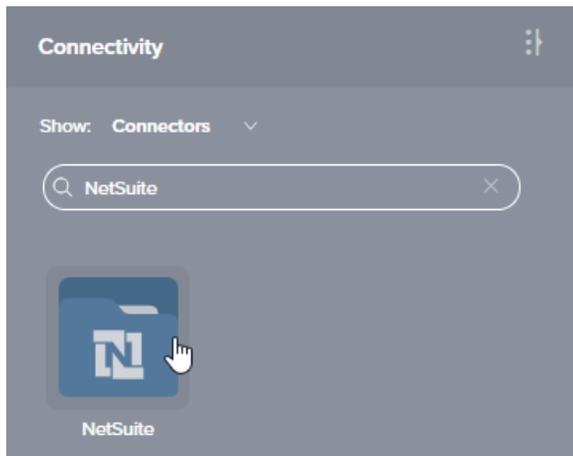# NetSuite Connection

Introduction

A NetSuite connection is configured using the NetSuite connector to establish access with a specific NetSuite instance. Once a connection is established, you can configure one or more NetSuite activities associated with that connection to use as a source or target within an operation. In addition, the NetSuite connection can be used within scripts.
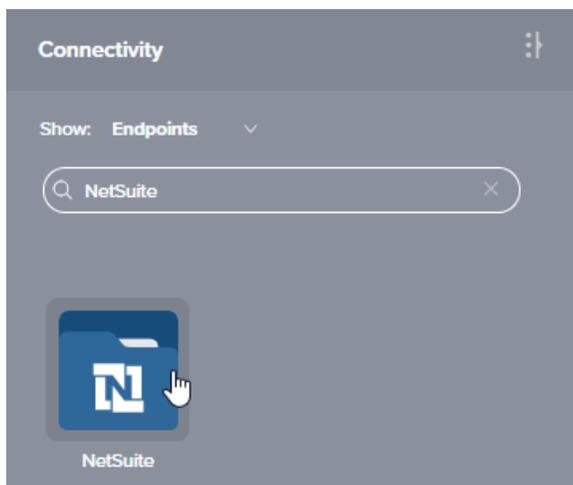
Creating or Editing a NetSuite Connection

From the design canvas, open the **Connectivity** tab of the design component palette:

To configure a new NetSuite connection, within the **Connectors** filter, click the NetSuite connector block:

To configure an existing NetSuite connection, within the **Endpoints** filter, double-click the NetSuite connection block:

This will open a configuration screen for the NetSuite connection, covered next.

Configuring a NetSuite Connection

Configuration of a NetSuite connection includes these fields:

Last updated: Dec 12, 2019

Configure NetSuite

ENDPOINT NAME*

NetSuite

ACCOUNT*

1234567

WSDL DOWNLOAD URL*

https://webservices.netsuite.com/wsdl/v2016_2_0/netsuite.wsdl

CONSUMER KEY*

808b8c3c753d5dbc474e92e5c7bffce29136b67384b51c7cfe87497a...

CONSUMER SECRET*

••

TOKEN KEY*

908051347955e1b3281dd1b9c1b12971c3150f6cb047e4d1ecf78c86a...

TOKEN SECRET*

••

SIGNATURE ALGORITHM

HmacSHA1

CALL TIME OUT

Test

Save Changes   Discard Changes   Delete

✓ **TIP:** Fields with a variable icon ⊻ support using global variables, project variables, and Jitterbit variables. Begin typing an open square bracket [into the field or click the variable icon to display existing variables to choose from.

- **Endpoint Name:** Enter a name to use to identify the NetSuite connection. The name must be unique for each NetSuite connection and must not contain forward slashes (/) or colons (:). As a specific connection and its activities are referred to as an endpoint, this name is also used to identify the NetSuite endpoint.
- **Account:** Enter the NetSuite account ID associated with the NetSuite account you want to use. Accounts associated with a NetSuite sandbox environment may be indicated with a suffix such as _SB1, _SB2, etc.
- **WSDL Download URL:** Enter the URL of the NetSuite WSDL used in the NetSuite instance. Jitterbit supports these WSDL versions:

  - https://webservices.netsuite.com/wsdl/v2019_2_0/netsuite.wsdl (supported with all agent versions)
  - https://webservices.netsuite.com/wsdl/v2019_1_0/netsuite.wsdl (supported with agents version 10.4 or higher)
  - https://webservices.netsuite.com/wsdl/v2018_2_0/netsuite.wsdl (WSDL versions 2018.2 and lower are supported with agents version 9.9 or higher)
  - https://webservices.netsuite.com/wsdl/v2018_1_0/netsuite.wsdl
  - https://webservices.netsuite.com/wsdl/v2017_2_0/netsuite.wsdl
  - https://webservices.netsuite.com/wsdl/v2017_1_0/netsuite.wsdl
  - https://webservices.netsuite.com/wsdl/v2016_2_0/netsuite.wsdl (default)
  - https://webservices.netsuite.com/wsdl/v2016_1_0/netsuite.wsdl
  - https://webservices.netsuite.com/wsdl/v2015_2_0/netsuite.wsdl

> ⓘ **NOTE:** As of January 2018, NetSuite uses the same URL for both their production and sandbox domain. If your NetSuite sandbox has not been refreshed since these changes were implemented, you may need to use a sandbox-specific WSDL URL. For more information, see NetSuite's documentation About Sandbox Accounts on the NetSuite Domain (login to NetSuite required).

> ⚠ **CAUTION:** If you receive an error regarding the data center while testing the connection, you may need to use a different WSDL URL. For more information, see Data Center Error later on this page.

- **Consumer Key** and **Consumer Secret:** Enter the NetSuite Consumer Key and Consumer Secret values obtained from NetSuite. For instructions on obtaining these values, see Gathering Values for Jitterbit Harmony under NetSuite.
- **Token Key** and **Token Secret:** Enter the NetSuite Token ID and Token Secret values obtained from NetSuite. For instructions on obtaining these values, see Gathering Values for Jitterbit Harmony under NetSuite.

> ⚠ **CAUTION:** If you are using a NetSuite sandbox account, each time the sandbox is refreshed, you will need to create new tokens.

- **Signature Algorithm:** The Consumer Secret and Token Secret are used to sign the request using either of these supported signature algorithms: **HmacSHA1** or **HmacSHA256**. You may use either one to affect how the payload is encrypted.
- **Call Time Out:** Optionally enter the call timeout value in seconds if you want to the timeout value to be less than the agent setting.

> ⓘ **NOTE:** The default agent setting for timeout of NetSuite calls is 300 seconds.

- **Test:** Click this button to verify the connection to the Salesforce instance with the provided credentials.
- **Save Changes:** Click this button to save and close the connection configuration.
- **Discard Changes:** After making changes to a new or existing configuration, click **Discard Changes** to close the configuration without saving. A message will ask you to confirm that you want to discard changes.
- **Delete:** After opening an existing connection configuration, click **Delete** to permanently delete the connection from the project and close the configuration (see Component Dependencies, Deletion, and Removal).

Next Steps

After configuring a NetSuite connection, you can configure one or more NetSuite activities associated with that connection to use as a source or target within an operation, or you can use the NetSuite connection within a script as described below.

Configuring Activities

NetSuite activities interact with the NetSuite connection to act as sources (providing data within an operation) or targets (receiving data within an operation). For more information, see these activities:

- **Search:** Retrieves existing records from a NetSuite connection and is used as a source in an operation.
- **Create:** Creates new records in a NetSuite connection and is used as a target in an operation.
- **Update:** Updates existing records in a NetSuite connection and is used as a target in an operation.
- **Get List:** Retrieves a list of existing records based on ID from a NetSuite connection and is used as a target in an operation.
- **Upsert:** Both updates existing records and creates new records in a NetSuite connection and is used as a target in an operation.
- **Delete:** Deletes records in a NetSuite connection and is used as a target in an operation.

Using NetSuite Connections in Scripts

NetSuite connections can be referenced in a script using certain script functions that use a `netSuiteOrg` as a parameter. For example:

- `NetSuiteGetSelectValue`
- `NetSuiteGetServerTime`
- `NetSuiteLogin`

To add a NetSuite connection to a script to be referenced by one of these functions, drag the configured endpoint from the **Endpoints** tab of the script component palette to the script. Or, if you already know the function you want to use, add it from the **Functions** tab first; then position the cursor after the opening parenthesis of the function and press `Control+Space` to display a list of autocomplete suggestions. Select a connection to insert the connection reference into the script.

For more details on referencing activities in scripts, see Endpoints on the Jitterbit Script page.

Troubleshooting

This section describes issues and workarounds to common issues experienced with NetSuite connections.

Data Center Error

A connection may have previously been tested successfully, but now fails with this error:

> *You are not requesting the correct data center for your company.*

This error may result from a NetSuite account having been moved to another data center on NetSuite's side, resulting in a change to the WSDL URL.

To resolve, change the WSDL to either (1) the account-specific domain or (2) the data center-specific domain. Details on obtaining the URL for your specific account or data center are provided in NetSuite's documentation on URLs for Account-Specific Domains and Understanding NetSuite URLs and Data Centers (login to NetSuite required).

NetSuite recommends using the `getDataCenterUrls` SOAP call to dynamically discover your account-specific NetSuite domain. Account-specific domains are independent of the data center your account is in, so if your data center changes, you will not need to adjust the URL. Once you have the correct WSDL, use it in the configuration of the NetSuite connection in Cloud Studio.

Example of Account-Specific WSDL

- **Original:** `https://webservices.netsuite.com/wsdl/v2016_2_0/netsuite.wsdl`
- **Account-Specific URL:** `http://123456.suitetalk.api.netsuite.com/wsdl/v2016_2_0/netsuite.wsdl`

Example of Data Center-Specific WSDL

- **Original:** `https://webservices.netsuite.com/wsdl/v2016_2_0/netsuite.wsdl`
- **Data Center-Specific WSDL:** `https://webservices.na3.netsuite.com/wsdl/v2016_2_0/netsuite.wsdl`

Permissions Error

Even if testing a NetSuite connection is successful, you may receive an `INSUFFICIENT_PERMISSION` error upon running operations containing activities using that connection.

In this case, you may need to use a different role for generating TBA access tokens or add permissions to the role you are currently using.

To resolve, while generating access tokens, make sure to generate them on either a Full Access or Administrator role or make sure the appropriate permissions are allowed for the role.

Detailed instructions are available in NetSuite's documentation Getting Started with Token-based Authentication (login to NetSuite required).