

Netsuite Import Integration

Treasure Data allows you to import your data from Netsuite.

For sample workflows of importing data from Netsuite, view [Treasure Boxes](#).

Continue to the following topics:

- [Prerequisites](#)
- [Use TD Console](#)
- [Install 'td' Command v0.11.9 or Later](#)
- [Create Seed Config File \(seed.yml\)](#)
- [Guess Fields \(Generate load.yml\)](#)
- [Execute Load Job](#)
- [Scheduled Execution](#)
- [Incremental Load](#)
- [Appendix](#)

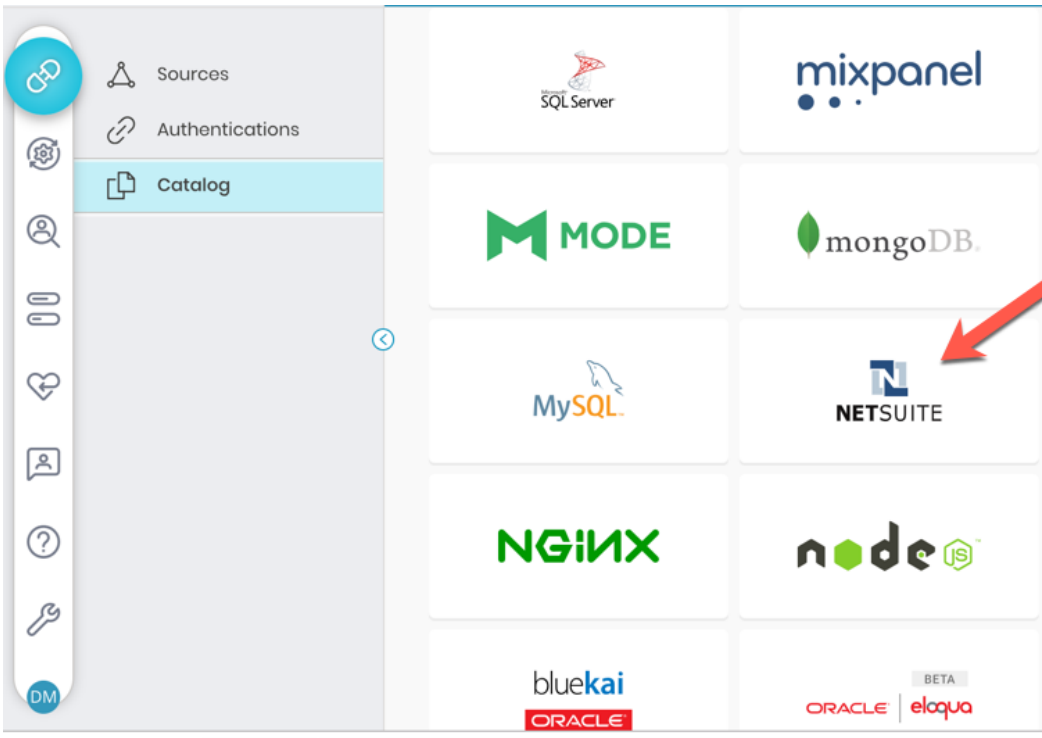
Prerequisites

- Basic knowledge of Treasure Data
- NetSuite account

Use TD Console

Create a New Connection

Go to Integrations Hub > Catalog and search. Select NetSuite.



Complete the NetSuite login information. Click **Continue**.

New Authentication

NetSuite

1 Credentials > 2 Details

Email

Password

Account ID

Role ID

Learn more CONTINUE

Provide a connection name.

New Authentication

NetSuite

✔ Credentials > 2 Details

Name

Share with others

BACK NEW SOURCE DONE

Click **Done**.

Create a New Source

After creating the connection, you are automatically taken to the Authentications page. Look for the connection you created and click **New Source**. Specify the source location.

New Source ✕

Using testingdk

1 Fetch from >
 2 Source Preview >
 3 Transfer to >
 4 Schedule >
 5 Details

Source

e.g. Customer, Invoice, etc. That should accept search operation.

Incremental?
When run repeatedly, attempt to only import new data since the last import.

NEXT

Next, you see a Preview of your data. To make changes to the data, such as skipping on errors or rate limits, click **Advanced Settings** otherwise, click **Nex**

Select the database and table where you want to transfer the data.

In the **Schedule** tab, you can specify a one-time transfer, or you can schedule an automated recurring transfer. If you select **Once now**, click **Start Transfer**. If you select **Repeat...** specify your schedule options, then click **Schedule Transfer**.

After your transfer has run, you can see the results of your transfer in **Data Workbench > Databases**.

You are ready to start analyzing your data.

Install 'td' Command v0.11.9 or Later

Install the newest [TD Toolbelt](#).

```
$ td --version
0.11.10
```

Create Seed Config File (seed.yml)

First, prepare seed.yml as shown in the following example, with your email, password, account, and role. You'll use "replace" mode:

```
in:
  type: netsuite
  target: Customer
  email: <YOUR_EMAIL_ADDRESS>
  password: <YOUR_PASSWORD>
  account: <YOUR_ACCOUNT_ID>
  role: <YOUR_ROLE_ID>
  incremental: <INCREMENTAL_OPTION> #(optional, default: true)
  from_datetime: <DATE TO LOAD DATA> #(optional, Date value i.e 2017-07-24 15:43:11 +0700 to specify the date
that data)
  sandbox: false # set "sandbox: true" if you want to get sandbox account data
out:
  mode: replace
```

The account and role (which is optional) can be found on Setup → Integration → Web Service Preference from the navigation bar.

The target specifies what type of object you want to dump from NetSuite. Types are listed on the [NetSuite official document](#) but some types are not supported. Target types tested by Treasure Data: Customer, Invoice, CustomerPayment, and CustomerRefund.

Specifying incremental results runs. The task attempts to import only data that is new, since the last import.

For more details on available out modes, see the Appendix.

Guess Fields (Generate load.yml)

Use `connector:guess`. This command automatically reads the target file, and intelligently guesses the file format.

```
$ td connector:guess seed.yml -o load.yml
```

When you open `load.yml`, you see guessed file format definitions including, in some cases, file formats, encodings, column names, and types.

```

in:
  type: netsuite
  target: Customer
  email: <YOUR_EMAIL_ADDRESS>
  password: <YOUR_PASSWORD>
  account: <YOUR_ACCOUNT_ID>
  role: <YOUR_ROLE_ID>
  incremental: true
  sandbox: false
  columns:
    - {name: access_role_external_id, type: string}
    - {name: access_role_internal_id, type: long}
    - {name: access_role_name, type: string}
    - {name: access_role_type, type: string}
    - {name: comments, type: string}
    - {name: company_name, type: string}
    - {name: credit_hold_override, type: string}
    - {name: currency_external_id, type: string}
    - {name: currency_internal_id, type: long}
    - {name: currency_name, type: string}
    - {name: currency_type, type: string}
    - {name: custom_fields, type: json}
    - {name: date_created, type: timestamp, format: "%Y-%m-%d %H:%M:%S %z"}
    - {name: days_overdue, type: long}
    - {name: default_address, type: string}
    - {name: email, type: string}
    - {name: email_preference, type: string}
    - {name: email_transactions, type: boolean}
    - {name: entity_id, type: string}
    - {name: entity_status_external_id, type: string}
    - {name: entity_status_internal_id, type: long}
    - {name: entity_status_name, type: string}
    - {name: entity_status_type, type: string}
    - {name: fax, type: string}
    - {name: fax_transactions, type: boolean}
    - {name: give_access, type: boolean}
    - {name: is_budget_approved, type: boolean}
    - {name: is_inactive, type: boolean}
    - {name: is_person, type: boolean}
    - {name: language, type: string}
    - {name: last_modified_date, type: timestamp, format: "%Y-%m-%d %H:%M:%S %z"}
    - {name: phone, type: string}
    - {name: print_transactions, type: boolean}
    - {name: sales_rep_external_id, type: string}
    - {name: sales_rep_internal_id, type: long}
    - {name: sales_rep_name, type: string}
    - {name: sales_rep_type, type: string}
    - {name: stage, type: string}
    - {name: subsidiary_external_id, type: string}
    - {name: subsidiary_internal_id, type: long}
    - {name: subsidiary_name, type: string}
    - {name: subsidiary_type, type: string}
    - {name: taxable, type: boolean}
    - {name: terms_external_id, type: string}
    - {name: terms_internal_id, type: long}
    - {name: terms_name, type: string}
    - {name: terms_type, type: string}
    - {name: unbilled_orders, type: string}
    - {name: url, type: string}
  filters: []
  out: {more: replace}
  exec: {}

```

You can preview how the system parses the file by using the preview command.

```
$ td connector:preview load.yml
```

If the system detects your column name or type unexpectedly, modify load.yml directly and preview again.

```
Currently, the Data Connector supports parsing of "boolean", "long", "double", "string", "timestamp", and "json" types.
```

Execute Load Job

Finally, submit the load job. It might take a couple of hours depending on the data size. Users need to specify the database and table where their data is stored.

```
$ td connector:issue load.yml --database td_sample_db --table td_sample_table
```

The above command assumes you have already created *database(td_sample_db)* and *table(td_sample_table)*. If the database or the table do not exist in Treasure Data, then this command will not succeed, so create the database and table [manually](#) or use `--auto-create-table` option with `td connector:issue` command to auto create the database and table:

```
$ td connector:issue load.yml --database td_sample_db --table td_sample_table --time-column created_at --auto-create-table
```

You can assign Time Format column to the "Partitioning Key" by "--time-column" option.

Scheduled Execution

You can schedule periodic Data Connector execution for periodic NetSuite import. We take great care in distributing and operating our scheduler in order to achieve high availability. By using this feature, you no longer need a cron daemon on your local data center.

Incremental Load

You can incrementally load data by setting `incremental`: to true and specify the `from_date` as shown in the following example:

```
in:
  type: netsuite
  target: Customer
  email: <YOUR_EMAIL_ADDRESS>
  password: <YOUR_PASSWORD>
  account: <YOUR_ACCOUNT_ID>
  role: <YOUR_ROLE_ID>
  incremental: <INCREMENTAL_OPTION> #(optional, default: true)
  from_datetime: <DATE TO LOAD DATA> #(optional, Date value i.e 2017-07-24 15:43:11 +0700 to specify the date that data)
  sandbox: false # set "sandbox: true" if you want to get sandbox account data
out:
  mode: replace
```

If you omit the `from_datetime`, then all available records (data) are fetched. Only timestamp, and timestampz(timestamp with timezone) are supported.

Create the Schedule

You can create a new schedule by using the `td connector:create` command. The name of the schedule, cron-style schedule, the database and table where their data will be stored, and the Data Connector configuration file are required.

```
$ td connector:create \
  daily_netsuite_import \
  "10 0 * * *" \
  td_sample_db \
  td_sample_table \
  load.yml
```

The ``cron`` parameter also accepts three special options: ``@hourly``, ``@daily`` and ``@monthly``.

By default, schedule is setup in UTC timezone. You can set the schedule in a timezone using `-t` or `--timezone` option. The `--timezone` option supports only extended timezone formats like 'Asia/Tokyo', 'America/Los_Angeles' etc. Timezone abbreviations like PST, CST are *not* supported and might lead to unexpected schedules.

List the Schedules

You can see the list of currently scheduled entries by `td connector:list`.

```
$ td connector:list
+-----+-----+-----+-----+-----+-----+
+-----+
| Name          | Cron          | Timezone | Delay | Database   | Table          |
Config         |               |          |       |            |                |
+-----+-----+-----+-----+-----+-----+
| daily_netsuite_import | 10 0 * * *   | UTC      | 0     | td_sample_db | td_sample_table | {"type"=>"netsui",
... } |
+-----+-----+-----+-----+-----+-----+
+-----+
```

Show the Setting and History of Schedules

`td connector:show` shows the execution setting of a schedule entry.

```
% td connector:show daily_netsuite_import
Name      : daily_netsuite_import
Cron      : 10 0 * * *
Timezone  : UTC
Delay     : 0
Database  : td_sample_db
Table     : td_sample_table
Config
---
// Displayed load.yml configuration.
```

`td connector:history` shows the execution history of a schedule entry. To investigate the results of each individual execution, use `td job <jobid>`.

```
% td connector:history daily_netsuite_import
+-----+-----+-----+-----+-----+-----+-----+-----+
+
| JobID | Status | Records | Database   | Table          | Priority | Started              | Duration |
+-----+-----+-----+-----+-----+-----+-----+-----+
+
| 578066 | success | 10000   | td_sample_db | td_sample_table | 0        | 2015-04-18 00:10:05 +0000 | 160      |
| 577968 | success | 10000   | td_sample_db | td_sample_table | 0        | 2015-04-17 00:10:07 +0000 | 161      |
| 577914 | success | 10000   | td_sample_db | td_sample_table | 0        | 2015-04-16 00:10:03 +0000 | 152      |
| 577872 | success | 10000   | td_sample_db | td_sample_table | 0        | 2015-04-15 00:10:04 +0000 | 163      |
| 577810 | success | 10000   | td_sample_db | td_sample_table | 0        | 2015-04-14 00:10:04 +0000 | 164      |
| 577766 | success | 10000   | td_sample_db | td_sample_table | 0        | 2015-04-13 00:10:04 +0000 | 155      |
| 577710 | success | 10000   | td_sample_db | td_sample_table | 0        | 2015-04-12 00:10:05 +0000 | 156      |
| 577610 | success | 10000   | td_sample_db | td_sample_table | 0        | 2015-04-11 00:10:04 +0000 | 157      |
+-----+-----+-----+-----+-----+-----+-----+-----+
+
8 rows in set
```

Delete the Schedule

td connector:delete removes the schedule.

```
$ td connector:delete daily_netsuite_import
```

Appendix

Modes for Out Plugin

You can specify the file import mode in the out section of seed.yml.

Append (Default)

Append is the default mode and records are appended to the target table.

```
in:
  ...
out:
  mode: append
```

Replace (In TD 0.11.10 and Later)

The replace mode replaces data in the target table. Any manual schema changes made to the target table remain intact with this mode.

```
in:
  ...
out:
  mode: replace
```

Further Information

The following list provides details of all available options:

```
target: Target object e.g. Customer, Invoice, etc. That should accept search operation. (string, required)
email: Email address. (string, required)
password: Password. (string, required)
account: Account ID. (string, required)
role: Role ID. (string, default: nil)
sandbox: Use sandbox environment if true. (bool, default: false)
from_datetime: Fetch data after this time. (string, default: nil)
```