</a> .
<b>netsuite.tsg.JobName</b>	"SuiteSolution Demo: {compname} - date : {datetime} import service   CSV rows: {csvrows}   User : {user}   File Name: {fileName}"

**Note:** When using Linux based OS, the following folders must be manually created by the user with the read/write permission:

- **baselogdir** – for defining the path separator, use forward-slash or "/" and add another "/" at the end of the folder name. (Example: `./logs/`)
- **files.path.inbox** – See [Create Folders](#).
- **files.path.processed** – See [Create Folders](#).
- **files.path.error** – See [Create Folders](#).

### 3.8.3 Specify the Queue Number

By default, all CSV import jobs use a single queue. This queue is defined as queue 1. In accounts that have purchased a SuiteCloud Plus license, five queues are available for import jobs, instead of a single queue. For CSV Integrator implementation with 'SuiteCloud Plus' Licensed Account, since there are 5 queues only, 1 is reserved and the user can specify the queue numbers 2-5 in the [configuration file](#).

```

netsuite.login = OAUTH
netsuite.csvStorage = LOCAL
netsuite.suiteCloudAcct = true
netsuite.csvImportQueue = 5

```

Figure 6 - Queue Number in the Configuration File

The broker will throw an error if the specified queue number is more than 5 or equal to 1.

Error messages:

- ERROR: CSV import queue number must be within 2-5
- ERROR: Invalid queue number (for non-integer value)
- ERROR: CSV import queue is required for multiQueueing. (if set to SuiteCloud Plus Licensed Account but no queue number is specified)

### 3.8.4 Specify the CSV File Delimiter

Version 2.2.0 of the CSV Integrator introduces the ability to specify the CSV File Delimiter of the file(s) to be imported, as long as the file delimiter is within the allowed CSV Column Delimiter of the Import Assistant of NetSuite.

**Scan & Upload CSV File**

**IMPORT TYPE**  
  
 Choose the category of data to import.

**RECORD TYPE**  
  
 Choose the record type of data to import.

**CHARACTER ENCODING**  
  
 Choose another character encoding format if you use an inter...

**CSV COLUMN DELIMITER**  
  
  
 Semicolon  
 Pipe  
 Space  
 Tab

**Figure 7 - Sample Saved CSV Import**

In *netsuite.properties* file, in **netsuite.csvFileDelimiter** property, the corresponding values for each CSV Column Delimiter are as follows:

NetSuite CSV Column Delimiter	Value	Remarks
Comma	,	
Pipe	\	Double backlashes + pipe
Semicolon	;	
Space	\s	Double backlashes + 's'
Tab	\t	Single backslash + 't'

### 3.8.5 Specify the CSV File Maximum Number of Rows

Version 2.2.0 of the CSV Integrator introduces the ability to specify the Maximum Number of Rows to be processed. This number shall be the maximum number of rows or lines a single chunk of data contains. This is the basis for the splitting of data.

The value entered in `netsuite.maxRow` should not exceed 24,999.

### 3.9 Create Saved CSV Import

Each record to be imported needs a CSV Import saved and should have a Script ID defined for each. Provide a Script ID for the Saved CSV Import to be used.

Create a Saved CSV Import by going to *Setup > Import/Export > Saved CSV Imports > New*. Follow the steps to set up and create a new import, then Save.

Save mapping & Start Import

Your CSV files are ready to be imported into your NetSuite account.  
You can save your Field Mapping to be used again later (optional). To work w

IMPORT MAP NAME

Provide a unique Import Map name to save the current mapping choices for :

DESCRIPTION

(Optional) Provide a useful description for this Import Map.

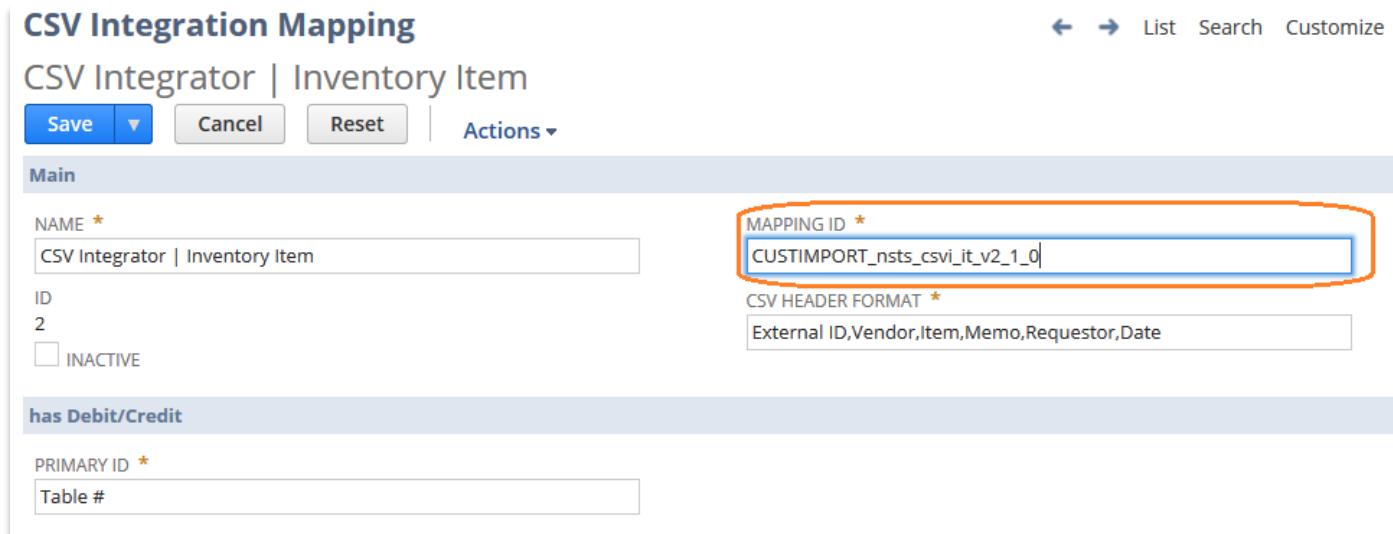
SCRIPT ID

(Optional) Provide a unique ID to identify this Import Map in SuiteScript.  
Your entry here is prefixed with CUSTIMPORT.

Figure 8 - Sample Saved CSV Import

### 3.10 Create an Integration Mapping Record

This is the record to store the Saved CSV Import Template ID. An Integration Mapping record must be defined for each Saved CSV ID. To create a mapping record, go to **Setup > Custom > CSV Integration Mapping > New.**



**Figure 9 - Sample Integration Mapping Record**

Set the following fields:

- **Name** - Name of the mapping record.
- **Mapping ID** - From generated Script ID after [Saved CSV Import](#) creation.
- **CSV Header Format** – Actual header names of the uploaded CSV file.
- **Primary ID (has Debit/Credit)**

Save the mapping record.



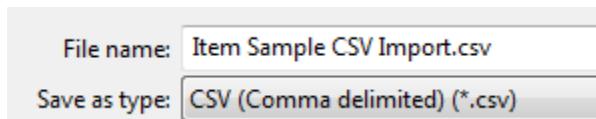
The user/role to be used on running the application must have read/write or view access to the mapping record.



Ensure there are no blank spaces between column headers. Having spaces will result in the following error while running CSV Broker: **ERROR: Invalid SCV header format**

### 3.11 CSV File Format

Ensure that the file to be uploaded is saved in **.csv** format.



On the MS Excel application, go to **File > Save As** and select the **CSV (Comma delimited) (\*.csv)** as the 'Save as type'.

Working with CSV Integrator

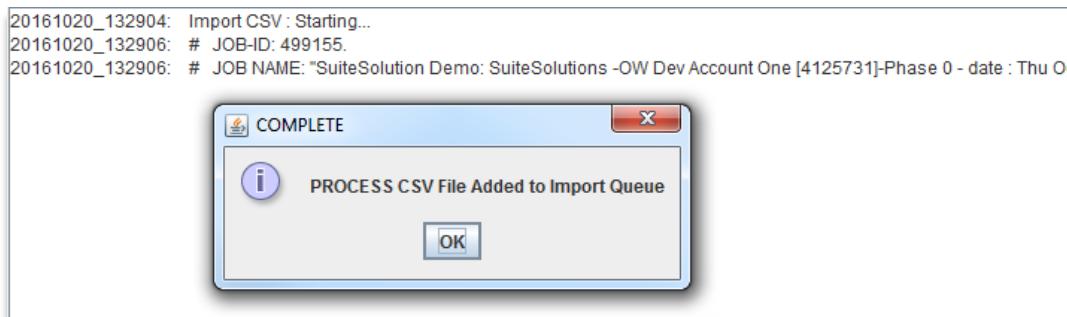
### 3.12 Using the Integrator

#### 3.12.1 Using the Integrator Broker User Interface - Local

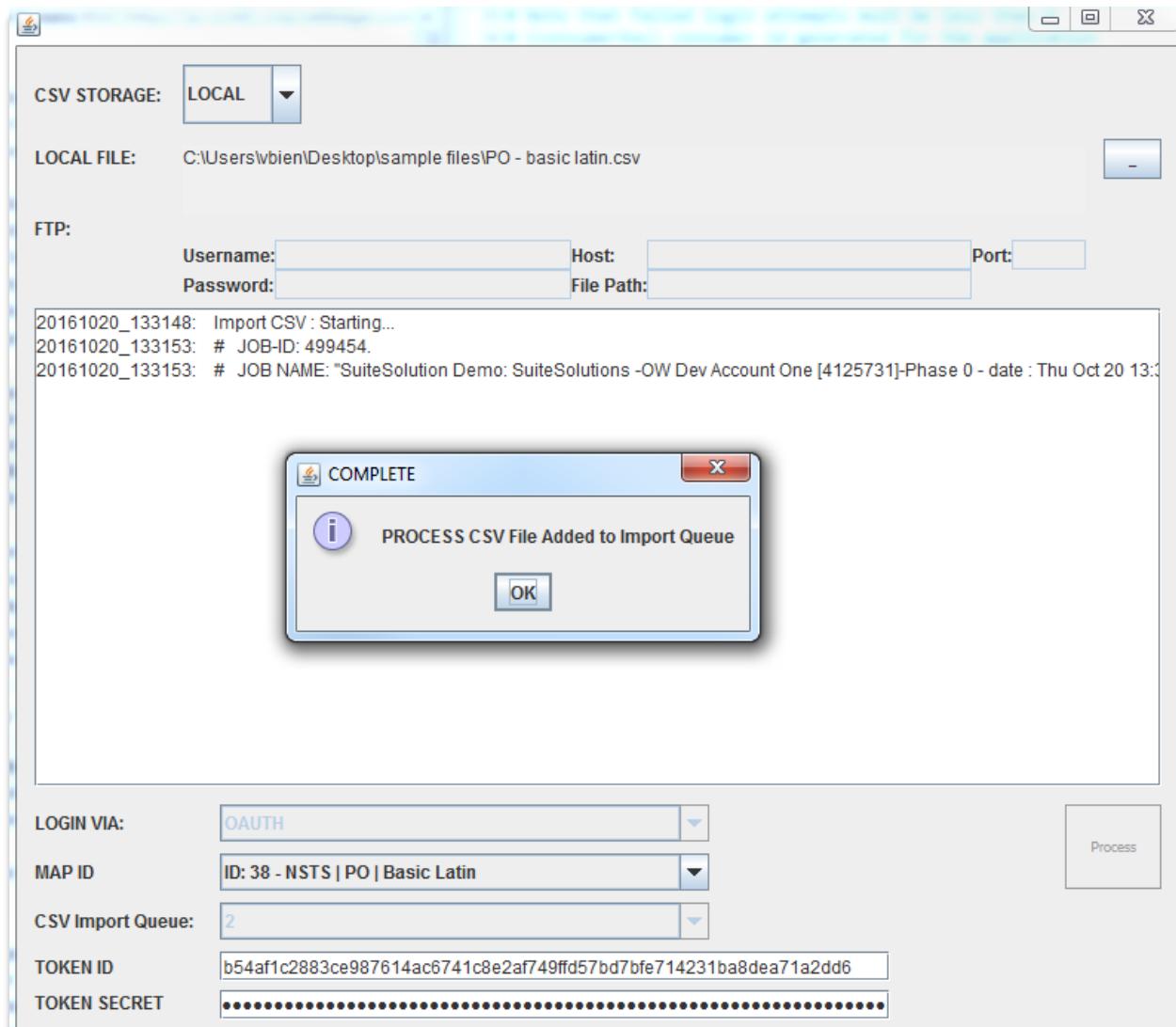
##### Pre-requisite:

- [CSV file for import.](#)
- [Broker Configuration.](#)
- [Saved CSV Import.](#)
- [Integration Mapping Record.](#)

1. Locate and open the **NSTSCSVIBroker.jar** file downloaded from the [File Cabinet](#).
2. The Broker User Interface will open.
3. On **CSV Storage**, dropdown, select **Local** on the CSV Storage dropdown.
4. On **Login Via**, the default value will be based on the login type specified on the configuration file:
  - **OAUTH** to login with authentication or
  - **NLAUTH** to login without authentication.
5. On **MAP ID**, select the [mapping record](#) created for the CSV.
6. **CSV Import Queue** depends on # specified in the properties file.
7. Click the browse button and locate the CSV file to import.
8. The file directory of the selected file will display on the Local File field.
9. Click the Process button.
10. A message box will display for status of process.
11. The JOB ID and import details will display on the log box of the UI.
12. A new job will be added to the CSV Import Job Status in NetSuite.



**Figure 10 - Import Status Message Box**



**Figure 11- Broker User Interface – Local**

Job Status						
		TOTAL				
DATE	JOB NAME	STATUS	PERCENT COMPLETE	MESSAGE	CSV RESPONSE	QUEUE
10/16/2015 12:47 am	"SuiteSolution Demo: SuiteSolutions Dev Account Two - date : Fri Oct 16 15:47:37 SGT 2015 import service   CSV rows: 169   User : rilagan@netsuite.com   File Name: ITM Import.csv" Part:1	Complete	100.0%	168 of 168 records imported successfully	CSV Response	2

**Figure 128 - CSV Import JOB Status**

Note: The processed file will not be moved to the 'Processed' folder.

### 3.12.2 Using the Integrator Broker User Interface - FTP

#### Pre-requisite:

- CSV file for import.
- [Broker Configuration](#).
- [Saved CSV Import](#).
- [Integration Mapping Record](#).

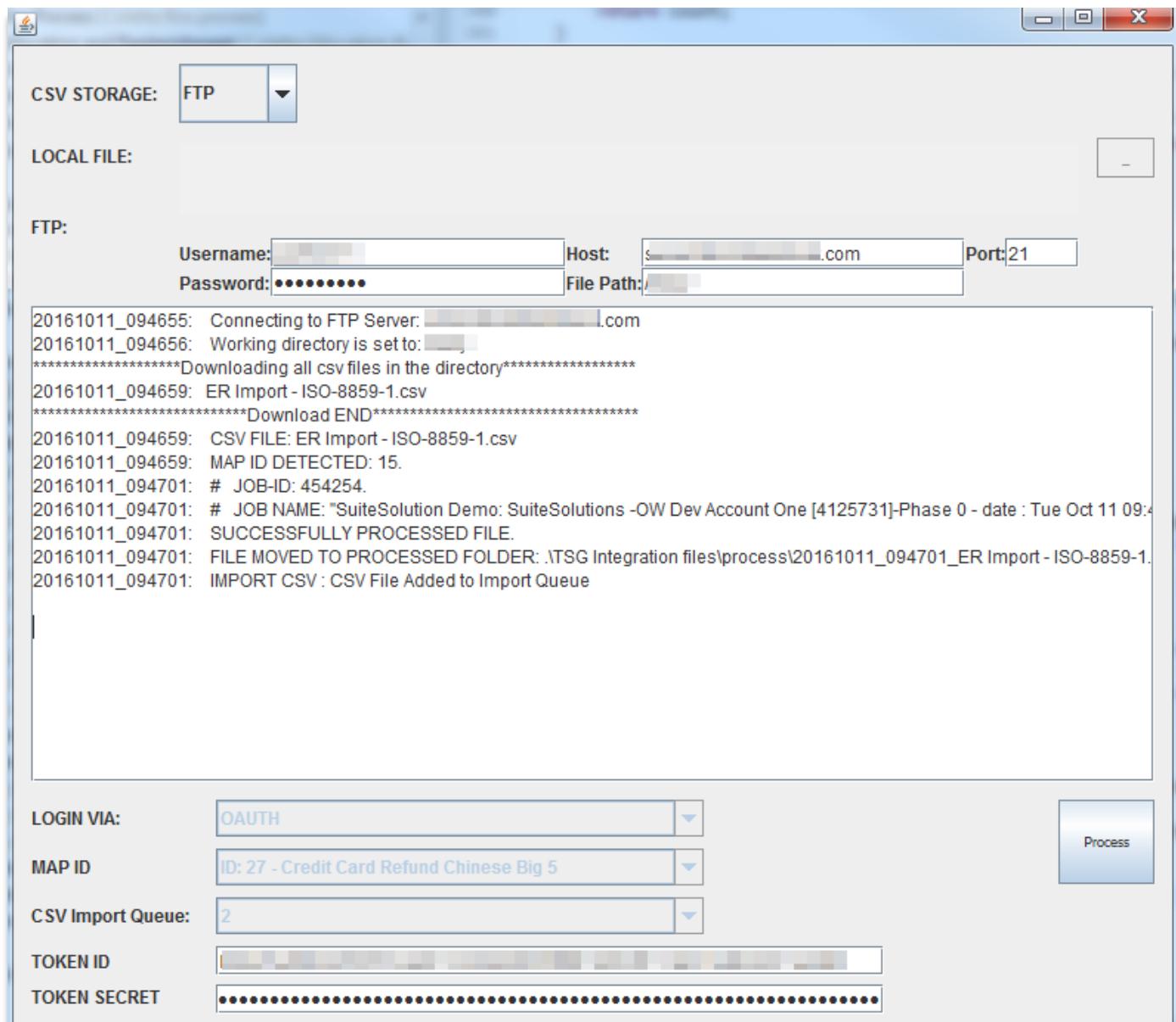
1. Locate and open the **NSTSCSVIBroker.jar** file downloaded from the [File Cabinet](#).
2. The Broker User Interface will open.
3. On **CSV Storage**, dropdown, select **FTP** on the CSV Storage dropdown.
4. The browse button will be disabled.
5. Enter the following FTP details:

a. Username	d. File Path
b. Password	e. Port
c. Host	
6. On **Login Via**, the default value will be based on the login type specified on the configuration file:
  - **OAuth** to login with authentication or
  - **NLAUTH** to login without authentication.
7. **CSV Import Queue** depends on # specified in the properties file.
8. Click the Process button.
9. The broker will download the files from the specified file path and will be moved to the local inbox folder specified.
10. A message box will display for status of process.
11. The JOB ID and import details will display on the log box of the UI.
12. The UI will display "Import CSV: CSV File Added to Import Queue".
13. A new job will be added to the CSV Import Job Status in NetSuite.
14. The processed files will be moved to the local 'Processed' folder specified. If files had error upon import, it will be added moved on the local 'Error' folder specified.
15. If 'removeProcessedFile' parameter is set to true, the files will be moved to the specified 'error' and 'processed' folder on the FTP Server. The user must have a write access to the FTP server to successfully move the files.

When the broker is executed on a scheduled interval and 'removeProcessedFile' is set to false, it will always import the CSV files on the FTP inbox folder even those that were processed before, unless manually deleted on the FTP inbox folder.



The broker will automatically search for the integration mapping using the file's header. Kindly ensure that the CSV Header Format in the file's corresponding Integration Mapping is unique.



**Figure 13 - Integrator Broker User Interface - FTP**

### 3.12.3 Using the Integrator Broker User Interface - SFTP

#### Pre-requisite:

- CSV file for import.
- [Broker Configuration](#).
- [Saved CSV Import](#).
- [Integration Mapping Record](#).

1. Locate and open the **NSTSCSVIBroker.jar** file downloaded from the [File Cabinet](#).
2. The Broker User Interface will open.
3. On **CSV Storage**, dropdown, select **SFTP** on the CSV Storage dropdown.
4. The browse button will be disabled.
5. Enter the following SFTP details:

a. Username	d. File Path
b. Password	e. Port
c. Host	
6. On **Login Via**, the default value will be based on the login type specified on the configuration file:
  - **OAUTH** to login with authentication or
  - **NLAUTH** to login without authentication.
7. The following properties should be specified on the configuration file:
  - **netsuite.sftp.knownhostpath (e.g. C:\Users\vbien\known\_hosts\_NS)**

Note: The file for this path should derived from the generated file of the **ssh-keyscan**.

    - **netsuite.sftp.privatekeypath (e.g. C:\Users\vbien\id\_rsa)**
    - **netsuite.sftp.passphrase**
8. **CSV Import Queue** depends on # specified in the properties file.
9. Click the Process button.
10. The broker will download the files from the specified file path and will be moved to the local inbox folder specified.
11. A message box will display for status of process.
12. The JOB ID and import details will display on the log box of the UI.
13. The UI will display "Import CSV: CSV File Added to Import Queue".
14. A new job will be added to the CSV Import Job Status in NetSuite.
15. The processed files will be moved to the local 'Processed' folder specified. If files had error upon import, it will be added moved on the local 'Error' folder specified.
16. If 'removeProcessedFile' parameter is set to true, the files will be moved to the specified 'error' and 'processed' folder on the SFTP Server. The user must have a write access to the SFTP server to successfully move the files.

When the broker is executed on a scheduled interval and 'removeProcessedFile' is set to false, it will always import the CSV files on the SFTP inbox folder even those that were processed before, unless manually deleted on the SFTP inbox folder.

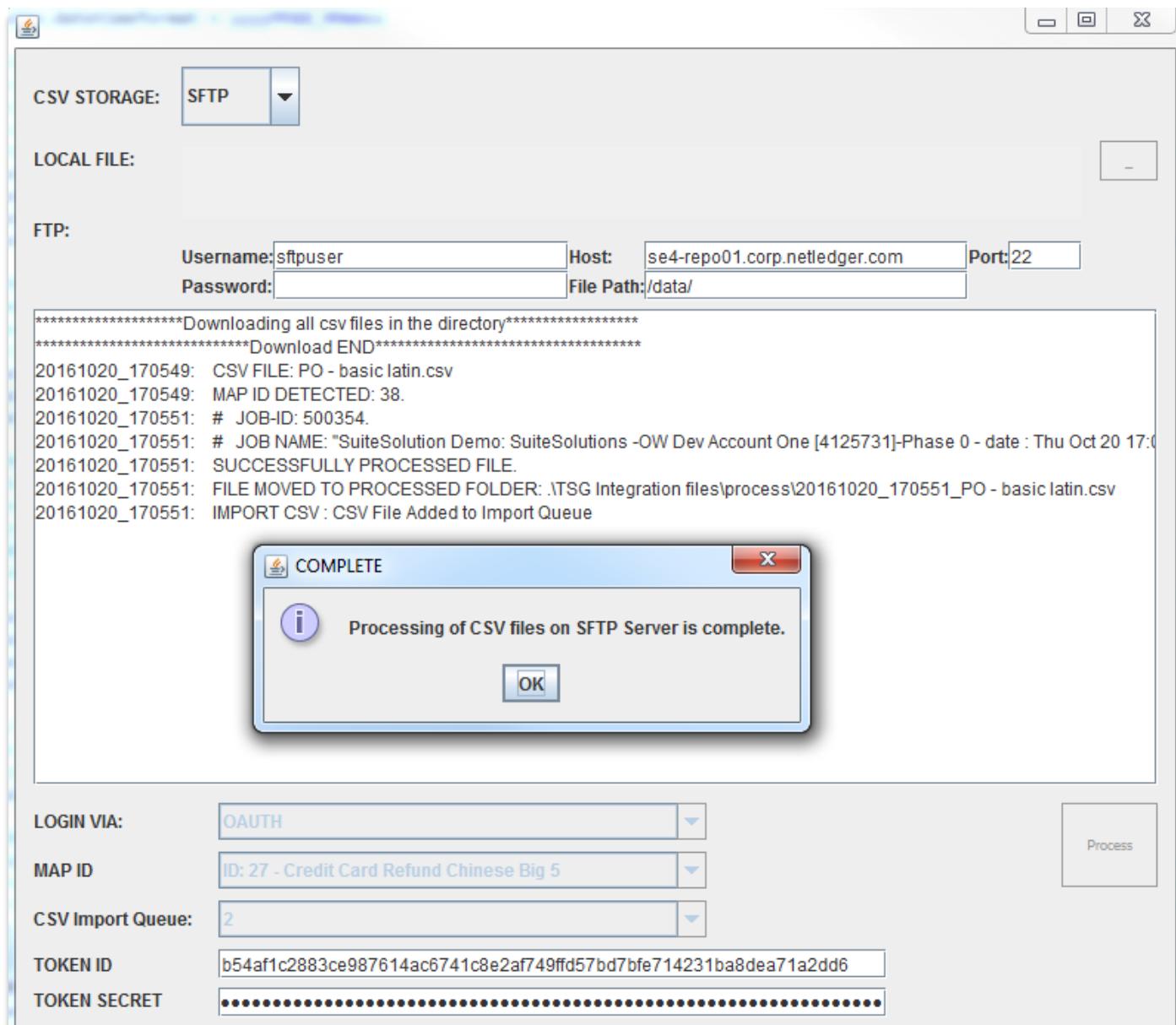


Figure 14 - Integrator Broker User Interface - SFTP

### 3.12.4 Using the Integrator Console Interface

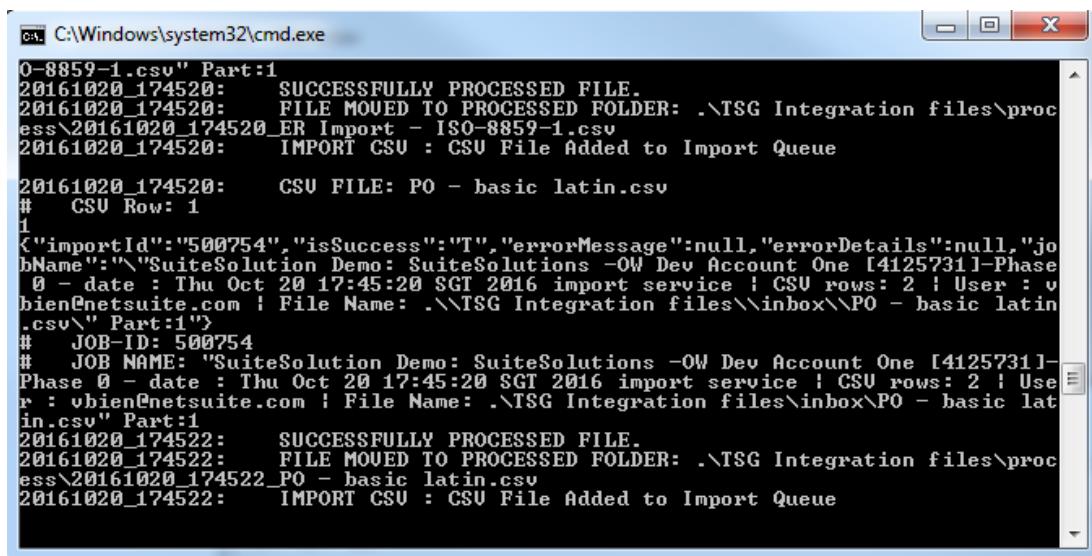
The console interface functionality of the solution may be used for the following instances:

- The Operating System does not have a user interface and can only be accessed through the console (e.g. Linux server, Unix server, Solaris).
- To automate the process – using the Integrator via Windows Scheduler (see [Using the Integrator via Windows Scheduler](#)).
- To interface with FTP or network path without the user interface.
- To integrate with other applications (i.e. the CSV Integrator shall be triggered after running an application).

#### Pre-requisite:

- CSV file for import.
- [Broker Configuration](#).
- [Saved CSV Import](#).
- [Integration Mapping Record](#).

1. Locate and open the **Run Via Console.bat** file downloaded from the [File Cabinet](#).
2. The console will automatically process all of the files inside the inbox folder.
3. The import status of the CSV will display on the console.
4. A new job will be added to the CSV Import Job Status in NetSuite.
5. The processed files will be moved to the 'Processed' folder.



```

C:\Windows\system32\cmd.exe
0-8859-1.csv" Part:1
20161020_174520:    SUCCESSFULLY PROCESSED FILE.
20161020_174520:    FILE MOVED TO PROCESSED FOLDER: .\TSG Integration files\processed\20161020_174520_ER Import - ISO-8859-1.csv
20161020_174520:    IMPORT CSV : CSV File Added to Import Queue

20161020_174520:    CSU FILE: PO - basic latin.csv
#   CSU Row: 1
1
{"importId":"500754","isSuccess":"T","errorMessage":null,"errorDetails":null,"jobName":"\"SuiteSolution Demo: SuiteSolutions -OW Dev Account One [4125731]-Phase 0 - date : Thu Oct 20 17:45:20 SGT 2016 import service | CSV rows: 2 | User : vbiens@netsuite.com | File Name: .\TSG Integration files\inbox\PO - basic latin.csv" Part:1"}
#   JOB-ID: 500754
#   JOB NAME: "SuiteSolution Demo: SuiteSolutions -OW Dev Account One [4125731]-Phase 0 - date : Thu Oct 20 17:45:20 SGT 2016 import service | CSV rows: 2 | User : vbiens@netsuite.com | File Name: .\TSG Integration files\inbox\PO - basic latin.csv" Part:1
20161020_174522:    SUCCESSFULLY PROCESSED FILE.
20161020_174522:    FILE MOVED TO PROCESSED FOLDER: .\TSG Integration files\processed\20161020_174522_PO - basic latin.csv
20161020_174522:    IMPORT CSV : CSV File Added to Import Queue

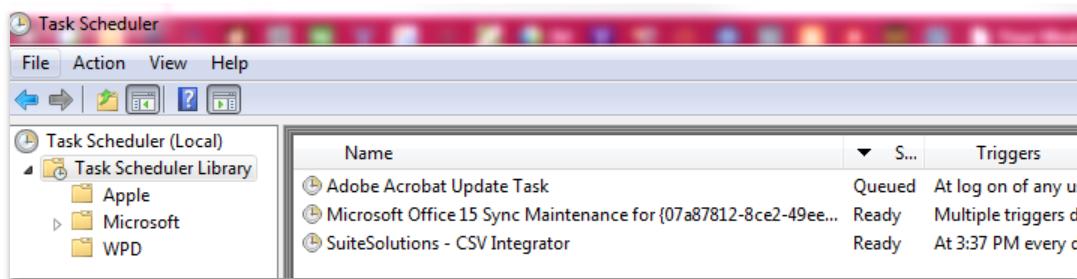
```

Figure 15 - Integrator Console Interface

### 3.12.5 Using the Integrator via Windows Scheduler

To run the integrator via Windows Scheduler, a task must be created using the Windows Task Scheduler.

1. Select Create Task from the Actions panel.
2. On the General tab, enter the name of the task.
3. On the Trigger tab, click the 'New' button to specify the schedule settings of the task.
4. On the Actions tab, click the 'New' button to set the program or script to run. Browse for the Run Via Console batch file associated with the [Broker Files](#).
5. Set the other preferences on the Conditions and Settings tab.
6. Click OK to save.
7. A new task will be added to the Task Scheduler Library.
8. The task will run on the schedule defined.



**Figure 16 - Created Windows Task for CSV Integrator**

### 3.13 CSV File with More Than 24 999 Rows

If a file has more than 24,999 rows, upon parsing of the data by the RESTlet; it will divide the rows into chunks of 25K rows and will invoke the CSV process. The mapping to use will be defined as a parameter on the deployment.

### 3.14 Upload a JE CSV file with Debit and Credit rows NOT balanced

If a JE's Debit and Credit line do not balance on CSV process, the import job will fail and the user will be notified of the error.

### 3.15 Upload a JE CSV file with Debit and Credit rows balanced

If a JE's Debit and Credit line do balance on CSV process, the import job will succeed and the user will be notified of the status.

### 3.16 Error in Process

In case of errors, the file will be moved to the error folder and an error log will be added.

### 3.17 RUN VIA UI

Clicking the RUN VIA UI file will display both the UI interface and the console that runs the JAR file.

## Appendix

### 3.18 Bundle Components

Listed on this section are the components of the solution such as the custom record, custom fields, saved searches, scripts, etc., that were not mentioned in [3 Setup and Configuration](#).

**Name:** SuiteSolutions – CSV Integrator

**Version:** 2.1.0

#### 3.18.1 Custom Lists/Records

- Records

##### Integration Mapping Record

Name	Name of the mapping record.
Mapping ID	From generated Script ID after <a href="#">Saved CSV Import</a> creation.
CSV Header Format	Actual header names of the uploaded CSV file.
Primary ID (has Debit/Credit)	

#### 3.18.2 CSV Imports

- Saved CSV Imports

- CSV Integrator : Inventory Item
- CSV Integrator : Expense Report
- CSV Integrator : Journal Entry

#### 3.18.3 Scripts

- RESTlet
  - Base CSV Integrator RS

#### 3.18.4 Roles and Permission

The access for configuring and processing *SuiteSolutions – CSV Integrator* should be aligned with the standard NetSuite Security and Permissions. By default, the CSV Integrator can be run by Administrator role. In the event a custom role is preferred over the Administrator role, the following requirements shall be set up (at the minimum):

1. Under *Role > Permissions* subtab > *Setup*, add **Set Up Company** and set the level to **View** or **Full**.
2. Under *Role > Permissions* subtab > *Setup*, add **Import CSV File** and set the level to **View** or **Full**.

Permissions		Forms	Searches	Users	Preferences	Dashboard	Translation	History
Transactions • Reports • Lists • Setup • Custom Record •								
PERMISSION *							LEVEL	
Accounting Lists							Edit	
Accounting Management							Edit	
Deleted Records							Full	
Mobile Device Access							Full	
Other Lists							Edit	
<b>Set Up Company</b>							<b>Full</b>	
Web Services							Full	

**Figure 17 - Created Custom Role**

3. Under *Customization > Lists, Records, & Fields > Record Types > CSV Integration Mapping*, set the **Access Type** to **Use Permission List**.

**CSV Integration Mapping**

Save | Cancel | Reset | Change ID | Actions ▾

NAME *	CSV Integration Mapping	SHOW OWNER <input type="checkbox"/> ON RECORD <input type="checkbox"/> ON LIST <input type="checkbox"/> ALLO
ID	customrecord_stg_integrator_mapping	ACCESS TYPE
INTERNAL ID	77	Use Permission List Require Custom Record Entries Permission Use Permission List No Permission Required
OWNER	Den D Geronimo	<input type="checkbox"/> SHOW NOTES <input type="checkbox"/> ENABLE MAIL MERGE

**Figure 18 - CSV Integration Mapping Access Type**

4. Under *Customization > Scripting > Scripts > Base CSV Integrator RS*, select the deployment record under the **Deployments** subtab. Under the Audience subtab, select the created custom role as one of the **Roles**, and select the employee to be used by the CSV Integrator as one of the **Employees**.

**Script Deployment**

Edit | Back | Actions ▾

SCRIPT Base CSV Integrator RS	STATUS Released
TITLE NSTS RESTLet Base Integrator	LOG LEVEL Debug
ID customdeploy_tsg_integrator_reslet	URL <a href="/app/site/hosting/restlet.nl?script=163&amp;deploy=1">/app/site/hosting/restlet.nl?script=163&amp;deploy=1</a>
<input checked="" type="checkbox"/> DEPLOYED	EXTERNAL URL <a href="https://rest.na1.netsuite.com/app/site/hosting/restlet.nl?script=163&amp;deploy=1">https://rest.na1.netsuite.com/app/site/hosting/restlet.nl?script=163&amp;deploy=1</a>

**Audience • Execution Log • System Notes**

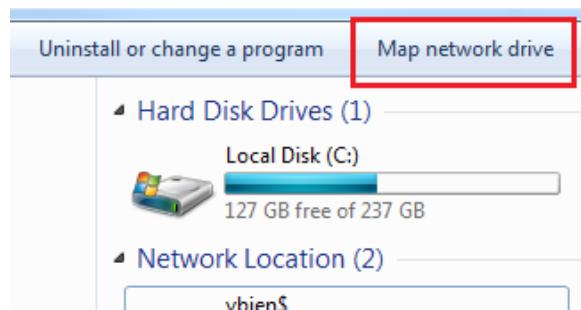
ROLES Accountant   CSV Integrator	SUBSIDIARIES <input type="checkbox"/> ALL EMPLOYEES
<input type="checkbox"/> ALL ROLES	PARTNERS <input type="checkbox"/> ALL PARTNERS
DEPARTMENTS	GROUPS <input type="checkbox"/> EMPLOYEES
	<b>Victor Bien</b>

**Figure 19 - Base CSV Integrator RESTlet Script Deployment**

### 3.18.5 Windows Map Network Drive

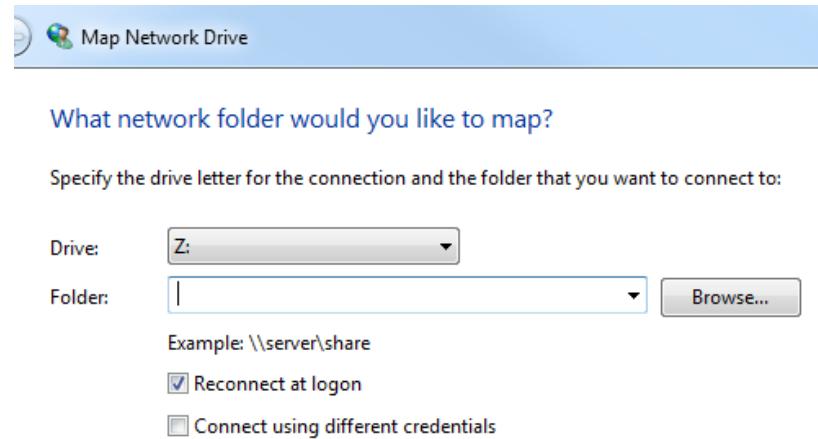
Files to be uploaded are sometimes contained in a remote location. In Windows, mapping network drive is the safest way to locate files in a remote location. This is to ensure successful connectivity to the remote location since some of the remote locations require credentials.

1. Under My Computer, click the **Map Network Drive**.



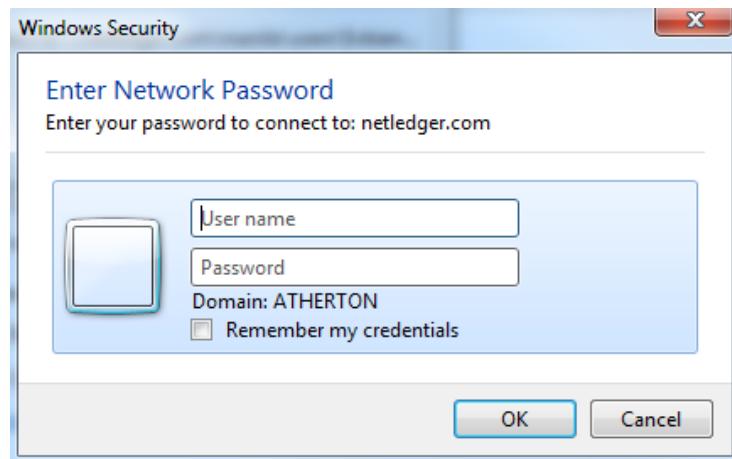
**Figure 20 – My Computer/Map Network Drive**

2. Select the Drive and the Remote Location/Folder accordingly.



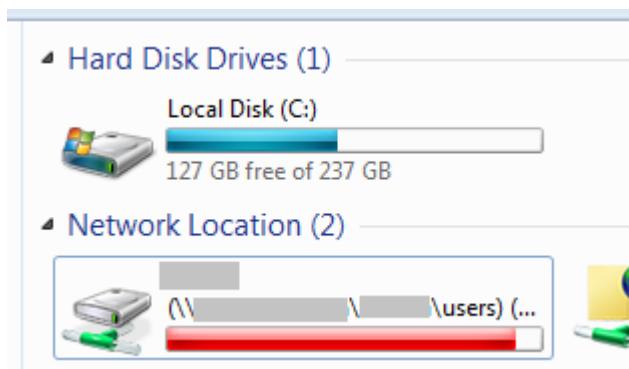
**Figure 21 – Map Network Drive page**

3. Enter the Username and Password.



**Figure 22 – Map Network Drive Windows Security Page**

4. A Network Location shall be created accordingly.



**Figure 23 – Network Locations**



The created Network Location can now be located or used either via User Interface or Console. For User Interface, select the created drive. For the Console, indicate the created drive in the properties file.

### 3.19 Limitations

The following items are not supported or limitations of the solution.

1. Integrating/importing of records not supported through NetSuite's CSV import functionality.
2. Broker cannot validate sanctity of the data in the CSV File.
3. There is a limit of 10 MB per string used as RESTlet input or output (NS Limitation)
4. Does not support multi-file import.
5. UTF8 with BOM encoding is not supported.
6. Partial Fulfillments via CSV Import are **not** allowed since we can only import sales order numbers and not the line items. (Please see [SuiteAnswers ID: 38467, Rules on Importing Item Fulfillments via CSV](#)).
7. Token-based Authentication shall only work for Basic Latin Character Set for both UTF-8 and ISO-8859-1.
8. Korean character encoding format is not supported by the Integration Broker.
9. Only one Import Mapping is supported for each record type. If there are two records with similar mapping e.g. Transactions, the mapping needs to be labelled to differentiate one transaction from another.
10. Broker cannot update whether the Script should trigger upon import. It needs to be specified in the saved CSV import mapping.

### 3.20 Solution Design

#### 3.20.1 RESTlet SuiteScript Automation to Import Files

- This SuiteScript automation will be invoke via a POST to a published RESTlet endpoint

- Accepting a JSON request originating from the Integration Broker, the RESTlet will determine the CSV import map id to be used for the import
- Next, the script will pass a raw string (CSV – retrieved from the JSON Request) as the Primary File
- NetSuite will queue the file for import and pass back a response containing the CSV Import Job ID

### 3.20.2 NetSuite Properties Configuration File

```

# Web Service (WSDL) url
# [netsuite.wsdl.url] is use to Specify the netsuite Webservice URL
# for sandbox account use
https://webservices.sandbox.netsuite.com/services/NetSuitePort_2015_2
# for production account use
netsuite.ws.url =
https://webservices.netsuite.com/services/NetSuitePort_2015_1

# LOGIN PREFERENCE
# {login} OAUTH for Token Based Authentication and NLAUTH for Standard Login
netsuite.login = NLAUTH

# DATE PREFERENCE
# {dateformat} Set the date format to be used for log file name
# {datetimeformat} Set the date time format to be used for logging
netsuite.dateformat = yyyy-MM-dd
netsuite.datetimeformat = yyyyMMdd_HHmmss

# ACCOUNT DETAILS (ALL REQUIRED)
# {account} netsuite company account
# {suiteCloudAcct} Set to true ONLY IF account is SuiteCloud Licensed Account
# to support multiQueueing of csv imports
# {csvImportQueue} Set this if account is suitecloud licensed account
# ({suiteCloudAcct} is set to true), this will be the queue no to be used on csv
# import. Select from queue no 2-5 only.
# Integration Restlet URL can be seen on Customization -> Script Deployments -
# > Base CSV Integrator RS -> URL (relative path)
netsuite.account = 4125731
netsuite.suiteCloudAcct = false
netsuite.csvImportQueue = 3
netsuite.tsg.Integration.restlet.url =
/app/site/hosting/restlet.nl?script=customscript_tsg_integrator_reslet&deploy=
customdeploy_tsg_integrator_reslet

# (FOR NLAUTH LOGIN PREFERENCE ONLY) LOGIN CREDENTIALS
# {email} email address used to log in on netsuite account
# {password} password used to log in on netsuite account
# {role} role id to log in
netsuite.email =
netsuite.password =
netsuite.role = 3

# (FOR OAUTH LOGIN PREFERENCE ONLY) LOGIN CREDENTIALS

```

```

# Note that failed login attempts must be less than 6
# {consumerKey} consumer id generated for the application
# {consumerSecret} consumer secret generated for the application
# {tokenId} token id generated for the user/role
# {tokenSecret} token secret generated for the user/role
# See suiteanswers for getting started on token based authentication
https://netsuite.custhelp.com/app/answers/detail/a\_id/41898
netsuite.consumerKey =
netsuite.consumerSecret =
netsuite.tokenId =
netsuite.tokenSecret =

# CSV FILES STORAGE
# Add the CSV Storage details if broker is run via console.
# {csvStorage} FTP if the files are to be read on ftp server, SFTP if the
files are to be read on sftp server, LOCAL if files are in local directory
netsuite.csvStorage = SFTP

# (FOR LOCAL AND FTP) CSV FILES STORAGE
# {inbox} File inbox directory, If FTP is enabled, CSV files will be
downloaded here. If LOCAL is enabled, CSV files are stored here.
# {processed} CSV files successfully processed on the inbox directory will be
deleted on the inbox and moved on this processed directory
# {error} CSV files not successfully processed on the inbox directory will be
deleted on the inbox and moved on this error directory
files.path.inbox = .\\TSG Integration files\\inbox\\
files.path.processed = .\\TSG Integration files\\process\\
files.path.error = .\\TSG Integration files\\error\\

# (FOR FTP PREFERENCE ONLY) CSV FILES STORAGE
# {username} ftp/sftp username
# {password} ftp/sftp username
# {host} ftp/sftp host is required
# {path} ftp/sftp path: location of csv files,must be a directory
# {port} ftp/sftp port
netsuite.ftp.username      =
netsuite.ftp.password      =
netsuite.ftp.host          =
netsuite.ftp.path           =
netsuite.ftp.port           =

# (FOR SFTP PREFERENCE ONLY) CSV FILES STORAGE
# {knownhostpath} Set the path that contains the file for known host. This is
used to verify the server's legitimacy that is provided by SSH.
# {privatekeypath} Set the path that contains the file for private key. This
is to be paired with the server's public key.
# {knownhostpath} Set the value for the passphrase. This is used for the
encryption of the private key
netsuite.sftp.knownhostpath      =
netsuite.sftp.privatekeypath     =
netsuite.sftp.passphrase        =

```

```

# (FOR FTP PREFERENCE ONLY) CSV FILES STORAGE
# {removeProcessedFile} Set to true if files processed will be moved on
processed or error folder on the server and deleted on the current folder
# {errorDir} Directory on the server for files that have an error upon process
# {processedDir} Directory on the server for files that are successfully sent
to CSV Import
netsuite.ftp.removeProcessedFile      = false
netsuite.ftp.errorDir                = /error
netsuite.ftp.processedDir           = /processed

# JobName definition
# {datetime} concat the Netsuite date time
# {csvrows} concat the csv row count
# {compname} concat the company name
# {fileName} concat the file name uploaded
# {user} concat the user email
netsuite.tsg.JobName = "SuiteSolution Demo: {compname} - date : {datetime}"
import service | CSV rows: {csvrows} | User : {user} | File Name: {fileName}"

# User Log files
baselogdir = .\\logs\\
execution.log = executionLog.txt
success.log = successLog.txt
failure.log = failureLog.txt

# Character Encoding
# Character Encoding must be the same for CSV Import configuration
# US-ASCII          :    Seven-bit ASCII, a.k.a. ISO646-US, a.k.a. the Basic
Latin block of the Unicode character set
# ISO-8859-1        :    ISO Latin Alphabet No. 1, a.k.a. ISO-LATIN-1- Western
(Windows 1252) encoding
# UTF-8             :    Eight-bit UCS Transformation Format (Unicode) - UTF-8
encoding
# UTF-16BE          :    Sixteen-bit UCS Transformation Format, big-endian
byte order
# UTF-16LE          :    Sixteen-bit UCS Transformation Format, little-
 endian byte order
# UTF-16             :    Sixteen-bit UCS Transformation Format, byte order
identified by an optional byte-order mark
# windows-1252       :    Windows Latin-1
# GB18030           :    Simplified Chinese, PRC standard
# Shift_JIS          :    Shift-JIS, Japanese
# MacRoman           :    MacRoman
files.Charset = ISO-8859-1

```

**Note:** To connect to the sandbox account, change the **netsuite.ws.url** to the appropriate NetSuite sandbox WebService url.