

Version 2015 Release 2 Release Notes

Revision Date: August 5, 2015
PDF File Version: v5

This document summarizes the changes to NetSuite between Version 2015 Release 2 (also known as 2015.2) and the previous release.

These release notes are subject to change.

Please make sure to review the release notes section of the [Help Center Weekly Updates](#) for a list of the changes made in this build of the release notes.

The Version 2015 Release 2 enhancements and changes listed in this document are not available to NetSuite customers until they are upgraded to Version 2015 Release 2. Please check the version number at the bottom of your Home page if you are unsure which NetSuite version you are using.

Click links in the following table for details about product changes:

Product	Section	Summary
NetSuite	Accounting	<ul style="list-style-type: none"> Period Close by Accounting Book Thomson Reuters Foreign Exchange Rate Provider Custom GL Lines Audit Log Record the Reason for Deleting a Transaction Transaction Numbering Enhancements Fixed Assets Management Enhancements
	Administration	<ul style="list-style-type: none"> Token-based Authentication Enhancements Enhanced Token-based Authentication Endpoint Enhancement to Security Questions New Column Added to Choose Role Page Quicker Deployment of Your Custom Checkout Domain Device ID Authentication CDN Caching for SCA in Sandbox New Development Environment for NetSuite Customizations Application Performance Management (Beta)
	Banking and Payment Processing	<ul style="list-style-type: none"> Preserve Customer Payment and Cash Sale Transactions When Payment Fails Prevent Selecting Incompatible Payment Processing Profiles

Product	Section	Summary
		<p>Display Web Store Payment Methods Based on Shopper Country</p> <p>Repeated Requests to Gateway Not Allowed on Posting Transactions</p> <p>Payment Events to Record Payments Completed Outside of NetSuite</p> <p>Enhancements to Payment Gateway Plugin</p>
	Customer Relationship Management (CRM)	<p>New Email Preferences Page</p> <p>Forwarding and Composing Messages in DMARC-Compliant Format</p> <p>Phasing out CRMSDK-Based Email Templates</p>
	Employee Management	<p>Payroll Enhancements</p> <p>Updated Timesheets</p>
	Items and Inventory Management	<p>Group Average Costing</p> <p>Accounting for Scrap Quantities in Final Component Quantity Calculations</p> <p>Amendments to Buildable Quantity Field Calculation</p>
	Mobile	NetSuite for Android 2.0
	Order Management	<p>Billing-Related Navigation and User Interface Changes</p> <p>Shipping Enhancements</p> <p>Recurring Billing Enhancements</p> <p>Item Detail Statement (IDS) Enhancements and Changes</p>
	Projects	<p>Limit Time and Expenses for Project Tasks</p> <p>Project Template Conversion Enhancements</p> <p>Allocate Resources to Project Tasks with Resource Allocations</p> <p>Job Costing Enhancements</p> <p>Charge-Based Billing Enhancements</p>
	Vendors, Purchasing, and Receiving	<p>Billing Address on Purchasing Transactions</p> <p>Purchases/Vendors Dropdown Menu Changes</p> <p>Deleting Subrecords on Purchase Contracts and Blanket Purchase Orders</p> <p>Order Requisition Enhancements</p> <p>Shared Vendor Saved Searches Display Secondary Subsidiaries</p> <p>Procurement Dashboard</p>

Product	Section	Summary
	Taxation	Enhancements to Certificate of Creditable Tax Withheld at Source (Form 2307) EU Sales List Support for Triangulation of Goods
	User Interface	New NetSuite Status Page New User Preference Enhances Accessibility Credit Card Processing User Interface Elements Relabeled Support for Internet Explorer 9 Ended Support for Safari 5 Ended StickyNotes User Experience Improvements Basic API Integration on SuiteSocial News Feed Email Capture Plugin Support for SuiteSocial Discontinued Japanese Custom Forms
SuiteCommerce	SuiteCommerce InStore	Introducing SuiteCommerce InStore
	SuiteCommerce Advanced	Cache Invalidation by URL Reference Implementation Release Notes
	General SuiteCommerce Settings	Asynchronous afterSubmit User Event Support for Web Store Users
SuiteCloud Platform	SuiteBundler	Bundle Support During Release Phasing Support for Managed Upgrades in Release Preview Accounts
	SuiteBuilder - Customization	Hierarchical Custom Record Types Ability to Make Custom Fields Inactive Preview of Advanced PDF/HTML Template Output Custom Segments (Beta) Changes to the Japanese Custom Address Form and Zip Code Lookup
	SuiteAnalytics (Dashboards, Reporting, and Search)	Tableau Export for Saved Searches and Reports Connect Browser More Options for Custom Fields in Summary Reports New Localization Preference for Search Sort Order Redesigned Report Results Page New Permission to Persist Search Results Labor Cost Now Available for Saved Searches

Product	Section	Summary
		<p>Changes to the Left Side Field in Financial Reports and SuiteAnalytics Connect</p> <p>End of Support for the ODBC 6.0 Driver</p> <p>Pivot Reports BETA Replacement</p>
	CSV Import	<p>CSV Support for Custom Segment Values</p> <p>More Formatting Options for CSV Files</p>
	SuiteFlow (Workflow Manager)	<p>Dynamic Mode for Workflows</p> <p>Action Ordering Rules for Workflows</p> <p>Changes for Workflows Referencing Email Templates</p> <p>Upcoming Change for Workflows with Timesheets</p>
	SuiteScript	<p>SuiteScript 2.0 – Beta Release</p> <p>Script Execution Log – Persistence Enhancement</p> <p>nlapITriggerWorkflow Change</p> <p>nlapIMergeRecord Deprecated</p> <p>Enhancements to Token-based Authentication for RESTlets</p> <p>SuiteScript Support for Use Deletion Reason Feature</p> <p>SuiteScript Change for Extended Accounting Period Close Process Feature</p> <p>Upcoming Change for Scripting with Timesheets</p> <p>New APIs</p> <p>Updated APIs</p> <p>2015.2 SuiteScript Records Browser</p>
	SuiteTalk (Web Services) Integration	<p>Better Tracking of Web Services Integrations</p> <p>Token-based Authentication for SuiteTalk (Web Services)</p> <p>Release Preview Version of 2015.2 WSDL Now Available</p> <p>Araxis Merge Diff File for the 2015.2 Endpoint</p> <p>2015.2 SuiteTalk Schema Browser</p>
	SuiteCloud IDE	Support for SuiteScript 2.0
SuiteApps	SuiteApps (Bundles) Released by NetSuite	<p>Recurring Billing Enhancements</p> <p>Enhancements to Certificate of Creditable Tax Withheld at Source (Form 2307)</p> <p>EU Sales List Support for Triangulation of Goods</p> <p>Procurement Dashboard</p> <p>Warranty and Repairs Management Enhancements</p>

Product	Section	Summary
		Fixed Assets Management Enhancements
		StickyNotes User Experience Improvements
		Basic API Integration on SuiteSocial News Feed
		Email Capture Plugin Support for SuiteSocial
		Enhancements to Payment Gateway Plugin
		Japan-Specific Enhancements and Changes
		Application Performance Management (Beta)

Accounting

Version 2015 Release 2 includes the following enhancements to accounting features:

- [Period Close by Accounting Book](#)
- [Thomson Reuters Foreign Exchange Rate Provider](#)
- [Custom GL Lines Audit Log](#)
- [Record the Reason for Deleting a Transaction](#)
- [Transaction Numbering Enhancements](#)
- [Fixed Assets Management Enhancements](#)

Period Close by Accounting Book

If NetSuite Professional Services has implemented Multi-Book Accounting in your account, you can individually close and reopen accounting periods associated with any accounting book, without impacting your other accounting books. When you enable the Extended Accounting Period Close Process feature, unique period close checklists become available for each of your accounting books including newly activated secondary accounting books. The closing tasks for all accounting books or the primary accounting book will be different from the closing tasks of a secondary accounting book.

You choose the accounting book for which you want to close a period through the Filters section at the top of the Manage Accounting Periods page. When you select an accounting book, the list of displayed accounting periods provides the closing status and a link to the respective period close checklist. If you select a secondary accounting book, only accounting periods after the book effective date display in the list.

Marking a check list task complete partially unlocks dependencies between certain tasks. For details about tasks specific to various accounting books as well as task and period dependencies, see the help topic [Closing Tasks and Their Dependencies](#).

You can edit transactions in a period that is locked in one or more accounting books, but closed in other accounting books. NetSuite determines whether a transaction can be changed based on the specified posting period in the transaction record. For details about how edits to transactions affect closed or locked accounting books, see the help topic [Transaction Edits in Closed, Locked Accounting Periods](#).

If you enable the Extended Accounting Period Close Process feature, you are not required to individually close accounting books. You can continue to close all of your books at the same time. For more information about the feature, see the help topic [Accounting Book Period Close Management](#).

Thomson Reuters Foreign Exchange Rate Provider

If you use the Currency Exchange Rate Integration feature, Version 2015 Release 2 offers Thomson Reuters as an additional exchange rate provider to the current provider, Xignite.

Thomson Reuters has received awards from Waters Ranking in 2013 for Best Reference Data Provider and Best Low-Latency Data Feed Provider. This rate provider offers real-time, current market rates. Thomson Reuters supports thousands of global and regional banks, institutional investors, and corporate clients around the world both in developed and emerging markets.

Thomson Reuters offers a four significant decimal exchange rate, which means that the leading zero displays but does not count as one of the four digits. For example, both 0.1234 and 0.0001234 represent four significant digits.

In this release, NetSuite offers three methods for obtaining currency exchange rates: triangulation calculation, direct, and inverse.

Triangulation calculation by NetSuite is useful when the two home currencies involved in the transaction are not available from the selected rate provider. This triangulated calculation provides a method for two entities to simultaneously transact business in one currency, and at the same time repatriate money and profits back to the two home currencies. For NetSuite to be able to calculate the exchange rate for a cross currency pair, you must specify the intermediate, or anchor currency to use in the calculation.

Direct currency exchange rates are those that are provided directly from your specified exchange rate provider.

Inverse rates are those exchange rates that are based on the direct rate used by another NetSuite account. The inverse rate that displays on the Currency Exchange Rates page at Lists > Accounting > Currency Exchange Rates is the most recent rate for that currency rate pair before the as-of-date. Note that if an as-of-date is not provided, today's date and the current time serve as the as-of-date. The initial rate stored within NetSuite was obtained when the new currency was created in the system. Subsequent rate updates and as-of-dates were provided either from the rate provider or from a manually entered exchange rate.

For more information about the Currency Exchange Rate Integration feature, rate providers, methods for obtaining exchange rates, and anchor currencies, see the help topic [Using the Currency Exchange Rate Integration Feature](#).

Custom GL Lines Audit Log

The Custom GL Lines Audit Log shows custom lines created by Custom GL Lines plug-in implementations in a NetSuite account. Use the audit log to view any custom lines created for a specific transaction, all transactions, or for a specific plug-in implementation. The audit log also lists custom lines for transactions deleted after the custom lines were created. In addition, if a transaction fails to save due a plug-in implementation error, the audit log displays the error generated by NetSuite when the plug-in implementation ran.

For more information, see the help topic [Working with the Custom GL Lines Audit Log](#).

The following screenshot shows the Custom GL Lines Audit Log default view for a transaction:

LOG ENTRY NO.	TRANSACTION TYPE	NUMBER	LINES ADDED	REVISION	ERROR ▲	ELAPSED TIME [MS]	DATE
6	Sales Order	182	2	6/9/2015 10:05 pm		379	6/9/2015 10:05 pm

You can click the links in the **Lines Added** and **Revision** columns to open the [Custom GL Audit Log Line Page](#) and [Custom GL Lines Plug-in Revision Page](#), respectively.

Accessing the Custom GL Lines Audit Log

You can access the audit log using one of the following methods:

- Click the link in the **Custom Script** column from the GL Impact report on a transaction. The link opens the Custom GL Lines Audit Log page for the transaction, even if the plug-in implementation that created the custom lines was deleted.

GL Impact								Return to Sales Order #184
								TOTAL: 6
ACCOUNT	AMOUNT (DEBIT)	AMOUNT (CREDIT)	POSTING	MEMO	NAME	DEPARTMENT	CLASS	CUSTOM SCRIPT
4002 Sales : Sales - Merchandise		4.12	No	Tax Revenue Increase				Tax Expense on Accruals
6130 Miscellaneous Expense	4.12		No	Tax Expense				Tax Expense on Accruals

Note: You must have administrator or full access permissions for the link in the **Custom Script** column to appear.

- Create a saved search using the **Custom GL Lines Audit Log** search type.

New Saved Search

Company
Competitor
Contact
Coupon Code
Custom GL Lines Plug-in Audit Log
Customer
Customer Satisfaction Survey

Custom GL Audit Log Line Page

The Custom GL Lines Audit Log Line page shows the balancing custom lines added to a specific transaction. Click the link in the **Lines Added** column on the Custom GL Lines Audit Log page to access the Custom GL Lines Audit Log Line page.

Custom GL Audit Log Line							List	Search	
VIEW: Default Customize View									
<div style="border: 1px solid #ccc; padding: 5px;"> FILTERS STYLE Normal </div>									
									TOTAL: 2
LOG ENTRY NO.	ACCOUNT ID	CREDIT	DEBIT	MEMO ▲	CLASS ID	DEPARTMENT ID			
6	83	0	0.9	Tax Expense					
6	56	0.9	0	Tax Revenue Increase					

Custom GL Lines Plug-in Revision Page

The Custom GL Lines Plug-in Revision page shows the specific version of the plug-in implementation that created a custom line. Click the link in the **Revision** column on the Custom GL Lines Audit Log page to access the Custom GL Lines Plug-in Revision page.

PLUG-IN IMPLEMENTATION	FILE	FILE CONTENTS	MODIFIED	MODIFIED BY	LIBRARY	LIBRARY MODIFIED
Tax Expense on Accruals	taxexpenseonaccruals_SO.js	function customizeGllmpact(transactionRecord, standardLines, customLines, book) { var acc1; var acc2;(more...)	6/9/2015 10:05 pm	A Wolfe-admin		

Delete Custom GL Lines Plug-In Implementation

As of Version 2015 Release 2, you can delete a Custom GL Lines plug-in implementation or remove a bundle that contains a Custom GL Lines plug-in implementation. If you delete the plug-in implementation, any custom lines created by the implementation remain in the NetSuite account on the transaction for which they were created. This enables you to delete a plug-in implementation and not lose the general ledger impact of the custom lines. Use the Custom GL Lines Audit Log to view all custom lines created by current and deleted plug-in implementations.

In previous releases, you could not delete a Custom GL Lines plug-in implementation if the plug-in implementation had previously created custom lines on a transaction.

The following screenshot shows the Custom GL Lines audit log for a deleted plug-in implementation:

LOG ENTRY NO.	TRANSACTION TYPE	NUMBER	LINES ADDED	PLUG-IN IMPLEMENTATION	REVISION	ERROR A	ELAPSED TIME [MS]	DATE
6	Sales Order	182	2	Tax Expense on Accruals	6/9/2015 10:05 pm		379	6/9/2015 10:05 pm

Record the Reason for Deleting a Transaction

In some countries, providing the reason why a transaction was deleted is a legal requirement. The Use Deletion Reason feature satisfies this requirement by making it mandatory for users to record the reason for deleting a transaction.

All transactions listed at Setup > Company > Auto-Generated Numbers on the Transactions subtab are impacted by the Use Deletion Reason feature. When this feature is enabled, users must provide a reason for deleting a transaction record. The Transaction Numbering Audit Log provides a list of the deleted transactions, their transaction number, and the date on which the transaction was deleted and by whom. It also provides the reason the user deleted the transaction record and any related memo.

The Use Deletion Reason feature includes two standard deletion reasons: Original Document Damaged and Other. These two deletion reasons are available throughout your NetSuite account. If users choose the standard reason Original Document Damaged, writing a memo is optional. If users choose the standard reason Other, writing a memo is required.

Users with the Administrator role or access to Accounting Lists can create and modify deletion reasons. To create, modify, inactivate, and choose display languages for a unique deletion reason, go to Setup > Accounting > Accounting Lists > New and click Transaction Deletion Reason.

Transactions can be deleted from the transaction record by clicking Actions > Delete. If inline editing is enabled in your account, transactions can also be deleted from the transaction's List page. Hover over the New column and from the menu that appears, click Delete Record. Both deletion methods present a Delete popup specific to the transaction type record being deleted. Users must choose a deletion reason and have the option to include a memo. Users choosing the deletion reason Other must include a memo. Clicking Delete removes the transaction record from your system.

You can disable the feature at any time. Transactions to be deleted will no longer require a deletion reason. Deletion reasons will not be accessible from Accounting Lists. The Transaction

Numbering Audit Log will continue to provide the deleted transactions and transaction numbers, deletion date, user performing the deletion, reason for deletion, and any related memo.

For more information about the feature, see the help topic [Recording a Reason for Deleting a Transaction](#).

Transaction Numbering Enhancements

Version 2015 Release 2 includes the following enhancements to the Auto-Generated Numbers feature. These enhancements have changed the way the auto-generated numbering feature behaves. If you currently use the feature, it is recommended that you read the help topic [Setting Up Auto-Generated Numbers](#) to learn about the changes.

- [Show Transaction Number on Transaction Records and Standard Reports](#)
- [Prevent Duplicated Auto-Generated Document Numbers](#)
- [Prevent Duplicate Numbers for External Numbering Sequences](#)
- [Reset Initial Transaction and Document Numbering Sequences](#)

Show Transaction Number on Transaction Records and Standard Reports

Auto-generated numbers for transaction records are internal, gapless numbers that cannot be overwritten and are generated when the record is saved. These internal numbers are generated for each transaction type.

Prior to Version 2015 Release 2, transaction types appeared on the Transactions subtab at Setup > Company > Auto-Generated Numbers. In Version 2015 Release 2, the subtab was renamed Transaction Numbers, and is invisible by default. If you must configure transaction type options, you can display the Transaction Numbers subtab. Go to Setup > Company > General Preferences and check the Show Transaction Numbering Setup box.

Some of the configuration options specified on the former Transactions subtab now appear on the new Transaction Numbers subtab. This occurs for account-based and externally generated numbered transactions because these configuration options modified both document number and transaction number sequences. Configuration options specified for transaction types other than account-based or externally generated numbers were copied to the new Document Numbers subtab. You can reconfigure options on the Document Numbers subtab, as required.

For new customers, document type numbers are not configured, except for the beginning number, which is set to 1.

By default, internal transaction numbers do not display on transaction type records or standard reports, but you can customize any transaction type record to display this number. To display internal transaction numbers on transaction records, go to Customization > Forms >

Transaction Forms. Locate the transaction type to be customized and then click the Customize link next to the transaction name. Click the Screen Fields subtab. In the Label column, locate Transaction Number, check the Show box and then save the record.

Internal transaction numbers are searchable when the number is hidden or visible on the user interface.

Transactions that are successfully imported or copied are numbered according to the internal transaction numbering sequence at the time the import.

Important: Prior to Version 2015 Release 2, NetSuite permitted overwriting the internal number of transaction records where the Allow Override option was enabled for the transaction type. In these historical transactions, there may be duplicate internal numbering. In Version 2015 Release 2, NetSuite prohibits overwriting transaction type numbering sequences, instead permitting the overwriting of document numbers. NetSuite will not renumber historical internal transaction numbers.

Prevent Duplicated Auto-Generated Document Numbers

Auto-generated numbers for document types are external numbers such as the number for a vendor bill, or account based numbers such as a check number. Document numbers can also be type based, which store a document number for type based transactions. Type based numbers are available as custom columns in searches and lists.

When a document type has the Allow Override option enabled, users can unintentionally introduce duplicated document numbers and numbering gaps into your account. To prevent duplicated document type numbers and numbering gaps, you can customize transaction type records to display the Generate TranID on Save check box. When this box is checked, NetSuite records the document number when the user saves the transaction record, rather than when the user opens the transaction record. This ensures that the document number is unique for the document type, as if the Allow Override option was not enabled.

If the Allow Override option is enabled for a specific document type and the Generate TranID on Save box is unchecked, NetSuite records the document number when the user opens the transaction record. This permits multiple instances of the same transaction record being assigned the same TranID value. For example, User One opens a transaction record assigned TranID 100. Before User One saves the transaction record, User Two opens a second instance of the same transaction record, which is assigned the same TranID 100. If you attempt to save a record that has a duplicate TranID, NetSuite presents an error message. You can cancel the message and save the duplicated number.

Prevent Duplicate Numbers for External Numbering Sequences

For external numbering sequences such as vendor bill, the document number is unique for each vendor. This permits the same document number being assigned to more than one

vendor. NetSuite prevents a vendor from having a duplicated document number on the same transaction type. This applies to all four external transaction types whether recorded through the user interface or through a CSV import: Bill (Vendor Bill), Bill Credit (Vendor Credit), Credit Card Charge (Customer Credit Card), and Credit Card Refund (CCard Refund).

Reset Initial Transaction and Document Numbering Sequences

In some countries, resetting an initial transaction number each fiscal year is a legal requirement. Users with the Administrator role can lower the initial record number for both transaction type numbers and document type numbers. Lowering the initial number permits duplicate numbering in your account. In addition, lowering the initial document number during the update of an existing transaction record does not update that document number on that transaction record.

Administration

Version 2015 Release 2 includes the following enhancements to Administration features:

- [Token-based Authentication Enhancements](#)
- [Enhanced Token-based Authentication Endpoint](#)
- [Enhancement to Security Questions](#)
- [New Column Added to Choose Role Page](#)
- [Quicker Deployment of Your Custom Checkout Domain](#)
- [Device ID Authentication](#)
- [CDN Caching for SCA in Sandbox](#)
- [New Development Environment for NetSuite Customizations](#)
- [Application Performance Management \(Beta\)](#)

Token-based Authentication Enhancements

Version 2015 Release 2 introduces enhancements to Token-based Authentication: an enhancement to the token endpoint, support for SuiteTalk (web services), and to token-based authentication for RESTlets.

For more information on these enhancements, see the following release notes:

- [Enhanced Token-based Authentication Endpoint](#)
- [Token-based Authentication for SuiteTalk \(Web Services\)](#)
- [Enhancements to Token-based Authentication for RESTlets](#)

Enhanced Token-based Authentication Endpoint

The Token-based Authentication endpoint has been enhanced, providing the ability to revoke a token. The new /revoketoken method is similar to the existing /issuetoken method.

Enhancement to Security Questions

Users must now provide their current password before they can change security questions and answers. The answers are now masked by default. Users can clear the Hide Answers box, if desired.

Update Security Questions

Please select and provide answers to the 3 questions shown below. The answers to these questions are used to verify your identity if you forget your NetSuite password.

After successfully completing your security questions, you may reset your password by using the "Forgot your password?" link on the login page.

CURRENT PASSWORD *

QUESTION 1
What was your childhood nickname?

ANSWER 1 *

QUESTION 2
In what city did you m...se/significant other?

ANSWER 2 *

QUESTION 3
What is your matern...ther's maiden name?

ANSWER 3 *

HIDE ANSWERS

For the detailed procedure, see the help topic [Updating Security Questions](#).

New Column Added to Choose Role Page

For Advanced Partner Center and Partner Center users, the Choose Role page now displays the Partner Name or Vendor Name record to which the respective user has access.

DEFAULT ROLE	ROLE	COMPANY	PARTNER / VENDOR	LAST LOGIN	TOTAL: 3
<input type="checkbox"/>	Accountant	NetSuite, INC		6/5/2015 6:43 am	
<input type="checkbox"/>	Advanced Partner Center	NetSuite, INC	Partner Company Name		
<input type="checkbox"/>	CEO	NetSuite, INC			

Quicker Deployment of Your Custom Checkout Domain

Prior to this release, NetSuite deployed custom domains two times a month. Procedures have been automated to minimize the time between the uploading of SSL certificate and the

deployment of your custom domain. Domains are now deployed many times each day. You will be notified by email when your domain goes live. Before uploading your SSL certificates, ensure you have completed all the necessary steps in [Prerequisites for Setting Up a Custom Checkout Domain in NetSuite](#) and in [Setting Up a Secure Checkout Domain in NetSuite](#).

Device ID Authentication

Point of sale (POS) devices and mobile devices can now be part of the authentication process. Administrators can ensure that users can access NetSuite only from authorized devices by maintaining a list of trusted devices, configure device-ID only roles, search for specific devices using standard NetSuite search capabilities, and see accepted and rejected devices in login audit trails.

CDN Caching for SCA in Sandbox

You no longer have to wait until your SuiteCommerce Advanced (SCA) web site moves into production to verify its performance. Now you can use Content Delivery Network (CDN) Caching to test your SCA web site optimizations and site loading time in your sandbox. With CDN caching enabled, website pages load more quickly because assets (such as images) that are reused are cached on the CDN. Setting up CDN Caching in your sandbox is the same as in production. See the help topic [CDN Caching in SuiteCommerce Advanced](#) for more information.

New Development Environment for NetSuite Customizations

Important: This feature is targeted to be available in September 2015.

The NetSuite Development account is a new team development environment. It provides an improved experience for developers of NetSuite customizations and an enhanced change management model. NetSuite Development accounts are never refreshed with production data and are not directly tied to production accounts. This separation of these development accounts prevents in-progress customizations from being overwritten by refreshes and enables controlled access to sensitive, restricted, and financial data. These new lightweight development accounts support rapid deployment and testing processes.

If you are interested in obtaining the NetSuite Development account, please contact your NetSuite account manager.

Banking and Payment Processing

Version 2015 Release 2 includes the following enhancements to banking and payment processing features:

- Preserve Customer Payment and Cash Sale Transactions When Payment Fails
- Prevent Selecting Incompatible Payment Processing Profiles
- Display Web Store Payment Methods Based on Shopper Country
- Repeated Requests to Gateway Not Allowed on Posting Transactions
- Payment Events to Record Payments Completed Outside of NetSuite
- Enhancements to Payment Gateway Plugin

Preserve Customer Payment and Cash Sale Transactions When Payment Fails

With Release 2015 Version 2, when a payment fails due to a payment hold, you no longer have to re-enter the customer payment or the cash sale data. Now you can set a new preference to preserve customer payment and cash sale transactions when the gateway responds with a payment hold.

To enable this preference, go to Setup > Accounting > Accounting Preferences. On the Items/ Transactions subtab, under Payment Processing, check the **Preserve Transactions when Payment Is on Hold** box.

Enable this preference if you use a single customer payment to cover multiple invoices. If this payment fails due to a payment hold, the transaction is preserved with the Unapproved Payment status. This eliminates the requirement of having to reselect all of the invoices for the customer payment.

Similarly, if you import a large number of customer payment or cash sale transactions, and some transactions fail during processing, you do not have to import these transactions again. Instead, they are saved with the Unapproved Payment status.

Transactions that have the Unapproved Payment status are non-posting. If you delete these transactions, they do not affect your accounting balance.

Important: A customer payment or a cash sale is preserved only if the status in the payment event is Payment Hold. If the status is Reject (for example, because of fraud or a stolen card), the transaction is not preserved.

To manually approve a cash sale or a customer payment that has a payment hold, open the transaction page, and then click the Override Payment Hold button.

In some cases, the gateway may later confirm the payment by sending another request with the Accept status to NetSuite. When this happens, the transaction status automatically changes

to Deposited or Not Deposited (depending on the account selected on the payment method record).

However, if the gateway sends a new request with a Reject status, the transaction remains non-posting and has the Unapproved Payment status.

Warning: When you approve a transaction with the Override Payment Hold button, NetSuite does not send a new request to the gateway to validate the payment. Before you click this button, make sure you have validated the payment outside of NetSuite. For example, you can check your bank account or contact the gateway back office.

Tip: To preview cash sale or customer payment transactions that have a payment hold, create a saved transaction search. In this search, filter transactions with Payment Event Result set to Payment Hold, and Type set to Cash Sale (or Payment). For more information, see the help topics [Viewing Customer Payments that Have a Payment Hold](#) and [Viewing Cash Sales that Have a Payment Hold](#).

Prevent Selecting Incompatible Payment Processing Profiles

With Version 2015 Release 2, you can prevent processing payments with payment processing profiles that are incompatible with customers' data. A new preference is available to filter payment processing profiles in the Payment Processing Profile field. (Prior to Version 2015 Release 2, this field was called Credit Card Processor.) Go to Setup > Accounting > Accounting Preferences. On the Items/Transactions subtab, under Payment Processing, check the **Use Strict Rules for the Selection of Payment Processing Profiles** box.

With this preference enabled, only payment processing profiles that support the selected payment method and the customer's currency are displayed. If you use NetSuite OneWorld, the payment processing profile must also support the customer's subsidiary.

Note: This restriction applies to the Preferred CC Processor field on the Financial subtab of the Customer record.

When this preference is enabled, it takes precedence over the **Restrict Payment Methods by Customer Currency** shopping preference. This means that even if the shopping preference is disabled, only payment methods that support the shopper's currency are displayed. In addition, if no payment method matches the shopper's currency, no payment method shows on the Payment Information page.

Display Web Store Payment Methods Based on Shopper Country

With Version 2015 Release 2, you can display payment methods on your web store based on the shopper's country. To specify a list of countries for a payment method, go to Setup >

Accounting > Accounting Lists. Edit the payment method. In the Countries field, select the countries for which you want to show this payment method.

Note: If you do not select any country in the Countries field, the payment method shows for all countries.

Important: This feature is enabled only for SuiteCommerce Advanced web stores.

Repeated Requests to Gateway Not Allowed on Posting Transactions

With Version 2015 Release 2, when a transaction posts to the selected ledger account, the Charge Credit Card and the Credit Card Approved (or CC Approved) boxes become disabled.

Disabling these boxes prevents sending multiple requests to the gateway after the transaction has posted to the account. It also prevents converting the state of transaction lines on the GL Impact page from posting to non-posting.

Payment Events to Record Payments Completed Outside of NetSuite

With Version 2015 Release 2, payment events are created to record non-credit card payments such as cash, check, or wire transfer when used with the cash sale, customer payment, customer deposit, customer refund, cash refund, and customer credit transactions.

Also, when you check the Credit Card Approved (or CC Approved) box for payments validated outside of NetSuite, payment events are created to record these payments.

Payment events for payments validated outside out NetSuite are recorded for both posting and non-posting transactions.

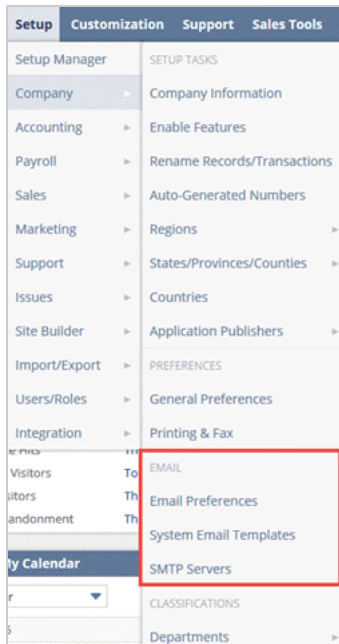
Customer Relationship Management (CRM)

Version 2015 Release 2 includes the following enhancement to CRM features:

- [New Email Preferences Page](#)
- [Forwarding and Composing Messages in DMARC-Compliant Format](#)
- [Phasing out CRMSDK-Based Email Templates](#)

New Email Preferences Page

Prior to Version 2015 Release 2, company email preferences could be set at Setup > Company > Preferences > Printing, Fax & Email Preferences. In Version 2015 Release 2, email preferences can be set at Setup > Company > Email > Email Preferences.



Forwarding and Composing Messages in DMARC-Compliant Format

DMARC has become the new standard for email authentication, making message delivery more accurate and dramatically reducing the frequency of spam and phishing email messages worldwide. Major ISPs and MSPs such as AOL, Comcast, Yahoo and Gmail have signed-up to DMARC, and today over 60% of the world’s mailboxes are protected using these policies.

NetSuite Version 2015 Release 2 introduces the first of a number of changes to the way email is authenticated within NetSuite. Two new system preferences are introduced on the new Email Preferences page at Setup > Company > Email > Email Preferences. These preferences create DMARC-compliant messages in the following instances:

Preference	Description
Forward email messages in DMARC-compliant format	This includes forwarded email messages (bounce messages, incoming email received from mail forms).
Compose email messages as DMARC compliant	This includes all messages sent using the rich-text editor within NetSuite. This preference requires that you have set up domainkeys identified mail (DKIM).

For more information on DMARC, see [what is DMARC?](#) and [Yahoo DMARC Policy](#).

Phasing out CRMSDK-Based Email Templates

In NetSuite Version 2015 Release 2, the following types of email will no longer support CRMSDK tags.

- Single email messages composed using the rich-text editor
- Bulk email merge operations
- Campaign email

Note: Campaigns set up to use an existing CRMSDK template will continue to work until you edit the campaign record.

Also from Version 2015 Release 2, using CRMSDK templates in **new** mass update jobs Lists > Mass Updates will no longer be possible. Users should use a scriptable template instead, however, existing mass updates using CRMSDK templates will continue to work. Planned for Version 2016 Release 1, mass updates still using CRMSDK templates will automatically be migrated to the scriptable format.

The above changes are part of the ongoing preparation to switch to the more powerful FreeMarker format of scriptable templates. These changes will be fully adopted in Version 2016 Release 1. For more information, see the help topic [Scriptable Templates](#).

Important: There will be **no** impact to existing SuiteFlow workflows using CRMSDK templates in this version. However, if you convert a CRMSDK template to a scriptable template, the new scriptable template will have an ID different from the old template. Any references to this template ID will must be updated in your scripts and workflows. Workflow support of CRMSDK templates will be deprecated in Version 2016 Release 1.

Employee Management

Version 2015 Release 2 includes the following enhancements to employee management features:

- [Payroll Enhancements](#)
- [Updated Timesheets](#)

Payroll Enhancements

The following enhancements are included for the Enhanced Premier Payroll feature in Version 2015 Release 2:

- [Simplified Overtime Pay Rate Setup](#)
- [Tax Compliance for Multi-State Employees](#)

Simplified Overtime Pay Rate Setup

If you are using enhanced premier payroll, you can now derive overtime pay rates using a multiplier of a linked payroll item. The derived rate is dependent on the rate for the payroll item to which it is linked. When the base rate changes, the derived rate changes automatically.

Two new fields on the payroll item record support derived rates:

Derived From – Select the payroll item on which the new rate will be based. This is the linked payroll item. This field is available for earning, deduction, and employer contribution payroll item types. Expense and commission earning types are not supported. Earnings can be derived from earnings. Deductions can be derived from earnings or deductions. Contributions can be derived from earnings, deductions, or contributions.

Multiplier – This field is visible only when a payroll item is selected in the Derived From field. Enter the multiplier as a number, not a percentage. This number can be a decimal fraction. Negative numbers are also supported.

The derived rate for the new payroll item is the Multiplier times the pay rate of the payroll item selected in the Derived From field.

A new multiplier field in the Payroll subtab of the Employee record enables you to override the default multiplier set in the derived payroll item. You can also override the default multiplier from the payroll batch.

Tax Compliance for Multi-State Employees

NetSuite has added support in enhanced premier payroll for tax compliance for employees who work in different states during a pay period. NetSuite calculates the pertinent taxes for the jurisdiction where work is performed, as well as the employee's primary office and home jurisdictions.

To enable the multi-state taxation option, go to Setup > Payroll > Set Up Payroll, and check the Enable Multi-State Calculation box in the Paychecks group.

When multi-state calculation is enabled, an editable Payroll Item Workplace field is added to the Earnings, Time, and Commissions subtabs for paychecks and bulk edit. The Payroll Item Workplace field is not available for expense reimbursements.

The dropdown list for the Payroll Item Workplace field includes all of the workplaces you have configured for your company. Select a workplace from the list to specify where the payroll item was earned. If you add a payroll item and leave the Payroll Item Workplace field blank, it defaults to the workplace on the employee record when you save the change.

Updated Timesheets

When creating a timesheet, the memo icon for any memos entered on individual time entries can be seen after the timesheet has been saved. Previously, when a memo was created on a time entry the memo icon was immediately updated. Updating the memo icon when the timesheet is saved improves speed and performance.

The release of an updated scripting engine providing improved SuiteScript and SuiteFlow support for timesheets is targeted for a future release. It is recommended that you do not modify or create any scripts or workflows for timesheets until after the release of the new scripting engine.

This recommendation is intended to save you time and effort by ensuring that you create or modify any scripts or workflows only one time using the new engine. Unmodified scripts and workflows will continue to function without changes until the new engine is released. Additional instructions and information will be available when script and workflow updates are required.

Order Management

Version 2015 Release 2 includes the following enhancements to order management features:

- [Billing-Related Navigation and User Interface Changes](#)
- [Shipping Enhancements](#)
- [Recurring Billing Enhancements](#)
- [Item Detail Statement \(IDS\) Enhancements and Changes](#)

Billing-Related Navigation and User Interface Changes

Billing improvements for Version 2015 Release 2 include navigation and other user interface changes to provide an improved user experience.

The following navigation changes are for NetSuite administrators or any role using the System Administrator Center:

Navigation Prior to Version 2015 Release 2	Navigation as of Version 2015 Release 2
Transactions > Billing > Bill Runs	Transactions > Billing > Process Billing Operations From Process Billing Operations, execute a bill run by checking the Bill Run box, then click Run.
Transactions > Billing > Cash Sales	Transactions > Sales > Enter Cash Sales
Transactions > Billing > Invoices	Transactions > Sales > Create Invoices
Transactions > Billing > Billing Groups	Lists > Relationships > Billing Groups

Navigation Prior to Version 2015 Release 2	Navigation as of Version 2015 Release 2
Transactions > Billing > Bill Run Schedules	Transactions > Billing > Schedule Billing Operations

The following navigation changes are for accounts receivables professionals or any role using the Accounting Center:

Navigation Prior to Version 2015 Release 2	Navigation as of Version 2015 Release 2
Financial > Billing > Bill Run	Financial > Billing > Process Billing Operations From Process Billing Operations, execute a bill run by checking the Bill Run box, then click Run.
Financial > Billing > Cash Sales	Customers > Sales > Enter Cash Sales
Financial > Billing > Invoices	Customers > Sales > Create Invoices
Financial > Billing > Billing Groups	Customers > Lists > Billing Groups
Financial > Billing > Bill Run Schedules	Financial > Billing > Schedule Billing Operations
Customers > Lists > Billing Schedules	Financial > Lists > Billing Schedules

One navigation change was made for sales representatives or any role using the Sales Center. Customers > Transactions > Sales Orders has been removed. Access Sales Orders from Opportunities > Transactions > Sales Orders.

The Bill Run record has been renamed to Billing Operation. On the Billing Operation page, select Bill Run in the Billing Operation Options section, then click Run to initiate a bill run.

Click Transactions > Billing > Process Billing Operations > Status to show the Billing Operations list. The New Bill Run button is now the New Billing Operation button. Columns in the list of processed billing operations are reordered for an improved user experience.

Submission ID	DATE/TIME	CUSTOMER GROUP	SHORTNAME	RECORDS	NOT STARTED	SUCCESS	GLOBAL FAILURES	FAILURES	CHANGES	CREDIT MEMOS	INVOICES	CASH SALES
201	6/18/2015 1:03 pm	Interiors Construction Services		0	0	0	0	0	0	0	0	0
101	6/18/2015 8:04 am			1	0	1	0	0	0	0	0	0
5	6/17/2015 9:50 pm			3	0	3	0	0	0	0	0	0
3	6/17/2015 9:50 pm	GRP - 2015041704130		3	0	3	0	0	0	0	0	3
1	6/17/2015 9:43 pm	GRP - 2015041704131		3	0	3	0	0	0	0	0	3

On the Billing Operation result page, subtabs are reordered. The Summary subtab is now first, and there is a new Submission Details subtab, which shows all parameters selected for a submitted batch process.

Billing Operation

Primary Information

SUBMISSION ID	NAME
201	
DATE/TIME	HISTORICAL NAME
6/18/2015 1:03 pm	
	DESCRIPTION
	Billing Operation
	OFFSET
	0

Billing Operation Options

BILLING RUN

CREDIT RUN

BILL RUN

Customer

CUSTOMER GROUP: Interiors Construction Services ENTITY TYPE: Customer

ALL CUSTOMERS

Summary | Records | Failures | Submission Details

RECORDS	NOT STARTED	SUCCESS	GLOBAL FAILURES	FAILURES	CREDIT MEMOS	INVOICES	CASH SALES
0	0	0	0	0	0	0	0

Shipping Enhancements

Version 2015 Release 2 introduces the following enhancements.

- **FedEx One Rate** – Offer flat rate shipping that does not require you to weigh or measure boxes and packs under 50 pounds. This is available only for FedEx Express services within the U.S.
- **FedEx Address Validation** – Validate a delivery address when fulfilling an order. Previously, this was available only for U.S. addresses. In Version 2015 Release 2, address validation is now available for more than 40 countries.
- **Web Store Restrictions for Shipping Items** - Prior to Version 2015 Release 2, designating which shipping methods are available for a specific state was available only in the U.S. Now, you can set shipping restrictions for more than 50 countries in your web store.
- **Omit Packaging** – Select this preference to disable automatic packaging logic that occurs during item fulfillment. This prevents an error during item fulfillment if your shipping item exceeds 50 packages.

Note: This is not available for FedEx, UPS, and USPS.

- **Support for UPS Device ID** – Introduces a fraud prevention service deemed mandatory by UPS when you register a new UPS account or edit your existing UPS account.

For more information, see the help topic [Shipping](#).

Items and Inventory Management

Version 2015 Release 2 includes the following enhancements to items and inventory management features:

- Group Average Costing
- Accounting for Scrap Quantities in Final Component Quantity Calculations
- Amendments to Buildable Quantity Field Calculation

Group Average Costing

In Version 2015 Release 2, group average costing enables you to track one average cost for an item across multiple locations within a defined group.

Important: When using Group Average Costing, NetSuite is not currently able to report on the balance sheet by location. Before using Group Average Costing, you must contact Customer Support to submit a request to enable it. If your request is approved, you are required to sign a disclaimer regarding this request. Contact NetSuite Customer Support for details.

Group average costing is available only when you have enabled the Multi-Location Inventory feature. Group average costing is available for inventory items and assembly items.

First, create a location costing group record to track the locations associated with that group. A location can be assigned to a location costing group in two ways:

- On the location record, select the appropriate location costing group.
- On the location costing group record, select the location.

A location can be assigned to only one location costing group. Locations are not required to be assigned to a location costing group. The location list on the location costing group record is filtered by your permission to access each location and subsidiary.

Each time an inventory related transaction with costing impact is processed for a location costing group member, a group average cost is recalculated. The single average cost is calculated by dividing the total inventory value across locations by the total quantity across all locations. This calculated cost is synced within the group and is used in costing calculations for all locations.

Note that individual items are not assigned to a location costing group. All inventory and assembly items within a location that are assigned the Group Average costing method are included in the location costing group assigned to that location.

If an item is assigned the Group Average costing method, but one or more of its locations are not included in a costing group, then for those locations, the item's costing method is effectively Average, not Group Average because no grouping calculations are done for those locations.

Inventory Adjustment Worksheets are not be available for items with a Group Average costing method.

Note: Individual location inventory account values may not equal an item's value, calculated as (group average cost x quantity). However, the consolidated group inventory account value is accurate when calculated as (group average cost x inventory quantity across all group locations) and these results are used for reports.

Adding Locations to a Location Costing Group

New locations you create can be assigned to an existing location costing group, but only if no transactions exist in the new location for any items with a costing method of Group Average.

An existing location can be assigned to a location costing group, but only if there are no transactions associated with the existing location for any items with a costing method of Group Average.

After a location record has been assigned to a location costing group, it can be removed from the location costing group only if there are no transactions associated with the location for any items with a costing method of Group Average.

Group Average Costing and Subsidiaries

Note the following regarding NetSuite OneWorld accounts.

The locations in a location costing group can be associated with one or more subsidiaries. When using locations from different subsidiaries in the same location costing group:

- All subsidiaries in the same location costing group must have the same base currency.
- If any subsidiary within a group is using the Multi-Book Accounting feature and the Multiple Currencies feature, all subsidiaries in the group must have the same secondary books and identical currencies. This is required to be able to calculate the group average cost across all subsidiaries in all currencies.

Accounting for Scrap Quantities in Final Component Quantity Calculations

In previous versions of NetSuite, when using order completion with backflush, values associated with scrapped material had to be manually input into the system. In Version 2015 Release 2, these values are now automatically accounted for in the Consumed Components on Completion calculations.

Amendments to Buildable Quantity Field Calculation

In previous versions of NetSuite, the Buildable Quantity field ignored attributes such as the Effective Date of Components and Revision Control when calculating the Buildable Quantity

value. In Version 2015 Release 2, these attributes are now included in the calculation, providing a more accurate value based on current and future production conditions.

Mobile

Version 2015 Release 2 includes the following enhancement to NetSuite for Android:

NetSuite for Android 2.0

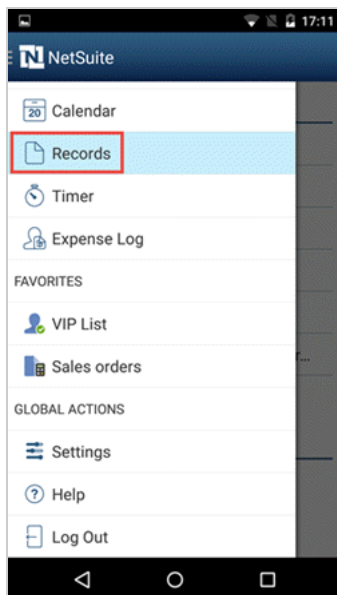
The strategy of creating a fully unified user experience across Android and iOS devices takes a step closer in this version, with the introduction of the following enhancements.

Important: NetSuite for Android 2.0 requires Android version 4.2 and higher.

- Complete support for all mobile supported records
- Full record customization support (SuiteCloud), including SuiteScript, Workflow, and other record-level customization
- Scheduling through the native Calendar feature
- Search functionality – global and saved search support
- Add to favorites

Mobile Records

In Android 2.0 users can access all mobile-supported record types directly from the application's Navigation Draw. This includes entity records, Transactions, and Custom records.



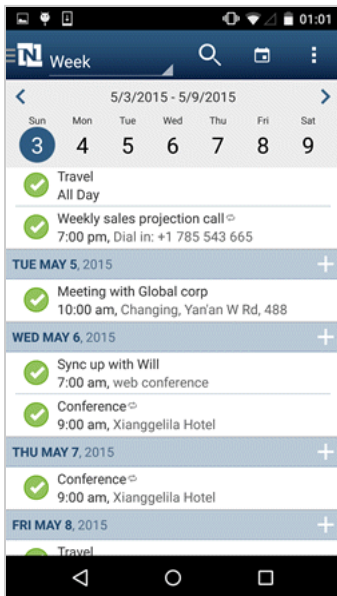
By default, the user is directed to the Recents screen. This screen displays the six most recently viewed records with an option to tap More for a complete list of recent records. The Recents screen also includes a Frequently Used Lists section, displaying the most used record types.

Customization Support

This version provides complete access to custom records including support for view, edit, create and delete (for any custom records already accessible via the NetSuite web application). You can also access Saved Searches based on custom records, and search for custom records using the Global Search feature.

Scheduling via Built-in Calendar

NetSuite for Android 2.0 introduces two mobile-optimized calendar views, Week and List. These views enable you to check your NetSuite activities directly within the application. Users can create and accept new activities, as well as customize which calendar to display based on an entity, group, or resource. Users can also filter the type of activity to display based on events, phone calls, and tasks.




Global Search

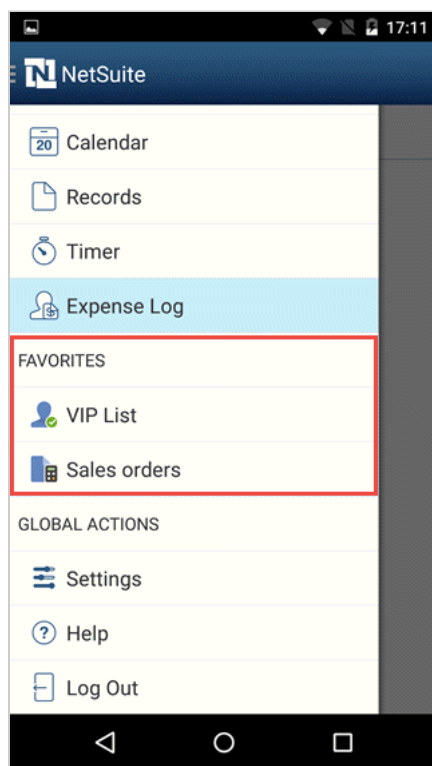
Search is the most commonly used way of accessing a record. NetSuite for Android 2.0 enables you to perform a Global Search using the icon in the tool bar. Search features include:

- default search of all record types
- record type prefixes such as **cust**

- predictive results after three characters
- quick look search history of the past ten searches

Add to Favorites

Users can create local Favorites to quickly access specific Lists. The Add to Favorites option is available under the Actions menu . If the user taps on Add to Favorites, a pop-up enabling them to define the name of the Favorite is displayed. The name of the Favorite should be pre-filled, but depends on the options currently applied:



Projects

Version 2015 Release 2 includes the following enhancements to Projects features:

- Limit Time and Expenses for Project Tasks
- Project Template Conversion Enhancements
- Allocate Resources to Project Tasks with Resource Allocations
- Job Costing Enhancements

- [Charge-Based Billing Enhancements](#)

Limit Time and Expenses for Project Tasks

On project records, disabling the Limit Time and Expenses preference now also applies to project tasks. Previously, when the preference was disabled, all resources could track time only against the project as a whole but not to individual project tasks. With Version 15 Release 2, disabling the preference now allows all resources to track time against both projects and project tasks.

For more information, see the help topic [Creating a Project Record](#).

Project Template Conversion Enhancements

With Version 15 Release 2, enhancements to project templates have been made to improve the speed and efficiency of conversion to and from a project record. When converting between a project and project template, editing of the project or template is not available until the conversion is complete. Each conversion is completed in the background, and you may continue to work in other areas of NetSuite. A message is displayed alerting you that some actions within the project or template are not available.

For more information, see the help topic [Creating Project Templates](#).

Allocate Resources to Project Tasks with Resource Allocations

With the Resource Allocations feature, you can now allocate resources directly to project tasks. Previously, project resources were first allocated to the project and then assigned to individual tasks.

Note: For each project, it is recommended that you choose to either allocate resources directly to project tasks or allocate resources to the project and then assign to project tasks. Using both resource allocations to project tasks and task assignments can create inconsistencies in your project data.

When creating a new resource allocation in NetSuite, after you select a project in the Customer:Project field, the Project Task field is populated with all available tasks from the selected project. Selecting a specific project task allocates the chosen resource to that task within the project. Leaving the Project Task field blank allocates the resource to the selected project.

On the project task record, there is now a Resources subtab. The Resources subtab lists all the project resources allocated to the task. The Estimated Work field is updated for both projects

and project tasks based on the allocated time for each resource. You can create new resource allocations from the Resources subtab by clicking New Resource Allocation.

Resources allocated to individual tasks can track time against those project tasks. If an allocation is deleted, the resource is no longer able to track any additional time against those project tasks.

There are also two new fields on both project and project task records—Allocated Work and Percent Complete by Allocated Work. The Allocated Work field displays the total number of hours allocated to the task and the project. The Percent Complete by Allocated Work field displays the progress of the project based on the allocated resources. This field can be useful in showing when projects require more effort than was originally planned.

Job Costing Enhancements

The following improvements have been made to the Job Costing feature.

- [Budget vs. Actual Report Enhancements](#)
- [Post Time Across Subsidiaries](#)
- [Posting Time Enhancements](#)
- [Post Time to a Selected Period](#)

Budget vs. Actual Report Enhancements

The Project Cost Budget vs. Actual report has a new Committed Cost column. This column displays project costs that have been committed but have not yet been resolved. For example, a project may include rental agreement with a contract to pay \$100 per month for equipment necessary to complete the project. Each month's rent for the life of the project would be a committed cost.

The Project Billing Budget vs. Actual report has two new columns—Committed Revenue and Recognized Revenue. The Committed Revenue column displays project revenue that has been committed but not yet received. For example, a project may include a service agreement with a contract for the customer to pay a service fee of \$300 per quarter for a specified amount of time. The revenue for future service fees would appear in this column. The Recognized Revenue column displays the cumulative dollar amount of the revenue recognized with either invoices or revenue commitments.

Post Time Across Subsidiaries

In NetSuite OneWorld, you can now post time across subsidiaries. NetSuite roles with the Post Time permission can post time to any subsidiary selected in the Subsidiary field on the role record. If no subsidiaries are selected on the role record, only the user's assigned subsidiary is available for posting time.

For example, if your company posts time at the parent company level for all subsidiaries, you can create a custom accountant role with the Post Time permission and all subsidiaries selected in the Subsidiary field on the role record. You can then assign that custom role to your accountant. When your accountant logs in with the new custom role to post time, NetSuite displays time available for posting from all subsidiaries.

If the selected Job Cost Variance account is not available for one of the subsidiaries, NetSuite displays an error message and no time is posted for that subsidiary.

Posting Time Enhancements

Version 15 Release 2 provides several enhancements to posting time. These enhancements offer better visibility and tracking of time transactions and journal entries, as they relate to job costing. On time transactions, a new Posted box is available. This field can be marked manually or automatically by posting time. The Posted field is visible only to roles with the Post Time permission.

Note: Manually marking a time transaction as posted should be used only for time transactions that have been previously accounted for and then imported into NetSuite. Any new time transactions created in NetSuite should be posted through the Post Time page at Transactions > Financial > Post Time.

The Posted field is displayed on approved time entries in edit mode when the time is associated with a project. The field is available for customization, SuiteScript, searching, and SuiteFlow. Checking the Posted box changes the status of the time transaction to Posted. The transaction is no longer available for posting and it appears on the Void Time page. Manually checking the Posted box on a time transaction does not create a corresponding journal entry. Voiding a manually checked time transaction clears the Posted box.

When you use Charge-Based Billing, time transactions associated with charges can no longer be edited. With Time Tracking, there is a new Mark as Posted button available in view mode to manually mark a time transaction with charges as posted. You can click the Void button on manually posted time transactions to clear the Posted box.

Important: The Mark as Posted and Void buttons are available only with Time Tracking. These buttons are not displayed when using Timesheets.

In addition to manually posting, you can now search journal entries to determine which entries were created by posting time transactions. Go to Transactions > Financial > Make Journal Entries > Search. In the Filter field, select Time. A popup window appears. In the Time field, select none of. Click the arrow button next to the multiple select field, and select -None-. Click Set. Click Submit to run the search. NetSuite will return a list of all journal entries that were created from time transactions.

Post Time to a Selected Period

When posting time for job costing, you can now select the accounting period. All open accounting periods are available in the Posting Period field. Each journal entry created is dated with the date of posting. Depending on your company's accounting settings, it may be possible to post time for transactions outside the selected period.

Note: If your company uses the Require Approvals on Journal Entries preference, the Posting Period field is not available.

To post time for job costing in a selected period, go to Transactions > Financial > Post Time. In the Posting Period field, select the accounting period you want to post time to. Check the box next to each time transaction you want to post. When you have finished, click Submit.

Charge-Based Billing Enhancements

The following improvements have been made to the Charge-Based Billing feature.

- [Billing Caps for Split Time](#)
- [New Employee Field on Charges](#)
- [Recurring Fixed Date Charge Rules](#)

Billing Caps for Split Time

Time-based charge rules can now split time transactions when fulfilling billing caps. Previously, a time-based billing cap required the exact amount to be tracked to fulfil a billing cap. Billing caps for time-based rules are honored and do not require an exact match from time transactions.

For example, you create a time-based charge rule with a billing cap of 10 hours. Your project resource tracks eight hours and six hours toward the project on two consecutive days. When you generate charges, the rule will generate charges for the eight hours plus two additional hours from the second day. The remaining four hours from the second day are listed on the time transaction as remaining time. If additional time-based rules are used to generate charges, the remaining four hours will be used to fulfill any new rules.

When a time transaction has time remaining or generates charges from multiple rules, a new Charges subtab is displayed with a list of the corresponding charges.

New Employee Field on Charges

For time-based and expense-based charges, an Employee field is now available on charge records. The Employee field automatically shows the employee generating the time or expense

transaction. The Memo field is also automatically populated from the corresponding time entry or expense line generating the charge.

Recurring Fixed Date Charge Rules

You can now set a recurrence schedule for fixed date charge rules. When creating a new fixed date rule, you can select to have the rule be a single, daily, weekly, monthly, or yearly event. Within each selection there are additional choices you can make to define the recurrence schedule.

You can also set both a start and end date for the series. Previously, the Date field was required when creating a fixed date rule. Now, the Date field has been replaced by the Series Start Date field. Any scripts or workflows utilizing the Date field will be transferred to the Series Start Date field.

When generating a forecast for a recurring fixed date charge rule without an end date, NetSuite includes each occurrence through the end date of the project plus one occurrence. When generating actual charges for a recurring fixed date charge rule without an end date, NetSuite continues to generate charges as long as the project is active.

After charges have been generated, a recurring charge rule can no longer be edited. You may inactivate a recurring charge rule if it is no longer necessary. To inactivate a charge rule, click Inactivate at the top of the charge rule record. When you inactivate a recurring charge rule, all forecast charges based on the rule are deleted and the Inactive box appears checked on the charge rule record. You may reactivate an inactive recurring charge rule by clicking Activate at the top of the charge rule record.

Vendors, Purchasing, and Receiving

Version 2015 Release 2 includes the following enhancements to vendors, purchasing, and receiving features:

- [Billing Address on Purchasing Transactions](#)
- [Purchases/Vendors Dropdown Menu Changes](#)
- [Deleting Subrecords on Purchase Contracts and Blanket Purchase Orders](#)
- [Order Requisition Enhancements](#)
- [Shared Vendor Saved Searches Display Secondary Subsidiaries](#)
- [Procurement Dashboard](#)

Billing Address on Purchasing Transactions

In Version 2015 Release 2, you are able to enter billing addresses on the following transactions:

- Vendor Bill
- Vendor Credit
- Vendor Return Authorization

Purchases/Vendors Dropdown Menu Changes

In Version 2015 Release 2, the Purchases/Vendors menu has been split into two menus: Purchases and Payables.

- The Payables menu is organized for the AP Clerk role.
- The Purchases menu is organized for a new role called Buyer.

The links available under each heading are as follows:

Purchases

Enter Purchase Orders

Approve Purchase Orders

Enter Requisitions

Approve Requisitions

Enter Request for Quote

Enter Purchase Contract

Approve Purchase Contract

Enter Blanket Purchase Orders

Approve Blanket Purchase Orders

Order Requisitions

Order Items

Receive Orders

Enter Vendor Ret Authorizations

Approve Vendor Returns

Ship Vendor Returns

Payables

Bill Purchase Orders

Enter Bills

Approve Bills

Pay Bills

Pay Single Vendor

Post Vendor Bill Variances

View Vendor Payment Status

Approve Vendor Payment Transfers

Enter Vendor Credits

Credit Vendor Returns

Buyer Role

The Buyer role is available when you enable the Accounting feature and has the following permissions.

Permission	Level
Find Transaction	Full
Purchase Order	Full
Requisition	Edit
Request for Quote	Full
Purchase Contracts	Full
Blanket Purchase Orders	Full
Requisition Approval	Full
Purchase Contract Approval	Full
Blanket Purchase Order Approval	Full
Receive Orders	Edit
Item Receipt	Edit
Vendor Return Authorizations	Full
Vendor Return Approvals	Full
Vendor Returns	Edit
Bill Purchase Orders	Edit

Bills	Edit
-------	------

Deleting Subrecords on Purchase Contracts and Blanket Purchase Orders

Using Version 2015 Release 2, blanket purchase orders and purchase contracts display a Delete button on line items, enabling you to delete the associated subrecord lines.

Order Requisition Enhancements

Previously, the Order Requisitions page grouped requisition lines on a purchase order based on all requisition lines having the same transaction date.

Now, on the Order Requisitions page, you can add multiple order lines to a purchase order regardless of the entry date or expected receipt date. The number of lines consolidated onto a single purchase order is limited only by the maximum identified in the Maximum Requisition Lines to Consolidate preference.

Note: In Version 2015 Release 2, the Maximum Requisition Lines to Consolidate preference has been renamed Maximum Purchase Lines to Consolidate.

Maximum Purchase Lines to Consolidate Preference

Enter a value in the Maximum Purchase Lines to Consolidate field to define the maximum number of lines to consolidate on purchase orders. This definition applies in the following cases:

- when you submit multiple orders on the Order Requisitions page and items from one or more requisitions are combined on a single purchase order.
- when you submit multiple orders on the Order Items page and items from one or more planned orders are combined on a single purchase order.

This value defaults to 10 but you can enter a number up to 250.

If you leave this field blank, the following is true:

- On the Order Requisitions page, all requisition lines for a vendor are consolidated into a single purchase order.
- On the Order Items page, all planned order lines for a vendor are consolidated into a single purchase order.

Preferences can be set at Setup > Accounting > Accounting Preferences, by a user who has the Set Up Accounting permission.

Order Requisitions Page

On the Order Requisitions page, a Purchase Contract field has been added. If a qualified contract exists for the selected vendor, the purchase contract with the lowest item base rate is displayed in this field. The purchase contract's base rate is displayed in the Order Rate field.

Note: Enter the vendor as the preferred vendor on the item record if you want a vendor/purchase contract combination to show automatically in the Order Items page.

Using the Order Requisitions page requires that you have enabled the Requisitions feature in your account.

Shared Vendor Saved Searches Display Secondary Subsidiaries

If you share vendor records with multiple subsidiaries, your saved vendor searches list all of the secondary subsidiaries that are assigned to the searched vendor record. For information about shared vendor records, see the help topic [Sharing Vendors with Multiple Subsidiaries](#). For information about saved searches, see the help topic [Using Saved Searches](#).

Taxation

Version 2015 Release 2 includes the following enhancements to taxation features:

- [Enhancements to Certificate of Creditable Tax Withheld at Source \(Form 2307\)](#)
- [EU Sales List Support for Triangulation of Goods](#)

User Interface

Version 2015 Release 2 includes the following enhancement to the NetSuite user interface (UI):

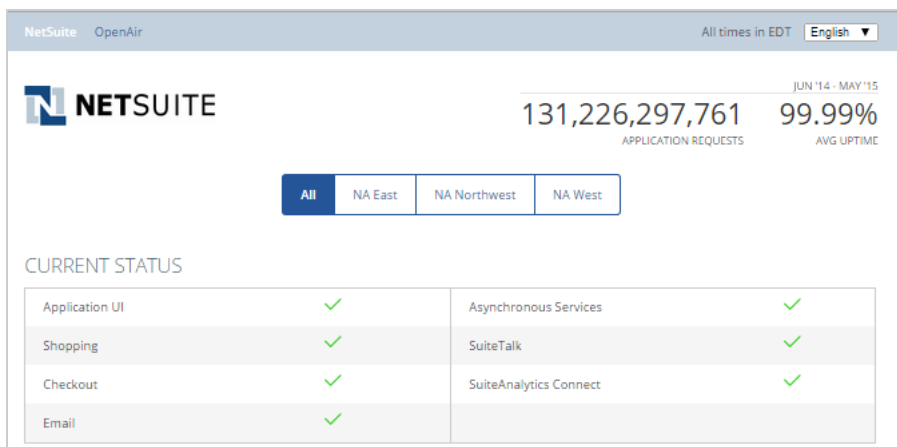
- [New NetSuite Status Page](#)
- [New User Preference Enhances Accessibility](#)
- [Credit Card Processing User Interface Elements Relabeled](#)
- [Support for Internet Explorer 9 Ended](#)
- [Support for Safari 5 Ended](#)
- [StickyNotes User Experience Improvements](#)
- [Basic API Integration on SuiteSocial News Feed](#)
- [Email Capture Plugin Support for SuiteSocial](#)

- Discontinued Japanese Custom Forms

New NetSuite Status Page

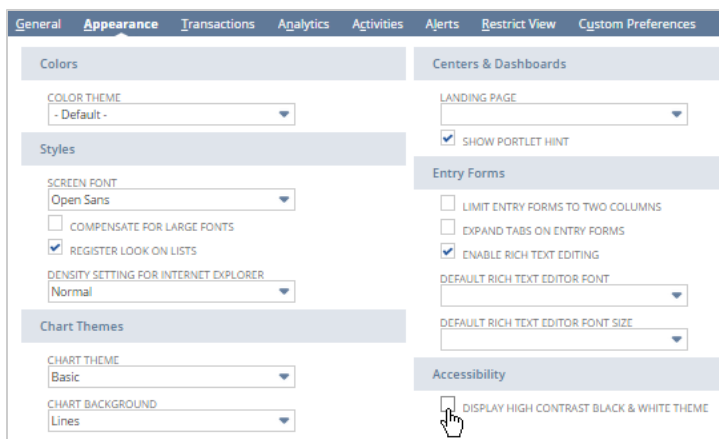
NetSuite provides a site, <http://status.netsuite.com>, where you can check the system status, even if the NetSuite UI is experiencing a service interruption or downtime.

The NetSuite Status page has a new look and provides clear information on the status of the NetSuite UI, shopping and checkout websites, Email, SuiteTalk web services, and SuiteAnalytics Connect. The new page provides you with current status historical data for various NetSuite services. See the help topic [Checking NetSuite System Status](#) for more information.



New User Preference Enhances Accessibility

A new user preference for accessibility allows users to enable a theme with optimal color contrast. The new preference is on the Appearance tab under Accessibility, accessed from the Set Preferences link on the dashboard. See the help topic [Setting Preferences for Appearance](#).



Credit Card Processing User Interface Elements Relabeled

The behavior of some credit card processing user interface elements has been extended to permit you to use electronic non-credit card payment methods. Examples of electronic non-credit card payment methods include online bank transfers and direct debits. Therefore, some pages and fields in the NetSuite UI have been relabeled:

- The Credit Card Processing page (at Setup > Accounting > Credit Card Processing) has been renamed Payment Processing Profiles (at Setup > Accounting > Payment Processing Profiles).
- The label of the Credit Card Processor field (ID: creditcardprocessor) has been changed to Payment Processing Profile on all transaction records that contain this field.

Note: The field internal ID has not changed.

Support for Internet Explorer 9 Ended

As of Version 2015 Release 2, NetSuite will not be tested with Internet Explorer 9 (IE9), and support will not be offered for defects specific to this browser.

Support for Safari 5 Ended

As of Version 2015 Release 2, NetSuite will not be tested with Safari 5, and support will not be offered for defects specific to this browser.

General SuiteCommerce Settings

Version 2015 Release 2 includes the following enhancement to general SuiteCommerce Settings:

Asynchronous afterSubmit User Event Support for Web Store Users

NetSuite Version 2015 Release 2 adds asynchronous afterSubmit user event support for web store users. When this feature is enabled, afterSubmit user events are triggered asynchronously when a sales order is created during web store checkout. When an afterSubmit user event runs in asynchronous mode, the next page is loaded during the period of time the afterSubmit user event is still processing. Web store Sales Orders process afterSubmit scripts in the background. This feature decreases the time needed to submit an online order.

To enable asynchronous mode, go to Setup > Enable Features > Web Presence subtab. Check the box next to Asynchronous AfterSubmit Sales Order Processing. Note that if an afterSubmit

script is running asynchronously, and the script includes a call to `nlapiRedirectURL`, the script will ignore the call.

Important: This is a global setting. This setting causes all `afterSubmit` user events and workflows triggered by web store checkout to run asynchronously.

SuiteCommerce InStore

Version 2015 Release 2 includes the following new feature:

Introducing SuiteCommerce InStore

Important: This feature is targeted to be available in September 2015.

SuiteCommerce InStore (SCIS) is a web-based, point-of-sale application available for US markets. SCIS is built on the same platform as SuiteCommerce Advanced, and provides an intuitive user interface designed for touch tablet screens. Contact your account manager for information about purchasing SCIS.

SCIS is delivered as a managed SuiteApp. Updates associated with future releases will be pushed directly to your account. During installation, you create a website that employees use to access the SuiteCommerce InStore point-of-sale application. After installation is complete, employees log in to that website to enter point-of-sale transactions.

Interaction with NetSuite

Data generated from SCIS is saved on standard records and custom records in NetSuite. Custom records associated with SCIS are installed by the SuiteApp. You must configure standard records and custom records as part of installation.

Sales associates use the SCIS website to view customer profiles, enter transactions, and to perform returns and exchanges. NetSuite administrators can track daily operations using lists and reports available in NetSuite. After a transaction is submitted through SCIS, a NetSuite administrator can view or modify the transaction in NetSuite. Saved searches for transaction and sales reports are installed by the SuiteApp.

Customer Profile

The Customer Profile includes contact information, sales graphs, and a transaction time line. If you already have a SuiteCommerce Advanced website, and you install SCIS, online customers' purchase history is accessible to sales associates through the Customer Profile. In this way, a sales associate gets a complete view of a customer's interactions with your organization. In addition to customer purchasing history, the Customer Profile includes a time line that shows

returns, shipments, and dates of interactions with your call center. To access the Customer Profile and time line, the sales associate can search for a customer in SCIS, and then click the customer's name.

SuiteCommerce InStore Capabilities

SCIS also includes the following capabilities:

- Cash and carry
- Orders for delivery
- Exchanges
- Returns
- Voiding lines
- Discounts and Promotions
- Gift Cards
- Initial loan and drawer drop
- Product recommendations and reviews (additional SuiteApp installation is required)

For more information, read the *SCIS Administrator's Guide*.

SuiteCommerce Advanced

Version 2015 Release 2 includes the following enhancements:

- [Cache Invalidation by URL](#)
- [Reference Implementation Release Notes](#)

Cache Invalidation by URL

Prior to Version 2015 Release 2, website developers and NetSuite Administrators could use the Clear Cache link only to view updated website content. Now, you can use the Cache Invalidation by URL page to send cache invalidation requests using specific URLs.

The majority of data is cached in SSP and SuiteScript method calls, such as the method calls that exist in the `sc.environment.ssp` file. To effectively clear the CDN cache, you must clear the cache using a URL that points to the file that includes those methods. On the Cache Invalidation by URL page, you can enter the website domain, and then enter a URL that includes the updated file and applicable parameters.

Note: You cannot clear CDN cache by using a URL that points to a product in the web store, such as **http://www.mywebstore.com/blue-hat**. You must use a URL that points to an SSP file or SuiteScript services file.

For example, you may have a banner on your website that you plan to change each day. Because most data is cached in SSP and SuiteScript method calls, you must submit a cache invalidation request using the URL for the SSP file that includes a definition for the banner on your site for example, **/page.ss**. Do this every day you have an update to the banner. On the Cache Invalidation by URL page, you would send the cache invalidation request using your website domain along with a URL similar to the example below:

```
http://www.mywebsite.com/c.7654321/cds/services/page.ss?internalid=123&cache=short
```

For more information about using the Cache Invalidation by URL page, read the help topic [Cache Invalidation by URL](#).

SuiteBundler

Version 2015 Release 2 includes the following enhancements to SuiteBundler features:

- [Bundle Support During Release Phasing](#)
- [Support for Managed Upgrades in Release Preview Accounts](#)

Bundle Support During Release Phasing

Bundle support during phasing of Version 2015 Release 2 is consistent with the support provided during previous releases. Review the description below for a reminder of how bundles from different versions are handled during release phasing.

During phasing of Version 2015 Release 2, some accounts continue to use Version 2015 Release 1 during the time that other accounts are upgraded to Version 2015 Release 2. Therefore, you may be using a version different from some of the accounts where your bundles are installed.

- Bundles developed with Version 2015 Release 1 can be installed into accounts that are already using Version 2015 Release 2.
- Bundles developed with Version 2015 Release 2 can be installed into accounts that are still using Version 2015 Release 1, but note the following limitations:
 - When a user in a Version 2015 Release 1 account installs a bundle from a Version 2015 Release 2 account, any object types in the bundle that are not supported in Version 2015 Release 1 are not installed. This may result in a potentially serious impact to bundle function.

- A bundle with a SuiteScript that uses a new API available only in Version 2015 Release 2 can be installed into a Version 2015 Release 1 account, but the script may not correctly function because Version 2015 Release 1 does not support the API.
- A bundle with a script developed with SuiteScript 2.0 in Version 2015 Release 2 cannot be installed into a Version 2015 Release 1 account.

Warning: The bundles that can be installed into a Release Preview account are only those created in another Release Preview account. A bundle originally installed from a production account or the production repository can be updated in a Release Preview account if the original account or the repository is also available in Release Preview. A bundle originally installed from a sandbox account cannot be accessed or updated in a Release Preview account. Also, any bundle created in a Release Preview account cannot be installed into a production or sandbox environment, even if it has been copied to the bundle repository.

Support for Managed Upgrades in Release Preview Accounts

As of Version 2015 Release 2, you can upgrade a managed bundle from one Release Preview account to another Release Preview account. Use this feature to test managed bundle upgrades in the Release Preview environment.

In previous releases, you could only upgrade managed bundles from one production account to another production account.

For more information about upgrading managed bundles, see the help topic [Upgrading Your Managed Bundle Install Base](#).

SuiteBuilder - Customization

Version 2015 Release 2 includes the following enhancements to SuiteBuilder features:

- [Hierarchical Custom Record Types](#)
- [Ability to Make Custom Fields Inactive](#)
- [Preview of Advanced PDF/HTML Template Output](#)
- [Custom Segments \(Beta\)](#)
- [Enhanced Options for Deleting Custom Transaction Types](#)
- [Changes to the Japanese Custom Address Form and Zip Code Lookup](#)

Hierarchical Custom Record Types

Version 2015 Release 2 supports hierarchies between records of the same custom record type. A new **Hierarchy** option is available on the custom record type definition page. When this

option is enabled, you can define a parent-child hierarchy between two records of that type. This hierarchy can be defined either on the parent record or on the child records.

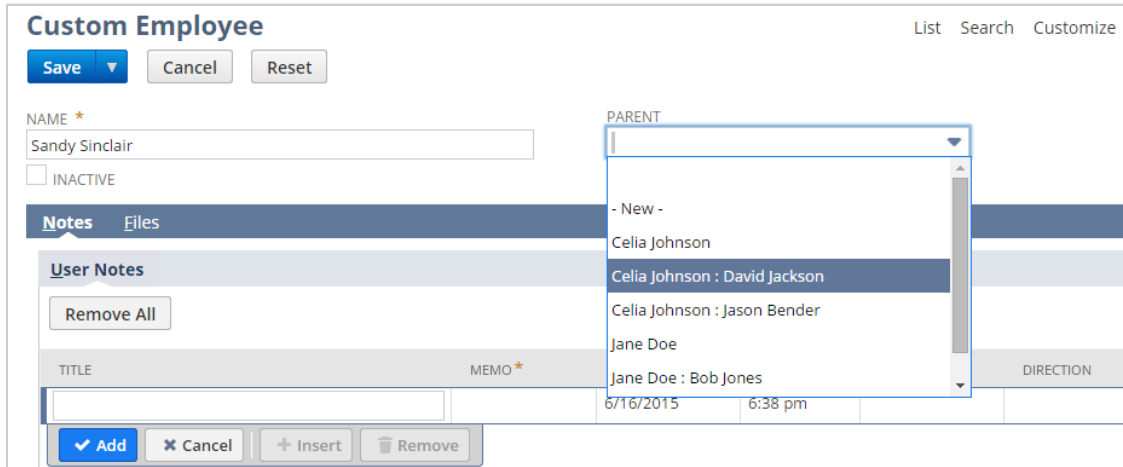
For example, if you have defined a custom record type that stores employee information, you can set up hierarchies among records of this type to reflect each employee's supervisor. First, you must enable the **Hierarchy** option for the custom employee record type.

The screenshot shows the 'Custom Record Type' configuration interface for 'Custom Employee'. The 'HIERARCHY' checkbox is checked and highlighted with a red box. Other visible options include 'ALLOW QUICK SEARCH', 'ALLOW QUICK ADD', 'ENABLE SYSTEM NOTES', 'INCLUDE IN GLOBAL SEARCH', 'INCLUDE IN SEARCH MENU', 'ENABLE OPTIMISTIC LOCKING', 'ENABLE INLINE EDITING', 'ENABLE NAME TRANSLATION', and 'INACTIVE'. The 'SHOW OWNER' and 'SHOW LAST MODIFIED' options are also visible.

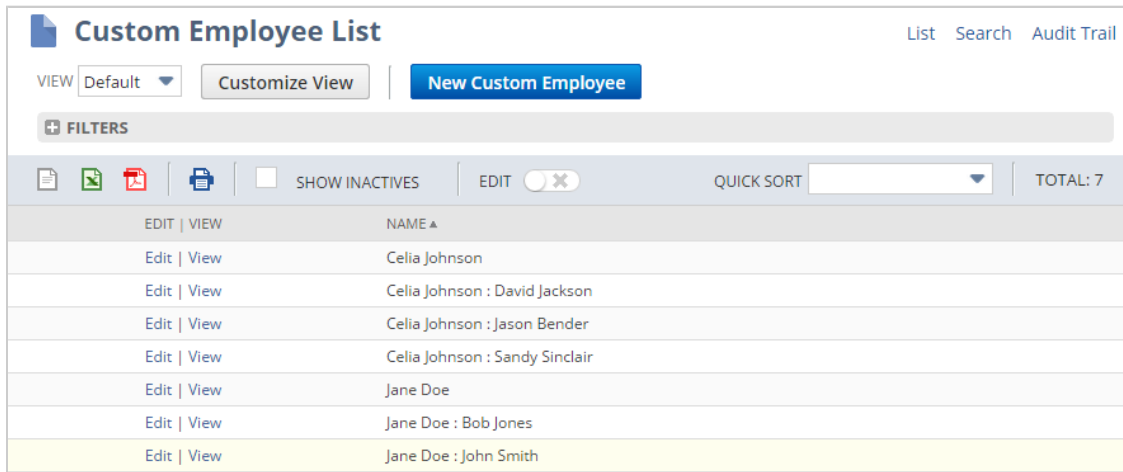
Then, on a parent record, you can click the **New Custom Employee** button on the Child Records subtab to create a child record.

The screenshot shows the 'Custom Employee' record page for 'Celia Johnson'. The 'Child Records' subtab is active, and the 'New Custom Employee' button is highlighted with a red box. The page displays a list of child records with columns for 'EDIT' and 'NAME'. The child records listed are 'Celia Johnson : David Jackson', 'Celia Johnson : Jason Bender', and 'Celia Johnson : Sandy Sinclair'.

Alternatively, on a child record, you can select a parent record from the dropdown list, or create a new record to be the parent.



The list page for a custom record type with hierarchy enabled displays child records in the format <parent record>:<child record>.



Important: Although this example illustrates only two levels of hierarchy, this feature supports multiple levels of hierarchy among custom record values.

Ability to Make Custom Fields Inactive

Before this release, it was possible to remove a custom field from a specific record type, or to delete the field completely. Version 2015 Release 2 supports the ability to make a custom field

inactive instead of deleting it. A new **Inactive** check box now appears on each custom field record.

The screenshot shows the 'Custom Entity Field' configuration interface. At the top, there are buttons for 'Save', 'Cancel', 'Reset', 'Change ID', 'Download XML', and 'Apply to Forms'. Below these are input fields for 'LABEL *' (containing 'Reason for Termination'), 'DESCRIPTION', 'PACKAGE', 'ID' (containing 'custentity1'), and 'OWNER' (containing 'Cindy Sloan'). There are also dropdown menus for 'TYPE' (set to 'Free-Form Text') and 'LIST/RECORD'. A 'STORE VALUE' checkbox is checked, and 'USE ENCRYPTED FORMAT' is unchecked. On the right side, under 'Actions', there are checkboxes for 'SHOW IN LIST', 'GLOBAL SEARCH', 'RECORD IS PARENT', and 'INACTIVE' (which is highlighted with a red box). At the bottom, there is a navigation bar with tabs for 'Applies To', 'Display', 'Validation & Defaulting', 'Sourcing & Filtering', 'Access', 'Translation', and 'History'. The 'Applies To' section contains a grid of checkboxes for various record types: CUSTOMER, JOB, VENDOR, EMPLOYEE (checked), OTHER NAME, CONTACT, PARTNER, GENERIC RESOURCE, WEB SITE, and GROUP. There are also checkboxes for 'AVAILABLE EXTERNALLY', 'PRINT ON STATEMENT', 'PRINT ON PRICE LIST', and 'PROJECT TEMPLATE'.

Also as part of this feature, a new **Show Inactives** option is available on all custom field list pages, including the list of custom fields on each custom record type's Fields subtab. By default, the **Show Inactives** option is disabled so that inactive custom fields are filtered out of lists. When this option is enabled, the list displays both inactive and active custom fields, and each custom field in the list has an **Inactive** check box next to it. You can check this box to make a custom field inactive, and clear this box to reactivate an inactive custom field.

When a custom field is made inactive, it no longer appears on any forms or reports and it is not returned by global search, just like a deleted field. However, data and associated forms for an inactive custom field are maintained in NetSuite, whereas a deleted field is completely removed from the system. If an inactive field is later made active again, all of its data is restored, and the field appears on all of the same forms as before it was made inactive. Note that some settings are not maintained for inactive fields, including display formatting.

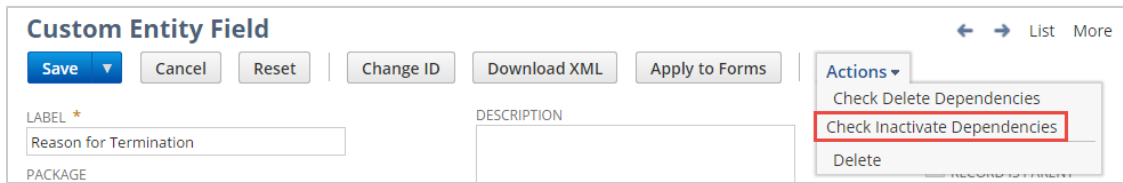
A custom field cannot be made inactive if other NetSuite records depend on it. For example, a custom field cannot be made inactive if it meets any of the following conditions:

- It is sourced by another custom field.
- It is used in lead conversion mapping.
- It is used in the Parameter, Values section for a workflow action.
- It is used in a workflow definition condition.
- It is used in a filter or a condition for a saved search.

In addition to the above examples, there are other dependencies that may prevent you from making a custom field inactive.

When you try to make a custom field that has dependencies inactive, an error message is returned with a link to a Dependent Records page where you can review details about these dependencies.

You can use a new option under the Actions menu on each custom field record to proactively check for dependencies before you try to make the custom field inactive. Another option is available to check dependencies that may prevent you from deleting a field.

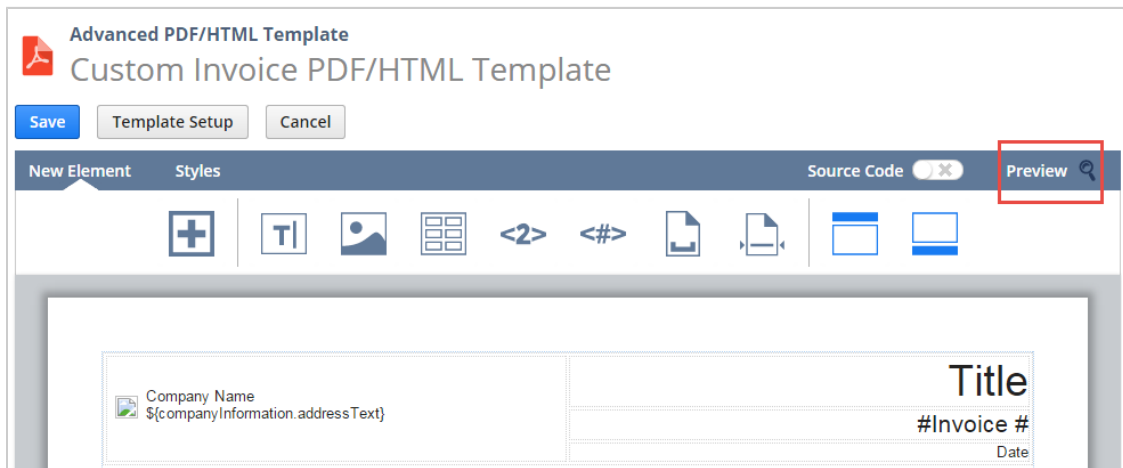


When you do one of these checks, a Dependent Records page appears. If this list has no records, you can make the custom field inactive or delete it without errors.

Warning: Script custom fields cannot be made inactive.

Preview of Advanced PDF/HTML Template Output

As of Version 2015 Release 2, the advanced PDF/HTML template editor supports preview of template output. Before this release, if you wanted to view template output or check for template errors, you had to save a template, associate it with a custom form, associate the custom form with a transaction, then print the transaction. Now, as you are editing a template, you can click the **Preview** button to generate a sample PDF based on the current template definition. The **Preview** button is available in both WYSIWYG and source code modes.

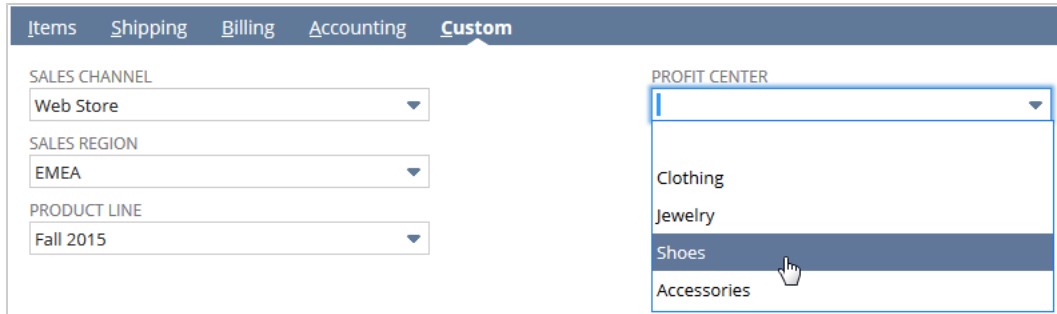


This preview capability enables you to see how printed transactions based on a template will appear. Previewing is also a quick way to check for errors. If the sample PDF successfully generates, it is likely that transactions using this template will successfully print in production. If an error occurs when you click the **Preview** button, you can review the error message and try to fix the issue. Note that preview PDFs use simulated data.

One World 2955 Campus Drive San Mateo CA 94403 US		Invoice #123456789012345 06/22/2015			
Bill To Sweet Dreams, Inc. 1001 Cloud Street San Francisco CA 94112	Ship To Sweet Dreams, Inc. 1001 Cloud Street San Francisco CA 94112	TOTAL \$9,999.99 Due Date: 06/22/2015			
Terms	Due Date	PO #	Sales Rep	Shipping Method	Partner
Net 30	06/22/2015	123456789012345	US Cust	FedEx	US Cust
Quantity	Item	Options	Rate	Amount	
9,999.99	FedEx Lorem ipsum dolor sit amet consectetur ac orci sociis ornare laoreet.		\$9,999.99	\$9,999.99	
9,999.99	FedEx Lorem ipsum dolor sit amet consectetur ac orci sociis ornare laoreet.		\$9,999.99	\$9,999.99	
9,999.99	FedEx Lorem ipsum dolor sit amet consectetur ac orci sociis ornare laoreet.		\$9,999.99	\$9,999.99	
9,999.99	FedEx Lorem ipsum dolor sit amet consectetur ac orci sociis ornare laoreet.		\$9,999.99	\$9,999.99	
9,999.99	FedEx Lorem ipsum dolor sit amet consectetur ac orci sociis ornare laoreet.		\$9,999.99	\$9,999.99	
9,999.99	FedEx Lorem ipsum dolor sit amet consectetur ac orci sociis ornare laoreet.		\$9,999.99	\$9,999.99	
9,999.99	FedEx Lorem ipsum dolor sit amet consectetur ac orci sociis ornare laoreet.		\$9,999.99	\$9,999.99	
9,999.99	FedEx Lorem ipsum dolor sit amet consectetur ac orci sociis ornare laoreet.		\$9,999.99	\$9,999.99	
				Subtotal	\$9,999.99
				Tax (%)	\$9,999.99

Custom Segments (Beta)

The Custom Segments feature lets you create custom classification fields similar to class, department, and location. You can create an unlimited number of custom segments, define possible values for each segment, and add the segments to specific record types. People working in NetSuite can then use the segments to classify records appropriately.



You can configure segments to appear on the GL Impact page. Additionally, standard NetSuite reports can be customized to use segments as filters and columns. When a segment is used as a column, it can also be grouped with other columns. For example, the following screenshot shows a customized version of the Sales by Item report. This report groups two segments in the column at the left.

Sales by Audience and Profit Center View Detail		TOTAL REVENUE	SALES CHANNEL	SALES REGION
PROFIT CENTER	QTY. SOLD	ITEM NAME		
- Women's				
- Shoes				
	1	Ballet Flat	\$65.00 Web Store	Madrid
	2	Jeweled Sandal	\$90.00 Brick-and-Mortar Stores	New York
	1	Patent Heel	\$452.00 Partner Outlet	Paris
	1	Slingback Sandal	\$115.00 Web Store	Sao Paulo
Total - Shoes		5	\$722.00	
+ Accessories		16	\$2,575.00	
+ Clothing		141	\$12,156.00	
+ Outerwear		46	\$5,519.00	
+ Jewelry		16	\$11,048.00	
Total - Women's		224	\$32,020.00	
- Men's				
+ Outerwear		28	\$6,154.00	
+ Clothing		36	\$13,852.61	
Total - Men's		64	\$20,006.61	
- Unisex				
+ Accessories		20	\$1,270.00	
Total - Unisex		20	\$1,270.00	
Total		308	\$53,296.61	

In these ways and more, custom segments enhance your ability to categorize data and meet your organization's unique reporting and analysis needs.

Important: Custom segments are currently available as a beta feature.

Before you can create custom segments, you must enable the feature. Navigate to Setup > Company > Enable Features. On the SuiteCloud subtab, select the Custom Segments (Beta) box and click Save.

By default, only account administrators and users with the Full Access role have permission to create and edit custom segments. However, you can assign this permission to other roles as needed. For details, see the help topic [Granting a Role Permission to Work With Custom Segments](#).

Authorized users can create custom segments at Customization > Lists, Records, & Fields > Custom Segments > New. Almost all segment-configuration options can be managed from this page, including the following:

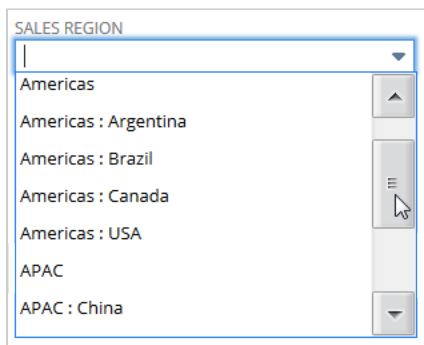
- [Configure Segment Values to Be Hierarchical](#)
- [Configure a Segment to Default Statically or Dynamically](#)
- [Filter a Segment's Values Based on Other Segments](#)
- [Filter a Segment's Values Based on Class, Department, Location, and Subsidiary](#)
- [Configure a Segment to Appear on the GL Impact Page](#)
- [Create Segments as Multi-Select Fields](#)
- [Display Segments Selectively](#)

Custom segments are part of the SuiteGL feature set. For more on SuiteGL, see the help topic [Getting Help on SuiteGL Features](#).

Important: Custom segments cannot be bundled. Additionally, objects that reference segments cannot be successfully bundled. These objects can be added to a bundle, but their inclusion will cause installation or update of the bundle to fail.

Configure Segment Values to Be Hierarchical

In the list of available segment values, you can create hierarchal relationships. For example, a segment called Sales Region might have values such as Americas, APAC, and EMEA. However, you may also want to add options that are more specific. When you create the new options, you can specify in the segment definition that they are children of the broader geographic categories. These relationships are shown in the list of values from which the user can choose.



Configure a Segment to Default Statically or Dynamically

If appropriate, you can configure a segment to default to a specific value. You have the following options:

- You can configure a segment to default to a static choice. This default is used on any record type where the segment appears, unless overridden. For example, suppose you have a segment called Sales Region, which includes the values Americas, EMEA, and APAC. If the majority of your records require a value of Americas, you could specify that Americas is the default.
- You can create dynamic defaulting for specific categories of record types, such as transactions and entities. For example, you may have a segment called Sales Region that you have applied both to the Customer record type and to all sales transactions. You can configure the segment so that, when it occurs on a sales transaction, it defaults to the value selected on the corresponding customer record. This type of defaulting overrides static defaulting.

Filter a Segment’s Values Based on Other Segments

In some cases, you may want to limit a segment’s available values based on choices the user made for other segments on the same record.

For example, suppose your organization sells merchandise worldwide. You may have multiple sales channels, such as brick-and-mortar stores, a web store, and partner outlets. However, although you may permit customers everywhere to use your web store, other channels may be available only in certain geographic regions.

In this case, you could create two segments, Sales Region and Sales Channel. You could configure a record’s Sales Channel values to be filtered based on the selection the user makes for Sales Region. For example, you could configure a Sales Region called Denmark to permit the selection of only two Sales Channel values.



You can also set up more complex filtering. You can configure a segment’s available values to be dependent on multiple segments and on other classification fields, as described in the next section.

Filter a Segment's Values Based on Class, Department, Location, and Subsidiary

You can set up a segment so that its values are filtered based on choices the user made in the Class, Department, Location, and Subsidiary fields. This capability is similar to the behavior described in [Filter a Segment's Values Based on Other Segments](#).

For example, consider a company that sells clothing and accessories. This company may have a segment called Profit Center, with values such as Clothing, Shoes, and Jewelry. If the company markets products to both men and women, it may also have departments called Men's and Women's.

Both the Men's and Women's departments may sell clothing and shoes, but jewelry might be available only for women. In this case, you could configure the Profit Center segment so that its values are filtered based on the value of the Department field. A record with a Department value of Men's would show only two of the three Profit Center values.

The screenshot shows a configuration window for a segment. Under the 'Classification' section, 'SUBSIDIARY' is set to 'Parent Company' and 'DEPARTMENT' is set to 'Men's'. Below this are tabs for 'Items', 'Shipping', 'Billing', and 'Accounting'. The 'PROFIT CENTER' dropdown menu is open, displaying a list of options: '- New -', 'Clothing', and 'Shoes'. A mouse cursor is hovering over the 'Clothing' option.

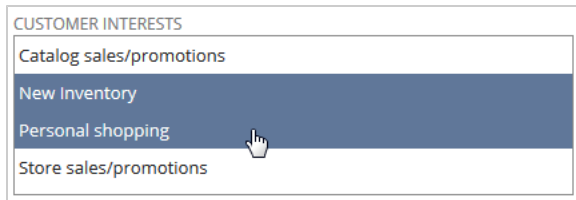
Configure a Segment to Appear on the GL Impact Page

In some cases, you may want to apply segments to transactions. You can apply segments to the body of a transaction or make them columns in a transaction sublist. In both cases, you can configure these segments so that their values appear on the GL Impact page for the transactions where they are used.

GL Impact											Return to Invoice #3	
ACCOUNT	AMOUNT (DEBIT)	AMOUNT (CREDIT)	POSTING	MEMO	NAME	SUBSIDIARY	DEPARTMENT	CLASS	PRODUCT LINE	PROFIT CENTER	SALES CHANNEL	SALES REGION
1100 Accounts Receivable	165.00		Yes		Anonymous Customer	Parent Company	Women's	Full-Priced	Winter 2015-16	Clothing	Web Store	Denmark
4000 Sales		165.00	Yes		Anonymous Customer	Parent Company	Women's	Full-Priced	Winter 2015-16	Clothing	Web Store	Denmark
											TOTAL: 2	

Create Segments as Multi-Select Fields

If appropriate, you can set up a segment so that the user can save multiple selections. This option differs from the Class, Department, and Location fields, where the user can select only one option.



Some limitations exist for this capability. For example, if a segment is configured to appear on the GL Impact page, it cannot be a multi-select field.

Display Segments Selectively

The Custom Segments feature includes a rich array of permissions options. One option is that you can choose to make a segment visible only to users in certain roles. For example, you might choose to create a segment called Rating that is applied to the Employee record type. This segment might contain sensitive information that should be available only to supervisors. In this case, you can configure the segment to be available to users in a Supervisor role but hidden for all other users. Additionally, you can choose to make the segment read-only for certain roles.

You also have many options for deciding how users can manage custom segments. For example, your account will probably have many roles that do not have permission to create or edit segments. However, if appropriate, you can give these roles permission to create and edit values for one particular segment. Note also that segment values can be created and updated by using the CSV Import Assistant. For details, see [CSV Support for Custom Segment Values](#).

Enhanced Options for Deleting Custom Transaction Types

Version 2015 Release 2 includes enhanced options for deleting custom transaction types that were installed as part of a bundle. In previous releases, you were prevented from deleting these transaction types if instances of the type existed. Now, the system permits deletion of the transaction type as part of a bundle removal or a bundle update.

These enhancements apply only to bundle operations. If a custom transaction type was created manually in the UI and a transaction instance exists, you cannot delete the transaction type.

When a bundle operation is used to delete a transaction type, the system takes the following actions with any existing instances of that transaction type:

- The transactions are migrated to a new transaction type called Deprecated Custom Transaction.
- The transactions are locked to editing, although they can still be voided.
- If the transactions are in open periods, they can be deleted.
- Body fields are added to the transaction showing:
 - The name of the custom transaction type that was deleted.
 - The status of the transaction at the time the type was deleted
- Classification fields on the body of the transaction are moved to the line level.
- Custom fields are discarded.

To see a list of transactions that were instances of deleted custom transaction types, use the Transaction search type. Set the Type field to Deprecated Transaction Type.

SuiteAnalytics (Dashboards, Reporting, and Search)

Version 2015 Release 2 includes the following enhancements in the SuiteAnalytics area:

- Tableau Export for Saved Searches and Reports
- Connect Browser
- More Options for Custom Fields in Summary Reports
- New Localization Preference for Search Sort Order
- Redesigned Report Results Page
- New Permission to Persist Search Results
- Labor Cost Now Available for Saved Searches
- Changes to the Left Side Field in Financial Reports and SuiteAnalytics Connect
- End of Support for the ODBC 6.0 Driver
- Pivot Reports BETA Replacement

Tableau Export for Saved Searches and Reports

Important: This feature is targeted to be available by September, 2015.

With Version 2015 Release 2, you can export the results of NetSuite saved searches and reports as analysis-ready Tableau® workbooks (*.twbx). You can use Tableau Desktop software to view and customize these exported results.

Two permissions are required for export of search and report results to Tableau: Export Lists and Tableau® Workbook Export.

Also, the Tableau® Workbook Export feature must be enabled in your account. To enable this feature, go to Setup > Company > Enable Features. On the Analytics subtab, select Tableau® Workbook Export and click Save.

Connect Browser

Version 2015 Release 2 includes the Connect Browser, a new tool that provides information about SuiteAnalytics Connect tables and columns.

The Connect Browser includes a summary of NetSuite data available through SuiteAnalytics Connect. The Browser provides a page for each table. This page lists columns, column descriptions, and related tables.

The Connect Browser also includes domain diagrams that illustrate the relationships among tables in particular business domains. In this release, these diagrams cover only the most widely used domains.

The Connect Browser is targeted to be available in fourth quarter, 2015. This new browser will be integrated with the SuiteScript Records Browser and the SuiteTalk Schema Browser.

More Options for Custom Fields in Summary Reports

As of Version 2015 Release 2, you can customize the way custom fields are used in summary reports.

Previously, all custom fields supported only the Group summary type, which caused the data on summary reports to be grouped by these custom fields. Now there are other summary types available for those custom fields that have numeric, percentage, or currency values.

To change the summary type for a custom field, open the report in the Report Builder, select the custom field column, and choose an appropriate summary type for it.

The summary type defines how to group data on the summary report. Now you can choose among the following summary types for custom fields:

- **Group** – The default summary type for custom fields. Rolls up report results by the field to which you apply this summary type.
- **Count** – Counts the number of report results for this field.
- **Sum** – Sums the values for the field.
- **Minimum** – Displays the minimum value.
- **Maximum** – Displays the maximum value.
- **Average** – Calculates the average value for the field.

For more information about grouping data on summary reports, see the help topic [Arranging Data on Summary Reports](#).

New Localization Preference for Search Sort Order

As of Version 2015 Release 2, you can sort search results in the U.S. English sort order, regardless of your NetSuite language settings.

Every language has an alphabetical order with specific conventions for handling diacritics, capital letters, and numbers. This sort order and these conventions may also vary based on the country where a language is used.

Now you can choose between the U.S. English sort order and the sort order specific to your NetSuite language settings. To choose a search sort order, go to Home > Set Preferences. On the General subtab, select your preferred sort order from the Search Sorting list and click Save.

In addition, account administrators can assign default search sort order settings to a company or subsidiary and restrict users from changing these settings.

Redesigned Report Results Page

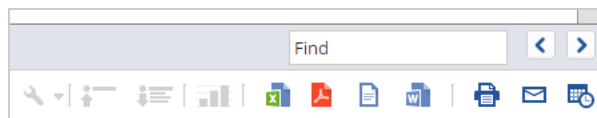
As of Version 2015 Release 2, the Report Results page is displayed with a different font and includes a redesigned footer.

Report results are now displayed using the Open Sans font, unless you have set a different default font at Home > Set Preferences, on the Appearances subtab.

Before Version 2015 Release 2, the default font for reports was Arial. On financial reports, you can still use Arial as the default font. To enable this option, at Home > Set Preferences, on the Analytics subtab, select Customize Font on Financial Reports and click Save. Enabling this option changes the default font for financial reports back to Arial and enables you to configure the font further in the Financial Report Builder.

Please note that if you previously configured the font style, size, and color on your custom reports, these settings will be preserved: only the font itself will change. To learn more about configuring font options for financial reports, see the help topic [Setting Financial Statement Formatting Options](#).

In addition, as of this release, the export options in the report footer appear as a set of icons. Before Version 2015 Release 2, the export options were included in the Export menu in the report footer.



New Permission to Persist Search Results

Persisting search results enables you to run saved searches asynchronously, for up to 3 hours, saving search results in a CSV file. This capability is useful for business-critical searches that contain a high volume of data. Prior to this release, only account administrators could persist search results.

Version 2015 Release 2 includes a new Persist Search permission that administrators can assign to other roles. Users with these roles can run saved searches in the background with results persisted in a CSV file. This permission is a Lists type permission.

For more details, see the help topic [Persisting Search Results](#).

Labor Cost Now Available for Saved Searches

You can now create a saved search including the Labor Cost field from project records. The labor cost is displayed in the base currency of the user generating the search results.

Changes to the Left Side Field in Financial Reports and SuiteAnalytics Connect

NetSuite Account records include a **Left Side** field that indicates whether the normal balance for an account is debit or credit. A **Yes** (or **true**) value for this field means an account is a debit account. A **No** (or **false**) value means an account is a credit account.

Previously, financial reports and SuiteAnalytics Connect incorrectly interpreted the **Left Side** field values for Cost of Goods Sold, Expense, and Other Expense accounts. These types of accounts were erroneously classified as credit accounts.

As of Version 2015 Release 2, financial reports and SuiteAnalytics Connect interpret Cost of Goods Sold, Expense, and Other Expense accounts as debit accounts. In financial reports the **Left Side** column for these types of accounts will have a value of **Yes**. Similarly, in SuiteAnalytics Connect, the **is_leftside** column for these types of Account tables will have a **T (true)** value.

This new behavior ensures that all account types are accurate and consistent throughout NetSuite.

End of Support for the ODBC 6.0 Driver

The ODBC 6.0 driver has not been available for download as of Version 2014 Release 2, when the NetSuite ODBC 7.0 drivers for 32-bit and 64-bit Windows and Linux were released. SuiteAnalytics Connect ODBC 7.0 drivers offer superior performance, enhanced security, and improved navigation of the NetSuite Connect schema by exposing primary and foreign key

metadata. For these reasons, NetSuite encourages all SuiteAnalytics Connect customers to update to the ODBC 7.0 drivers as soon as possible.

To place more resources into strategic SuiteAnalytics projects that benefit all customers, NetSuite will no longer support the legacy ODBC 6.0 driver as of February 1, 2016. To upgrade to the latest SuiteAnalytics Connect ODBC driver, remove your legacy ODBC driver and then download and install the latest SuiteAnalytics Connect driver.

For information on removing the ODBC 6.0 driver, see the help topic [Removing Connect Drivers](#).

For more information on installing, configuring, or transitioning to the new driver, see the help topic [Downloading and Installing Connect Drivers](#).

To learn how to set up certificate authentication for the ODBC driver, see the help topic [Authentication Using Server Certificates for ODBC](#).

Pivot Reports BETA Replacement

The SuiteAnalytics team will replace Pivot Reports BETA with an improved analytics solution as a part of the core NetSuite application starting with Version 2017 Release 1 or earlier. Therefore, the Pivot Reports BETA SuiteApp will be discontinued starting with Version 2017 Release 1.

CSV Import

Version 2015 Release 2 includes the following changes to CSV Import:

- [CSV Support for Custom Segment Values](#)
- [More Formatting Options for CSV Files](#)

CSV Support for Custom Segment Values

Version 2015 Release 2 includes support for importing the values of custom segments.

Custom segments are classification fields similar to class, department, and location. When the Custom Segments feature is enabled, you can use the NetSuite UI to create as many custom segments as necessary for your organization. For example, you might create a custom segment called Sales Channel that has values such as Brick-and-Mortar Store, Call Center, and Web Store.

With the new CSV Import option, you can import values for an existing custom segment. For example, with the segment called Sales Channel, you could use the Import Assistant to import new values such as Catalog and Partner Outlet.

The Custom Segments feature is new in this release and is described further in the [Custom Segments \(Beta\)](#) release note.

For more information about this import, see the following:

- [Prerequisites for Importing Values](#)
- [Segments Appear Under Classification Import Type](#)

Important: Custom segments are currently available as a beta feature.

Prerequisites for Importing Values

Before you can import values for custom segments, you must do both of the following:

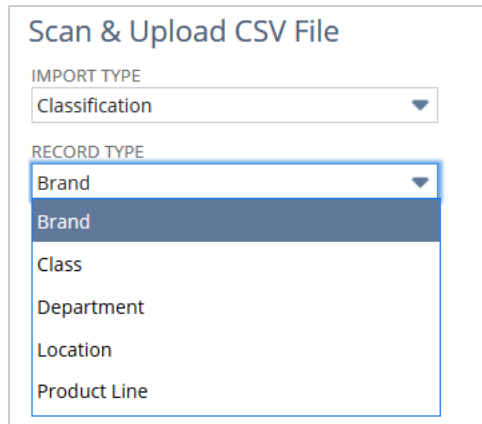
- Enable the Custom Segments (Beta) feature at Setup > Company > Enable Features, on the SuiteCloud subtab.
- Create at least one custom segment in your account. You can create custom segments at Customization > Lists, Records, & Fields > Custom Segments > New.

By default, only administrators and users with the Full Access role have permission to import custom segment values. However, you can grant permission to other roles. The following permissions are required:

- **The global Custom Segments permission** – You can add this permission on the Permissions > Setup subtab of the role record. At least the View level is required.
- **The Value Access Level permission for the custom segment** – You can add this permission on the Custom Record subtab of the role record. To be able to add values, the role needs at least the Create level. To be able to update values, or to be able to set the parent for a value, the user needs at least the Edit level.

Segments Appear Under Classification Import Type

To import values, set the Import Type to Classification. In the Record Type dropdown list, the system displays both the names of your custom segments and any other classifications enabled in your account. For example, custom segments named Brand and Product Line would appear with Class, Department, and Location, if those features were enabled.



Scan & Upload CSV File

IMPORT TYPE
Classification

RECORD TYPE
Brand
Brand
Class
Department
Location
Product Line

For each import, you can define values for the following fields:

- The name of the value.
- A parent for the value. For example, if your segment has a value called Retail Store, that value might be the parent of values such as Express Kiosk and Superstore.
- Inactive, which determines whether the segment value is available to be selected.
- Translations of the value's name, if your account uses the Multi-Language feature.

For more information, see the help topic [Custom Segment Value Import](#).

More Formatting Options for CSV Files

In Version 2015 Release 2, the Import Assistant supports additional formatting options for CSV files.

Newly Supported Value Separators

In addition to the comma, which was previously the only permitted CSV column separator, it is now possible to use the semicolon, pipe, space, and tab.

Comma Supported as a Decimal Mark

Now you can import data that uses the comma as the decimal mark in numbers.

Setting the Value Separator and Decimal Mark for Imports

You can set the value separator and decimal mark for a single CSV job import in the Import Assistant. You set the value separator in the new **CSV Column Delimiter** field on the Scan & Upload File page. You set the decimal mark in the new **CSV Decimal Delimiter** advanced option on the Import Options page.

The default settings for these Import Assistant fields are based on the values set for CSV Column Delimiter and CSV Decimal Delimiter at Home > Set Preferences, on the General subtab, in the Formatting section. If most or all files you import use the same value separator and decimal mark, you can set these preferences to avoid the need to set these values in the Import Assistant each time. The default value separator is the comma, and the default decimal mark is the period.

An account administrator or another user with appropriate permissions can assign default CSV import formatting settings for the whole company at Setup > Import/Export > CSV Import Preferences. For a subsidiary, these settings can be assigned at Setup > Company > Subsidiaries, the subsidiary's page Preferences subtab, on the CSV subtab.

To learn more about these formatting options, see the help topics [Setting CSV Import Preferences](#) and [Tips for Using Numbers in CSV Files](#).

SuiteFlow (Workflow Manager)

Version 2015 Release 2 includes the following enhancement to SuiteFlow features:

- [Dynamic Mode for Workflows](#)
- [Action Ordering Rules for Workflows](#)
- [Changes for Workflows Referencing Email Templates](#)
- [Upcoming Change for Workflows with Timesheets](#)

Dynamic Mode for Workflows

Workflows will consistently run in dynamic mode in NetSuite accounts created as of Version 2015 Release 2 or later. Consequently, the dynamic effects of setting field values including validation, sourcing, and calculation will be visible in real-time when running workflows. This visibility allows subsequent actions and transitions to react to these effects. In previous releases, dynamic mode was not consistently used when running workflows.

In NetSuite accounts that existed prior to NetSuite Version 2015 Release 2, existing and newly created workflows will continue to run as they did prior to this release.

Note: In some cases, SuiteFlow cannot use dynamic mode. These cases include actions and transitions triggered on Before Record Load. For more information, see the help topic [Dynamic Mode for Workflows](#).

Action Ordering Rules for Workflows

In previous releases, new actions were added to the end of a list of actions for a workflow and could be rearranged in any order. Also, though actions were ordered by trigger in the context

panel, this order had no effect on the way actions were run. Instead, actions were run in the order they were listed on the global list of actions, viewed on the State Definition window.

Beginning in Version 2015 Release 2, a set of rules was created to organize actions in the same way on both the global list and on the context panel. Workflow actions must be ordered based on their triggers, according to the following order:

1. Entry
2. Exit
3. Before Record Load
4. Before User Edit
5. Before Field Edit
6. After Field Edit
7. After Field Sourcing
8. Before User Submit
9. After Record Submit
10. Scheduled

When a user adds a new action to a workflow, the new action is added right after the last action of the same trigger. If the trigger has no other action, SuiteFlow goes up the order of triggers and tries to do the same for each preceding trigger. If no such preceding action is found, the new action is added to the beginning of the list of actions.

When a user changes the trigger of an existing action, the action is put in the same position as if it were newly added. This position is immediately after the last action of the same trigger or preceding trigger, or at the beginning of the list of actions if no preceding action is found.

When a user wants to move an action, if there are other actions with the same trigger, the action can be placed before or after an action with the same trigger. Alternately, it can be placed right after the last action with the preceding trigger. SuiteFlow goes up the order of triggers and picks the first **last action** available. If there are no actions with preceding triggers, the action can be moved to the first position in the list.

If you attempt to move an action to an inappropriate place in the action list, an error appears.

Note: The global action order can be viewed in the Workflow State window. In the Context Panel, actions are grouped by their trigger type, so only the mutual order of actions of the same trigger type is visible.

Changes for Workflows Referencing Email Templates

Version 2014 Release 1 introduced scriptable templates as a replacement for CRMSDK-based email templates. CRMSDK templates were deprecated with Version 2015 Release 1, and will no

longer be supported as of Version 2016 Release 1. To ensure that existing templates continue working after the Version 2016 Release 1 upgrade, you must convert them from CRMSDK templates to scriptable templates.

In Version 2015 Release 2, there will be no impact to existing workflows that reference CRMSDK templates. However, if you convert a CRMSDK template to a scriptable template, the new scriptable template will have an ID different from the old CRMSDK template. Any workflow references to this template must be updated to use the new ID for the scriptable template. Workflow references to CRMSDK templates will no longer be supported as of Version 2016 Release 1.

For more information about the transition to scriptable templates, see [Phasing out CRMSDK-Based Email Templates](#). For information about the SuiteScript impact, see [nlapiMergeRecord Deprecated](#).

Upcoming Change for Workflows with Timesheets

The release of an updated scripting engine providing improved SuiteScript and SuiteFlow support for timesheets is targeted for a future release. It is recommended that you do not modify or create any scripts or workflows for timesheets until after the release of the new scripting engine.

SuiteScript

Refer to the following sections for details on all SuiteScript updates for Version 2015 Release 2:

- [SuiteScript 2.0 – Beta Release](#)
- [Script Execution Log – Persistence Enhancement](#)
- [nlapiTriggerWorkflow Change](#)
- [nlapiMergeRecord Deprecated](#)
- [Enhancements to Token-based Authentication for RESTlets](#)
- [SuiteScript Support for Use Deletion Reason Feature](#)
- [SuiteScript Change for Extended Accounting Period Close Process Feature](#)
- [Upcoming Change for Scripting with Timesheets](#)
- [New APIs](#)
- [Updated APIs](#)
- [2015.2 SuiteScript Records Browser](#)

SuiteScript 2.0 – Beta Release

Version 2015 Release 2 introduces the beta version of SuiteScript 2.0, a complete re-factor of the SuiteScript model.

Important: SuiteScript 2.0 is a beta feature. The contents of this feature are preliminary and may be changed or discontinued without prior notice. Any changes may impact the feature's operation with the NetSuite application. NetSuite warranties and product service levels shall not apply to the feature or the impact of the feature on other portions of the NetSuite application. NetSuite may review and monitor the performance and use of the feature.

Note: The introduction of SuiteScript 2.0 does not change how SuiteScript 1.0 scripts are written. There is only one SuiteScript 2.0 enhancement that impacts SuiteScript 1.0 – the enhancement to the Script record creation process. See [Script Record UI Enhancements](#) for additional information.

- [Advantages to SuiteScript 2.0](#)
- [Script Creation Process](#)
- [SuiteScript 2.0 – The Basics](#)

Advantages to SuiteScript 2.0

- [Modular Architecture](#)
- [Modern JavaScript Syntax and Behavior](#)
- [New and Improved API Functionality](#)
- [Asynchronous Client Side Processing \(Promises\)](#)
- [New Batch Processing Framework \(Map/Reduce Script Type\)](#)

Modular Architecture

SuiteScript 2.0 is built on modularity. With it, you have access to a complete set of new APIs, contained within modules. These modules are organized based on behavior. SuiteScript 2.0 also enables you to create your own custom modules. You can use these custom modules to organize your code (as a replacement for SuiteScript 1.0 libraries). Additionally, you can add custom modules to SuiteApps and expose those modules to third parties.

SuiteScript 2.0 implements its modular architecture with the Asynchronous Module Definition (AMD) specification. AMD is used to define JavaScript modules and their dependencies. For additional information regarding AMD, see <http://requirejs.org/docs/whyamd.html>.

Dependency Management and Improved Performance

SuiteScript 2.0 gives you built-in dependency management. It also gives you improved performance. With SuiteScript 2.0, you define the dependencies that must load prior to module

execution. The module loader loads only the modules that you specify. Because you are not loading all available dependencies at one time (as you do in SuiteScript 1.0), your script loads faster. Plus, required dependencies are loaded asynchronously in client-side scripts.

Complexity Management and Intuitive Code Organization

SuiteScript 2.0 encompasses a large, robust set of APIs. Modern SuiteApps require complex SuiteScripts that typically consist of many lines of code. SuiteScript 2.0 gives users built-in complexity management. Modularity adds encapsulation and provides intuitive code organization. It also ensures there are no global variable or method naming conflicts. Dependency management enables you to concentrate on logic instead of dependencies and load order.

Modern JavaScript Syntax and Behavior

The underlying design principle of this new version of SuiteScript is that **SuiteScript 2.0** **=== JavaScript**. The syntax is straightforward and the behavior is intuitive. SuiteScript 2.0 is designed to encourage the best practices required to scale to large, complex SuiteApps.

The following is a basic example of a SuiteScript 2.0 script.

```
/**
 * @NAPIVersion 2.0
 * @NScriptName SyntaxSample
 * @NScriptId syntax_sample
 */
define(['N/file'],
function(file)
{
    function doSomething()
    {
        var fileObj = file.create({
            name: 'test.text',
            fileType: file.Type.PLAINTEXT,
            contents: 'Hello World\nHello World'
        });
        fileObj.folder = 30;
        var fileId = fileObj.save();
    }
    return
    {
        afterSubmit: doSomething
    }
}
);
```

Consistent Behavior

Consistency is an important aspect of SuiteScript 2.0. The following are several prominent examples.

- All SuiteScript 2.0 methods take a plain JavaScript object as an input. In the example below, the method `record.load` takes in a JavaScript object that consists of two key/value pairs: `type: record.Type.SALES_ORDER` and `id: 6`.

```
var recObj = record.load({
  type: record.Type.SALES_ORDER,
  id: 6
});
```

Within the SuiteScript 2.0 API, all method inputs are named `options`. For example, the `record.load` signature is listed as `record.load(options)`.

Note: JavaScript objects can contain a mixture of value types. See the applicable SuiteScript 2.0 API help topic for supported value types.

- All SuiteScript 2.0 booleans take a value of `true` or `false`. All other boolean values (for example: `T` or `F`) throw an error.
- Parameter types in SuiteScript 2.0 are strictly adhered to. You must pass in valid parameter types, as listed in the SuiteScript 2.0 help. SuiteScript 2.0 does not convert invalid parameter values to valid values.

nlapi/nlobj Prefix Retirement

SuiteScript 2.0 is modeled to look and behave like modern JavaScript. To facilitate that objective, SuiteScript 2.0 methods and objects are not prefixed with `nlapi` and `nlobj`.

This change also reflects the modular organization of SuiteScript 2.0. SuiteScript 1.0 methods and objects respectively belong to the `nlapi` and `nlobj` namespaces. SuiteScript 2.0 methods and objects are encapsulated within various modules.

Properties and Enumerations

SuiteScript 2.0 adopts the usage of properties and enumerations. Most SuiteScript 1.0 getter and setter methods are replaced with properties. Enumerations encapsulate common constants (for example, standard record types).

Note: JavaScript does not include an enumeration type. The SuiteScript 2.0 documentation utilizes the term `enumeration` (or `enum`) to describe the following: a plain JavaScript object with a flat, map-like structure. Within this object, each key points to a read-only string value.

Updated Sublist Indexing

The standard practice in the development world is to start indexing at 0. This behavior is observed in the majority of programming languages. SuiteScript 1.0 starts sublist indexing at 1. To bring SuiteScript into alignment with modern JavaScript, sublist indexing within SuiteScript 2.0 begins at 0.

New and Improved API Functionality

SuiteScript 2.0 includes the following new functionality.

Expanded Support for HTTP Content Type Headers

SuiteScript 2.0 adds support for all HTTP content types. This enhancement applies to both client request and server response HTTP headers. Note that SuiteScript 2.0 uses the same content type default value used by SuiteScript 1.0.

New Encryption/Encoding Functionality

SuiteScript 2.0 adds enhanced encryption, decryption, and hashing functionality with the crypto module. Additional encoding functionality is exposed in the encode module.

Asynchronous Client Side Processing (Promises)

With SuiteScript, asynchronous processing is especially important in client remote object scripts. Client remote object scripts are client scripts that require a response from the server. You can use nested callback functions to execute code asynchronously, but this method is confusing when dealing with complex operations. The end result is often code that is difficult to read.

In SuiteScript 2.0, all client scripts now support the use of promises. With promises, developers can write asynchronous code that is intuitive and efficient. SuiteScript 2.0 provides promise APIs for selected modules (see [SuiteScript 2.0 Promise APIs](#)). In addition, you can create custom promises in all client scripts (see [Custom Promises](#)).

A promise is a JavaScript object that represents the eventual result of an asynchronous process. After this object is created, it serves as a placeholder for the future success or failure of the operation. During the period of time that the promise object is waiting, the remaining segments of the script can execute.

A promise holds one of the following values:

- fulfilled – The operation is successful.
- rejected – The operation failed.
- pending – The operation is still in progress and has not yet been fulfilled or rejected.

When it is first created, a promise holds the value pending. After the associated process is complete (from success or failure), the value changes to fulfilled or rejected. A success or failure callback function attached to the promise is called when the process is complete. Note that a promise can only succeed or fail one time. When the value of the promise updates to fulfilled or rejected, it cannot change.

For additional information regarding promises, see <https://www.promisejs.org/>.

SuiteScript 2.0 Promise APIs

To make promises more accessible, SuiteScript 2.0 provides client-side promise APIs for the following modules:

- email
- http
- https
- record
- search
- transaction

Important: Although these modules as a whole are supported in client and server-side scripts, their promise APIs are supported only in client scripts.

The available promise APIs are named so that they correspond with their synchronous counterparts. The distinction is that the promise APIs have names that are suffixed with `.promise`. For example, the `search.create(options)` API has a promise version named `search.create.promise(options)`.

The following is a basic example of how to use a promise API in a client script.

```
/**
 * @NAPIVersion 2.0
 * @NScriptName PromiseAPISample
 * @NScriptId promise_api_sample
 */
define(['N/search'],
function(search)
{
    function doSomething()
    {
        search.create.promise({
            type: 'salesorder'
        })
        .then(function(result) {
            log.debug("Completed: " + result);
            //do something after completion
        })
        .catch(function(reason) {
            log.debug("Failed: " + reason)
            //do something on failure
        });
    }
    return
    {
        pageInit: doSomething
    }
});
```

This example demonstrates how to chain promises created with promise APIs.

```
/**
 * @NAPIVersion 2.0
 * @NScriptName ChainedPromiseAPISample
 * @NScriptId chained_promise_api_sample
 */
define(['N/search'],
  function(search)
  {
    function doSomething()
    {
      var filter = search.createFilter({
        name: 'mainline',
        operator: search.Operator.IS,
        values:['T']
      });
      search.create.promise({
        type: 'salesorder',
        filters:[filter]
      })
      .then(function(searchObj) {
        return searchObj.run().each.promise(
          function(result, index){
            //do something
          })
      })
      .then(function(result) {
        log.debug("Completed: " + result)
        //do something after completion
      })
      .catch(function(reason) {
        log.debug("Failed: " + reason)
        //do something on failure
      });
    }
    return
    {
      pageInit: doSomething
    }
  }
);
```

Custom Promises

The following example shows a custom promise. Custom promises do not utilize the promise APIs provided with SuiteScript 2.0 (see [SuiteScript 2.0 Promise APIs](#)).

```
/**
 * @NAPIVersion 2.0
 * @NScriptName CustomPromiseAPISample
 * @NScriptId custom_promise_api_sample
 */
define(function(){
  function doSomething(addresses){
    var promise = new Promise(function(resolve, reject){
      var url = 'https://your.favorite.maps/api/directions?start=' + addresses.start
      + '&end=' + addresses.end,
```

```
        isAsync = true,
        xhr = new XMLHttpRequest();

        xhr.addEventListener('load', function (event) {
            if (xhr.readyState === 4) {
                if (xhr.status === 200) {
                    resolve(xhr.responseText );
                }
                else {
                    reject( xhr.statusText );
                }
            }
        });
        xhr.addEventListener('error', function (event) {
            reject( xhr.statusText );
        });
        xhr.open('GET', url, isAsync);
        xhr.send();
    });

    return promise;
}

return {
    lookupDirections: doSomething
};
});
```

New Batch Processing Framework (Map/Reduce Script Type)

Map/reduce is a programming model that enables large amounts of data to be efficiently batch processed. In general (not specific to SuiteScript), the basic flow of map/reduce is as follows:

- Take in a large data set.
- Parse the data into key/value pairs.
- Group values based on keys.
- Evaluate the data in each group.
- Output the results.

SuiteScript 2.0 introduces a new server-side script type based on the map/reduce model. Map/reduce scripts provide a structured framework that simplifies the development of scripts that batch process. In addition to batch processing, SuiteCloud Plus users can also use map/reduce scripts to process records in parallel across multiple work queues. Users manually select the number of work queues to utilize from the script deployment record.

Note: The map/reduce script type automatically tracks governance usage and yielding. The SuiteScript 1.0 functions `nlapisetRecoveryPoint` and `nlapiyieldScript` have no counterparts in SuiteScript 2.0.

The map/reduce script type goes through at least three of five possible stages.

Important: A map/reduce script is not required to go through both the Map and Reduce stages. One of these stages can be skipped, but the script must go through one of them.

The stages are processed in the following order. Note that each stage must complete before the next stage begins.

- Get Input Data – Takes in the original data set. This stage is always processed first and is required. The Get Input Data stage runs sequentially..
- Map – Parses each row of data into zero or more key/value pairs. If this stage is skipped, the Reduce stage is required. Data is processed in parallel in this stage.
- Shuffle – Groups values based on keys. This stage is automatically processed after the Map stage is processed. Data is processed sequentially in this stage.
- Reduce – Evaluates the data in each group. If this stage is skipped, the Map stage is required. Data is processed in parallel in this stage.
- Summarize – Summarizes the metadata of the task. This stage is optional and is not technically a part of the map/reduce process. The Summarize stage runs sequentially.

You can think of map reduce scripts as a subclass of scheduled script. Deployment is handled in the same manner. Both script types have three options for deployment: by schedule, from the **Save and Execute** option on the deployment record, or through the task module.

With the map/reduce script type, SuiteScript 2.0 also introduces a new map/reduce API. The map/reduce API includes four entry point functions that control the script's flow into the stages listed above.

- `getInputData()` – Starts the map/reduce process. Takes the script into the Get Input Data stage and returns an array, object, or search result set. This entry point function is required.
- `map()` – Takes the script into the Map stage, applies your custom logic to each item, and then returns key/value pairs. Note that the Shuffle stage is automatically processed after the Map stage is processed. If this entry point function is skipped, `reduce()` is required.
- `reduce()` – Takes the script into the Reduce stage, applies your custom logic to each group, and then outputs the result. If this entry point function is skipped, `map()` is required.
- `summarize()` – Takes the script into the Summarize stage and applies your custom logic to the result set. This entry point function is optional.

The following example shows the basic framework for a map/reduce script.

```
define(['N/record'],
function(record)
{
    function soSomething1() {
        //logic goes here
    }
}
```



```
    }  
    function doSomething2(context) {  
    //logic goes here  
    }  
    function doSomething3(context) {  
    //logic goes here  
    }  
    function doSomething4(context) {  
    //logic goes here  
    }  
    return  
    {  
    config: {  
    exitOnError: false //default  
    }  
    getInputData: doSomething1,  
  
    map: doSomething2,  
  
    reduce: doSomething3,  
  
    summarize: soSomething4  
    }  
  }  
);
```

SuiteScript 2.0 – The Basics

- [Features Not Supported in SuiteScript 2.0 \(Beta\)](#)
- [Version Cohabitation Rules](#)
- [SuiteScript 2.0 Feature Setting](#)
- [Script Creation Process](#)
- [Script Types](#)
- [Script Entry Points](#)
- [Global Objects](#)
- [SuiteScript 2.0 Modules](#)
- [Custom Modules](#)
- [JSDoc](#)
- [Script Record UI Enhancements](#)
- [NetSuite Debugger Support](#)

Features Not Supported in SuiteScript 2.0 (Beta)

The following are not supported in the beta release of SuiteScript 2.0:

- Form-level client scripts

- Mass update scripts
- Workflow action scripts
- Bundle installation scripts
- SSP scripts
- Custom Plug-ins
- Core Plug-ins

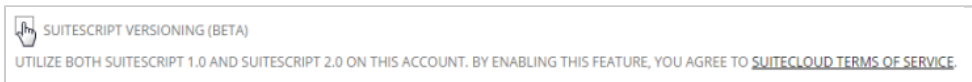
Version Cohabitation Rules

Your script (entry point script and supporting library scripts) must use either SuiteScript 1.0 or SuiteScript 2.0. You cannot use APIs from both versions in one script.

You can, however, have multiple scripts that use different SuiteScript versions. These can be deployed in the same account, in the same SuiteApp, and on the same record.

SuiteScript 2.0 Feature Setting

To utilize SuiteScript 2.0, you must first enable it in the NetSuite user interface (UI). To facilitate this, NetSuite introduces a new setting on the Enable Features page. You must enable this setting before you upload your first SuiteScript 2.0 script.



Important: After SUITESCRIPT VERSIONING (BETA) is enabled, it cannot be disabled.

Script Creation Process

Note: The following is a very basic process flow for script creation. Your specific process may vary, depending on the content of your script.

Stage	Description	Additional Information
1	Enable SuiteScript 2.0.	SuiteScript 2.0 Feature Setting
2	Add required JSDoc tags to your entry point script. Your entry point script is the script you attach to the script record.	JSDoc
3	Use the <code>define()</code> statement to load modules in your entry point script.	Module Loader
4	Add at least one entry point function to your entry point script. This is where most of your logic goes.	Script Types
5	Organize your code into custom modules (as a replacement for SuiteScript 1.0 libraries). Load these custom modules in your entry point script.	Custom Modules as Support Libraries
6	Upload your script to NetSuite.	Script Record UI Enhancements

Script Types

The following script types are supported in SuiteScript 2.0 (Beta).

Type	SuiteScript 2.0 Changes
Client Scripts	The SuiteScript 1.0 function <code>recalc</code> is replaced with <code>subListChanged</code> in SuiteScript 2.0 Important: Form-level client scripts are not supported in SuiteScript 2.0 (Beta).
Map/Reduce Scripts	Map/Reduce is a new script type in SuiteScript 2.0. For additional information, see New Batch Processing Framework (Map/Reduce Script Type) .
Portlet Scripts	SuiteScript 2.0 portlet scripts behave in the same way as SuiteScript 1.0 scripts. All portlet scripts must include an additional JSDoc tag, <code>@NScriptPortletType</code> , at the beginning of the script. For additional information, see SuiteScript 2.0 JSDoc Tags .
RESTlet Scripts	SuiteScript 2.0 RESTlet scripts behave in the same way as SuiteScript 1.0 scripts.
Scheduled Scripts	SuiteScript 2.0 scheduled scripts behave in the same way as SuiteScript 1.0 scripts.
Suitelet Scripts	SuiteScript 2.0 suitelet scripts behave in the same way as SuiteScript 1.0 scripts.
User Event Scripts	SuiteScript 2.0 user event scripts behave in the same way as SuiteScript 1.0 scripts.

Script Entry Points

In SuiteScript 2.0, script entry points (an example client script entry point is `pageInit`) are no longer selected on the script record. Script entry points are now included in the code. Each entry point script (the script attached to the script record) must include at least one entry point and corresponding entry point function. An entry point function is a public function that is executed when a script entry point is triggered. The primary logic within an entry point script is contained within its entry point functions.

Important: Place all logic in at least one named function. In an entry point script, your logic must be executed after an entry point is invoked. Logic executed before an entry point is invoked throws an error.

```
define(['N/record'],
function(record) {
    function alertPageLoaded(context) {
        alert('Page is loaded');
    };
    function fieldChanged(context) {
        var currentRecord = context.currentRecord;
        auditRecordChanges(currentRecord.type, currentRecord.id);
        updateMemoFieldWhenItemFieldIsChanged(context);
    }
    function auditRecordChanges(type, id) {
        var logRecord = record.create({type: 'customrecord_log', isDynamic: true});
        logRecord.setValue('custrecord_recordType', type);
        logRecord.setValue('custrecord_recordId', id);
        logRecord.setValue('custrecord_changedDate', new Date());
        logRecord.save();
    }
    function updateMemoFieldWhenItemFieldIsChanged(context) {
```

```
    if(context.fieldId === 'item') {
        context.currentRecord.setValue('memo', 'item field is changed');
    }
}
return Object.freeze({
    pageInit: alertPageLoaded,
    fieldChanged: fieldChanged
});
});
```

When a SuiteScript 2.0 script is uploaded to NetSuite and a script record is requested, several new things occur. NetSuite analyzes the script's source code to determine the script version, script type, and script entry points. NetSuite also parses the code for JSDoc tags. See [JSDoc](#).

Global Objects

The following are global objects within SuiteScript 2.0. You are not required to load these to use them.

define() Statement

Use the `define()` statement to load SuiteScript 2.0 modules and create custom modules. The `define()` statement returns a value.

require() Statement

The `require()` statement is also used to load modules. The `require()` statement has no return statement. Use the `require()` statement to debug your scripts in the NetSuite Debugger.

log Object

The log object replaces the SuiteScript 2.0 `nlapILogExecution()` API.

util Object

The util object includes the basic functions used to verify type (for example, `isObject()` and `isString()`)

JSON

See <http://www.ietf.org/rfc/rfc4627.txt> for additional information.

Promise

See [Asynchronous Client Side Processing \(Promises\)](#) for additional information.

SuiteScript 2.0 Modules

SuiteScript 2.0 introduces a complete set of new APIs that are organized into various modules, based on behavior. These modules are described below.

Note: SuiteScript 2.0 gives you the ability to load all SuiteScript 2.0 modules at one time. This is done by passing the modules' parent directory to the require statement and its callback function: `require(['N'], function(N) {...});` This is a convenient way to load all modules, but does sacrifice the performance advantage of only loading modules that are needed. NetSuite provides this feature so that you can test and familiarize yourself with SuiteScript 2.0. It is not recommended that you load all modules at once in a production environment.

Module Name	Supported Script Types	Description
auth	Server-side scripts	The auth module changes NetSuite login credentials.
config	Server-side scripts	The config module accesses NetSuite configuration settings. The following configuration pages are supported: <ul style="list-style-type: none"> • Set Preferences page (Home > Set Preferences) • Company Information page (Setup > Company > Company Information) • General Preferences page (Setup > Company > General Preferences) • Accounting Preferences page (Setup > Accounting > Accounting Preferences) • Accounting Periods page (Setup > Accounting > Manage Accounting Periods) • Tax Periods page (Setup > Accounting > Manage Tax Periods) • Enable Features page (Setup > Company > Enable Features)
crypto	Server-side scripts	The crypto module encapsulates hashing, hash-based message authentication, and symmetrical encryption.
currency	Client and server-side scripts	The currency module returns an exchange rate between two currencies. Note: Currency formatting is handled by the format module.
email	Client and server-side scripts	The email module sends regular, bulk, and campaign email messages.
encode	Server-side scripts	The encode module re-encodes an encoded string. The following encoding specifications are supported: <ul style="list-style-type: none"> • UTF-8 • Base16 • Base32 • Base64 • URL-safe base64 • hex
error	Client and server-side scripts	The error module creates custom SuiteScript errors. These custom errors are used in try-catch statements to abort script execution.

Module Name	Supported Script Types	Description
file	Server-side scripts	The file module works with files within NetSuite.
format	Client and server-side scripts	The format module converts strings into a specified format and parses formatted data into strings.
http	Client and server-side scripts	The http module replaces the client and server-side HTTP functionality found within the SuiteScript 1.0 API nlapiRequestURL. Note: In SuiteScript 2.0, all HTTP content type headers are supported.
https	Client and server-side scripts	The https module replaces the functionality found within the SuiteScript 1.0 API nlapiRequestURLWithCredentials. The https module replaces the client-side HTTPS functionality found within the SuiteScript 1.0 API nlapiRequestURL. Important: Some https members are supported only in server-side scripts.
record	Client and server-side scripts	The record module works with NetSuite records.
redirect	Varies by method	The redirect module redirects users to one of the following: <ul style="list-style-type: none"> • URL • Suitelet • Record • Task link • Saved search • Unsaved search
render	Server-side scripts	The render module encapsulates the following functionality: <ul style="list-style-type: none"> • Printing • PDF creation • Form creation from templates • Email creation from templates
runtime	Client and server-side scripts	The runtime module accesses the current runtime settings for the following: <ul style="list-style-type: none"> • company • script • session • system • user • version

Module Name	Supported Script Types	Description
search	Client and server-side scripts	The search module works with NetSuite searches and saved searches. This module also encapsulates search columns and search filters.
sso	Suitelets, portlet scripts, and user event scripts	The sso module generates outbound single sign-on (SuiteSignOn) tokens.
task	Server-side scripts	The task module works with tasks that are placed in the work queue and asynchronously executed. It can create the following task types: <ul style="list-style-type: none"> • Scheduled Script • Map/Reduce • CSV Import • Entity Deduplication • Workflow Trigger
transaction	Client and server-side scripts	The transaction module voids transactions.
ui	Suitelets, portlet scripts, and beforeLoad user event scripts	The ui module creates custom forms, sublists, and controls within the NetSuite UI. This module also encapsulates a scriptable multi-step NetSuite assistant.
url	Client and server-side scripts	The url module determines URL navigation paths within NetSuite. It also enables developers to format URL strings.
workflow	Server-side scripts	The workflow module provides SuiteFlow triggers.
xml	Varies by method	The xml module creates, edits, and parses XML Documents.

Custom Modules

SuiteScript 2.0 enables you to create your own custom modules. You can use these custom modules to organize your code (as a replacement for SuiteScript 1.0 libraries). Additionally, you can add custom modules to SuiteApps and expose those modules to third parties.

Custom Modules as Support Libraries

To define a custom module to organize your code, use the `define()` statement as shown below.

```
/**
 * counter.js
 * @NAPIVersion 2.0
 * @NScriptName CustomModuleExample
 * @NScriptId custom_module_example
 */
define(function(){
    var counter = 0;

    function incrementValue() {
```

```

        counter++;
    }
    function getValue() {
        return counter;
    }

    return {
        increment: incrementValue,
        value: getValue
    }
});

```

To load the custom module defined above, pass the module's file path as follows.

```

/**
 * customRecord.js
 * @NAPIVersion 2.0
 * @NScriptName EntryPointScriptExample
 * @NScriptId entry_point_script_example
 */
define(['N/record', './counter'],
    function(record, counter){
        function createCustomRec() {
            record.create(...);
            counter.increment();
        }

        return {
            createCustomRecord: createCustomRec
        }
    });

```

Public Custom Modules in SuiteApps

To define a bundled custom module so that it is exposed to third parties, you must add the `@NModuleScope` JSDoc tag and assign it the value `public`.

```

/**
 * @NAPIVersion 2.0
 * @NScriptName ExampleScript
 * @NScriptId example_script
 * @NModuleScope public
 */

```

To utilize a module bundled within an external SuiteApp, pass the bundle's file path as follows:

```

define(['./bundle/<bundle ID>/<module path>'],
    function(<module name>){
        <logic goes here>
    }
);

```

If the bundle ID changes, the module loader will still be able to locate it. The module loader uses the bundle ID to search the SuiteApp's deprecation path.

JSDoc

JSDoc 3 is a documentation generator for JavaScript source code. Users typically employ this tool to generate an HTML API reference. Developers insert specific comment blocks into their source code. These comment blocks start with `/**` and end with `*/`.

JSDoc also includes its own markup language comprised of JSDoc tags. These tags are prefaced with the `@` symbol. The JSDoc tags are added to the comment blocks, and are used to specify the various entities of a JavaScript file (for example, `@param`).

```
/**
 * Creates a file.
 * @param {string} name [required] - file name
 * @param {string} fileType [required] - file type
 * @param {string} contents [required] - file content
 * @returns {object} file.File
 */
```

JSDoc parses the source code for each comment block. The HTML output is then generated based on the content of the comment blocks and an evaluation of the code.

JSDoc 3 also provides users with the ability to create custom JSDoc tags. These tags can be defined to trigger events (for example, displaying a certain page). SuiteScript 2.0 includes five custom tags, two are required in each entry point script uploaded to NetSuite (see [SuiteScript 2.0 JSDoc Tags](#)). When a SuiteScript 2.0 script record is requested, NetSuite uses JSDoc 3 to evaluate the entry point script and parse the code for the required JSDoc tags. These tags are used to validate the SuiteScript version. They are also used to populate several fields on the script record (see [Script Record UI Enhancements](#) for additional information).

Note: SuiteScript 2.0 users can use JSDoc 3 to create their own documentation for scripts, custom modules, and SuiteApps. To take advantage of this tool, developers must download JSDoc 3 from the official website. For additional information on JSDoc 3, see <http://usejsdoc.org/>.

How to Insert SuiteScript 2.0 JSDoc Tags

JSDoc tag comment blocks must start with a `/**` and end with a `*/` in order to be recognized.

Valid Example:

```
/**
 * @NAPIVersion 2.x
 */
```

Invalid Examples:

```
/*
 * @NAPIVersion 2.x
```

```
*/
```

```
//@NAPIVersion 2.x
```

JSDoc tags consist of a key/value pair. The key ends with the first white space after a string starting with an @. The value starts with the next non-whitespace character, and ends with the next carriage return. Each comment line within the block is prefixed with an *.

Valid Example:

```
/**
 * @NScriptName My Example Script
 */
```

Invalid Example:

```
/**
 * @NScriptName My Example Script*/
```

SuiteScript 2.0 JSDoc Tags

Before you upload your SuiteScript 2.0 entry point script to NetSuite, the script must include the first two JSDoc tags listed below.

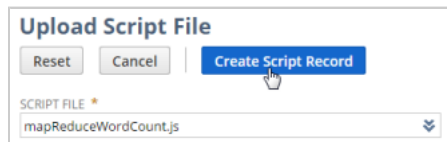
JSDoc Tag	Possible Values	Required/Optional	Description
@NAPIVersion	2.0 2.x	Required for entry point scripts Optional for custom modules	This JSDoc tag is used in two ways: <ul style="list-style-type: none"> For SuiteScript 2.0 entry point scripts, this tag is a required declaration. It signifies to NetSuite the SuiteScript version to use. For SuiteScript 2.0 custom modules that are not entry point scripts, this tag is an optional declaration. It can serve as a defense against future incompatible versions of SuiteScript (versions 3.x and higher) attempting to load it.
@NScriptName	A string	Required	This value populates the Script Name field on the Script record. The maximum length is 40 characters.
@NScriptId	A string consisting of	Optional	This value is used to populate the Script ID field on the Script

JSDoc Tag	Possible Values	Required/Optional	Description
	letters, numbers, or underscores		record. It is prefixed with "custscript_" on the script record. The default value is system generated. Spaces are not allowed. The maximum length is 28 characters.
@NModuleScope	public SuiteApp	Optional	This JSDoc tag is used in scripts that are part of a SuiteApp. <ul style="list-style-type: none"> If the value is set to public, any script in the account can load and use the module. If the value is set to SuiteApp, access to the module is limited to other modules from the same bundle. The default value is SuiteApp.
@NScriptPortletType	html list form links	Required for portlet scripts	Declares the portlet script type.

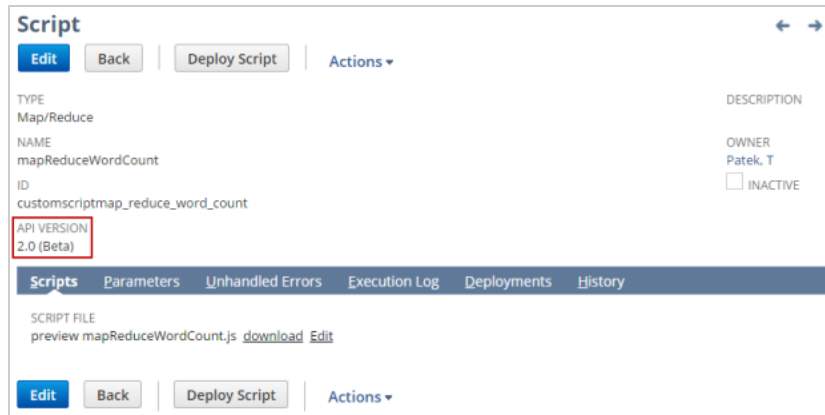
Script Record UI Enhancements

For Version 2015 Release 2, SuiteScript introduces a new process for script record creation. This enhancement impacts both SuiteScript 1.0 and SuiteScript 2.0 users.

The most obvious change is a new Upload Script File page. This page displays when you click Customization > Scripting > Scripts> New.



When you click **Create Script Record**, NetSuite parses the script to determine whether it is a SuiteScript 1.0 or SuiteScript 2.0 script. For SuiteScript 2.0 scripts, the script record is automatically populated based on the script's JSDoc tag values. You are not required to select the script type.



For SuiteScript 1.0 scripts, you must still select the script type from the Select 1.0 Script Type page. This page is unchanged. The Script record for SuiteScript 1.0 scripts, for the most part, is also unchanged. The one addition is the API Version field. This field is automatically populated with **1.0**.

NetSuite Debugger Support

SuiteScript 2.0 includes full NetSuite Debugger support. Note that you must use the `require()` statement to step through SuiteScript 2.0 scripts in the Debugger. The Debugger does not step through a `define` statement.

Script Execution Log – Persistence Enhancement

In this release, NetSuite improves the persistence of script execution logs with an enhanced repository. Previously, on each NetSuite server, if the database table exceeded the total storage limit, all logs (across all customers sharing the same database) were purged. In Version 2015 Release 2, user-generated execution logs are logged to both the existing database and a separate repository.

To view execution logs in the new repository, go to Customization > Scripting > Script Execution Logs. From this new page, you can use filters to display logs from various time periods such as last fiscal quarter. You can also download or print the script log. These logs are kept for 30 days.

Script Execution log

FILTERS

LOG LEVEL: All | DATE: All | FROM: | TO: | SCRIPT: All

DATE/TIME	LOG LEVEL	SCRIPT	DEPLOYMENT ID	USER	TITLE	DETAIL
6/17/2015 1:35:36 pm	EMERGENCY			Employee Defect	testfilteringscript1	emergency
6/17/2015 1:35:36 pm	ERROR			Employee Defect	testfilteringscript1	error
6/17/2015 1:35:36 pm	AUDIT			Employee Defect	testfilteringscript1	audit
6/17/2015 1:35:36 pm	DEBUG			Employee Defect	testfilteringscript1	debug
6/17/2015 1:35:36 pm	EMERGENCY			Employee Defect	testfilteringscript1	emergency
6/17/2015 1:35:36 pm	ERROR			Employee Defect	testfilteringscript1	error
6/17/2015 1:35:36 pm	AUDIT			Employee Defect	testfilteringscript1	audit

0 - 50 | TOTAL: 12016

Note: You can continue to access script execution details in the SuiteScript Debugger, from the Execution Log tab on the Script Deployment page, or using the scriptnote search type. However, logs stored in the existing database are not guaranteed to persist.

nlapiTriggerWorkflow Change

Prior to this release, workflow action script ids were unique per workflow. Beginning in Version 2015 Release 2, workflow action script ids will no longer be guaranteed to be unique per workflow. Script ids may be the same for one or more actions and will be identified by the parent workflow state.

To support this new behavior, a parameter, `stateid`, has been added for `nlapiTriggerWorkflow`. The value for this parameter can be either a numeric internal id or a script id. Note that internal ids are unique. There are no identical internal ids for the same record type in NetSuite, whereas script ids will now be able to be used for one or more actions. In cases where there are multiple actions with the same script id in a workflow, the new parameter links the workflow state to the action.

When the `stateid` is provided, NetSuite verifies that the `workflowid` parameter contains the `stateid` parameter. Next, NetSuite will check to make sure that the `stateid` parameter contains the proper `actionid` parameter. If any part of the verification fails, users will receive an error.

For more information, see the help topic [nlapiTriggerWorkflow \(recordtype, id, workflowid, actionid, stateid\)](#).

nlapiMergeRecord Deprecated

Version 2014 Release 1 introduced scriptable templates as a replacement for CRMSDK templates. CRMSDK templates were deprecated with Version 2015 Release 1 and will no longer be supported as of Version 2016 Release 1. To facilitate this final transition to scriptable templates, Version 2015 Release 1 also deprecated the SuiteScript function `nlapiMergeRecord`, used to perform mail merges with CRMSDK templates. This function will no longer be supported as of Version 2016 Release 1.

Version 2015 Release 1 included a new set of APIs that you must use with scriptable templates. For more information on these APIs, see the help topic [nlapiCreateEmailMerger\(templateId\)](#). Note that `nlapiMergeRecord` does not support scriptable templates. If your existing scripts use `nlapiMergeRecord`, you must do the following before your NetSuite account is upgraded to Version 2016 Release 1:

- Within the UI, convert your scripted CRMSDK templates to scriptable templates. See the help topic [Converting CRMSDK Templates to Scriptable Templates](#) for additional information.

- After your templates are converted, rewrite all applicable scripts, using the new Scriptable Email Template APIs. Use `nlapiCreateEmailMerger(templateId)` as your starting point.

SuiteScript Support for Use Deletion Reason Feature

SuiteScript supports the Use Deletion Reason feature that is newly available in Version 2015 Release 2. In some countries, providing the reason why a transaction was deleted is a legal requirement. This new feature satisfies this requirement by making it mandatory for users to record the reason for deleting a transaction. For more details, see [Record the Reason for Deleting a Transaction](#).

Every transaction record now includes two hidden fields that can be used to apply a deletion reason: `deletionreason` and `deletionreasonmemo`. A script can set values for these fields before deleting a transaction. For example, the following code snippet deletes a purchase order transaction with a reason of **Other** and a memo of **No longer needed**.

```
var po = nlapiLoadRecord('PurchaseOrder', 11);
po.setFieldValue('deletionreason', 1); // Other
po.setFieldValue('deletionreasonmemo', 'No longer needed');
nlapiSubmitRecord(po);

nlapiDeleteRecord('PurchaseOrder', 11);
```

If the Use Deletion Reason feature is enabled, and a script does not set values for these fields before deleting a transaction, the system sets default values to avoid errors. These defaults are: **Other** for `deletionreason` and **This transaction was deleted by script or web service** for `deletionreasonmemo`.

SuiteScript Change for Extended Accounting Period Close Process Feature

The new Extended Accounting Period Close Process feature, available as of Version 2015 Release 2 in accounts where Professional Services has implemented Multi-Book Accounting, causes a change in SuiteScript behavior. This feature makes it possible to individually close and reopen accounting periods associated with any accounting book, without impacting other accounting books. For more details, see [Period Close by Accounting Book](#).

Before this release, the period close status was shared among all accounting books in the system. It was possible to create a script that checked the period status, and based on this status, the script could perform actions such as creating additional transactions. The following code snippet provides an example:

```
function doStuffWithPeriod(periodId)
{
    var periodRecord = nlapiLoadRecord("accountingperiod", periodId);
```

```
if (period.getFieldValue("closed") == "F") // returns "T" if the period is closed in all books, "F" otherwise
{
    var journalRecord = nlapiCreateRecord("journalEntry");
    .
    .
    journalRecord.setFieldValue("postingperiod", periodId);
    .
    .
    .
    nlapiSubmitRecord(journalRecord); // fails if period is partially closed
}
}
```

Now, if the Extended Accounting Period Close Process feature is enabled, each accounting book has its own close status flag. However, the status accessible through SuiteScript remains shared, so that it only returns a value of **T** when the specified period is closed in all active books. If the period is partially closed, meaning closed in some but not all books, and that period ID is passed to a script like the above example, the script fails to execute successfully. This failure occurs because the script attempts to post a book generic transaction to all books, but the period is already closed in some of the books.

Enhancements to Token-based Authentication for RESTlets

Version 2015 Release 2 includes a new record type for generating consumer keys and secrets. This task is necessary if you use token-based authentication and want to provide application credentials to an external application that will call your RESTlets or send web services requests. As of Version 2015 Release 2, the new record type, called Integration, provides the **only** method of creating a consumer key and secret. The previous method, available at Setup > Integration > Token-Based Authentication, is no longer available. However, when your account is upgraded, existing token-based authentication application records are converted into integration records. Keys and secrets generated by using the former record are still valid.

The new integration record is available at Setup > Integration > Manage Integrations > New. In general, create one integration record for each external application that will call RESTlets or send web services requests.

A major benefit of the integration record is that it can be bundled and installed in another account. Integration records created from a bundle are installed with the corresponding consumer key and secret, although this data is not visible when you display the record in the UI.

The integration record also retains other benefits of the token-based authentication application record. For example, you can do all of the following:

- **View details about each application** – Each integration record shows details about the external application for which it was created. For example, you can enter a name and description of the application when you create the integration record.

- **Block a consumer key and secret** – You can use the integration record to block an application that is using the consumer key and secret created from that record. However, whereas in the previous record you used the Inactive checkbox, in the integration record, you use a dropdown list labeled Status. This field can be set to either Enabled or Blocked. Note that this action does not prevent the application from calling a RESTlet and authenticating using the NLAAuth method. As a best practice, do not supply NLAAuth details to applications that should be using token-based authentication.
- **View a list of all your applications** – When you create an integration record for each external application that integrates with your account, you can view a list of all your external applications at Setup > Integration > Manage Integrations.

Be aware that certain fields on the integration record do not react to RESTlet activity. For example, the Execution Log does not display details about RESTlet usage. Similarly, the check box labeled User Credentials does not affect the ability of the application to call RESTlets.

The Integration record type includes additional benefits for web services usage. For details, see [Better Tracking of Web Services Integrations](#).

For information about a related enhancement in Version 2015 Release 2, see [Enhanced Token-based Authentication Endpoint](#). For general details about token-based authentication, see the help topic [Token-based Authentication](#).

Upcoming Change for Scripting with Timesheets

The release of an updated scripting engine providing improved SuiteScript and SuiteFlow support for timesheets is targeted for a future release. It is recommended that you do not modify or create any scripts or workflows for timesheets until after the release of the new scripting engine.

New APIs

There are no new SuiteScript 1.0 APIs in Release 2015 Version 1.

Updated APIs

API	Notes
nlapiTriggerWorkflow (recordtype, id, workflowid, actionid, stateid)	See nlapiTriggerWorkflow Change for additional information.

2015.2 SuiteScript Records Browser

The 2015.2 version of the SuiteScript Records Browser is not yet available.

SuiteTalk (Web Services) Integration

Refer to the following sections for details on all SuiteTalk (Web Services) updates for Version 2015 Release 2:

- [Better Tracking of Web Services Integrations](#)
- [Token-based Authentication for SuiteTalk \(Web Services\)](#)
- [Release Preview Version of 2015.2 WSDL Now Available](#)
- [Araxis Merge Diff File for the 2015.2 Endpoint](#)
- [2015.2 SuiteTalk Schema Browser](#)

Better Tracking of Web Services Integrations

Version 2015 Release 2 introduces new enhancements for tracking and monitoring web services activity. These capabilities are useful if your account has multiple applications that send web services requests to NetSuite. For example, your sales department may use one application to send data about new customers. Your HR department may use a different application to update employee records. With the new enhancements, you can track and manage each application from the NetSuite UI.

These capabilities are made possible by a new record type, called Integration. Each application that sends web services requests or calls RESTlets can be represented by a separate integration record. With this approach, each web services request must include data that uniquely ties the sending application to an existing integration record.

Note: Some capabilities of the Integration record type are limited in relation to RESTlet activity. Additionally, if you currently use token-based authentication application records to represent applications that call RESTlets, these records are converted into integration records when your account is upgraded to Version 2015 Release 2. For details, see [Enhancements to Token-based Authentication for RESTlets](#).

Integration records are available in all accounts that have been upgraded to Version 2015 Release 2. However, until you upgrade to the 2015.2 endpoint, you are not required to take any actions. You have the option of letting your applications function as they did in previous releases.

However, if you upgrade to the 2015.2 endpoint, your account must have an integration record for each external account. More precisely, you must have a separate integration record for each application that sends requests by using the 2015.2 endpoint. You must also update your integrations so that they send data that uniquely identifies the sending application.

Regardless of which endpoint you use, you can begin taking advantage of the new enhancements right away. For more details, see the following topics:

- [Benefits of the Integration Record](#)
- [New Application IDs](#)
- [2015.2 WSDL Requirements](#)
- [About Default Web Services Integrations](#)
- [About the Auto-Installation Preference](#)
- [Effect on SSO Calls](#)
- [Transitioning for Customers](#)
- [Transitioning for Partners](#)

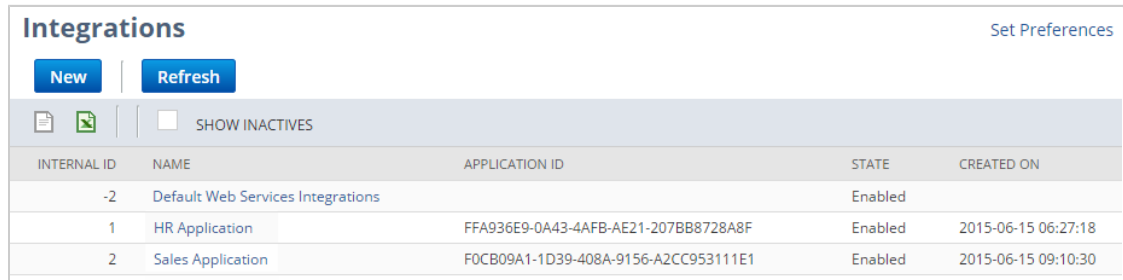
Benefits of the Integration Record

When you use integration records to represent all of your external applications, you can do any of the following:

- **View details about each external application** – Each integration record shows details such as the application's name and description.
- **Block an application** – You can use an integration record to block web services requests that come from the corresponding application. However, if you are using the 2015.1 or an earlier endpoint, be aware that some requests may be handled by the integration called [Default Web Services Integrations](#), which has its own configuration options. For details, see [About Default Web Services Integrations](#).
- **Select permitted authentication methods for the application** – If you upgrade to the 2015.2 endpoint, you can use the integration record to permit or disallow either of the following authentication options: user credentials and token-based authentication. For applications that send inbound SSO requests, see [Effect on SSO Calls](#). For details about the introduction of web services support for token-based authentication, see [Token-based Authentication for SuiteTalk \(Web Services\)](#). Note also that you can use the integration record to create consumer keys and secrets, which are required for token-based authentication. For details about a related enhancement in Version 2015 Release 2, see [Enhanced Token-based Authentication Endpoint](#).
- **Bundle an integration record** – You can bundle the application and install it in another account.
- **View an execution log specific to each application** – Previously, the only way of viewing SOAP requests was to use the Web Services Usage Log, which lists all requests together, regardless of which application sent them. The Web Services Usage Log remains

available at Setup > Integration > Web Services Usage Log. However, now you can use the integration record to view only those requests sent by a particular application.

- **View a list of all your applications** – You can view a list of all your applications at Setup > Integration > Manage Integrations.



INTERNAL ID	NAME	APPLICATION ID	STATE	CREATED ON
-2	Default Web Services Integrations		Enabled	
1	HR Application	FFA936E9-0A43-4AFB-AE21-207BB8728A8F	Enabled	2015-06-15 06:27:18
2	Sales Application	F0CB09A1-1D39-408A-9156-A2CC953111E1	Enabled	2015-06-15 09:10:30

New Application IDs

Every integration record has a unique application ID. This ID, a 32-character universally unique identifier, is generated when you create the record. These IDs are different from application IDs that may have been created in older versions of NetSuite.

Important: Old application IDs are permitted in requests that use the 2015.1 WSDL or earlier. They are not permitted in requests that use the 2015.2 WSDL. If they are included in requests that use the 2015.2 WSDL, the requests are denied.

2015.2 WSDL Requirements

Related to these enhancements, every web services request that uses the 2015.2 WSDL must include data that uniquely identifies the sending application. In all cases, this data is derived from a specific integration record. Each request must include one of the following:

- An application ID that was generated by an integration record, as described in [New Application IDs](#).
- Token-based authentication details.

If you have not yet upgraded to the 2015.2 WSDL, but you want to begin creating specific integration records to represent each of your applications, you must meet these same conditions. Each request must include data that is uniquely tied to an integration record, as described in the preceding list.

About Default Web Services Integrations

When your account is upgraded to Version 2015 Release 2, the system automatically creates an integration record called Default Web Services Integrations. This record represents all external

applications that are sending web services requests without a 32-character application ID or token-based authentication details.

Such requests are permitted only if the request uses the 2015.1 or a previous endpoint. These requests can include old application IDs created using a previous version of NetSuite. These requests are processed as they were in previous releases.

The Default Web Services Integrations record has many of the same qualities as an integration record created to track one specific external application. For example, the Default integration record includes an Execution Log that shows requests and responses. The Default integration record can also be blocked. However, the Default integration record cannot be used in conjunction with token-based authentication.

Exercise caution when editing the Default Web Services Integrations record. Before you create integration records to represent your external applications, the Default integration record represents all of your web services traffic.

About the Auto-Installation Preference

Auto-installation is the automatic creation of an integration record in your account. For example, in some cases, a partner may provide you with an updated version of an application. This update may include a new application ID created by using the integration record. The first time you send a request using this application, a new integration record is created in your account.

The **Require Approval during Auto-Installation of Integration** preference affects whether this new record is automatically enabled. If the preference is set to false, the State field on the new application is automatically set to Enabled, and all requests are permitted. If the preference is set to true, the State field on the new integration record is set to Waiting for Approval. In the latter case, you must manually edit the record and set the State to Enabled. Until you do, all requests sent by that application are blocked.

You can manage the new preference at Setup > Integration > Web Services Preferences.

Effect on SSO Calls

If you are using web services for single sign-on (SSO), and your requests use the 2015.1 WSDL or earlier, be aware of the following:

- Outbound single sign-on (SuiteSignOn) requests are not affected.
- Inbound single sign-on requests are treated exactly as requests that use user credentials to authenticate. You can block these requests by using the Default Web Services Integration record. You can also use the default integration record to view an execution log of these requests. If you prefer, you can create specific integration records for each application that sends inbound SSO requests. Check the User Credentials box to permit the application to authenticate by using inbound SSO.

Check back for details about how these enhancements affect SSO calls that are made by using the 2015.2 WSDL.

Transitioning for Customers

If you have created web services integrations for your own use, you can begin using the integration record right away. However, note that the token-based authentication option will not be supported until the 2015.2 WSDL becomes generally available.

For each of your external applications, you can create an integration record at Setup > Integration > Manage Integrations > New. Make a note of the application ID. The ID is automatically generated when you create the record. Then update the external application. You must modify the application so that when it generates SOAP requests, the appropriate application ID is included in the header of each request. You can make this change by using the ApplicationInfo class.

Documentation about sending token-based authentication details is targeted to be available when the 2015.2 WSDL is released.

Transitioning for Partners

If you are a partner, for each application you provide, follow the guidance described in [Transitioning for Customers](#). Then provide your customers with an updated version of the application that includes the required application information.

When you provide customers with your update, you may want to provide them with instructions about checking or setting the **Require Approval during Auto-Installation of Integration** preference, which is described in [About the Auto-Installation Preference](#).

Note also that you can include integration records in a bundle, which may be useful if you want to share additional customizations associated with the application.

Token-based Authentication for SuiteTalk (Web Services)

Note: This feature is targeted to be available for all endpoints when the 2015.2 WSDL (endpoint) becomes generally available. This feature will not be supported for endpoints earlier than 2015.2.

Token-based authentication is being extended to support SuiteTalk (web services), de-coupling web services authentication from password expiration. With this feature, client applications will be able to access web services using a token, significantly reducing the risk of compromising user credentials.

Release Preview Version of 2015.2 WSDL Now Available

A release preview version of [SuiteTalk Web Services version 2015.2](#) is now available.

When your account is upgraded to Version 2015 Release 2, SuiteTalk Web Services version 2015.2 becomes available in your live production account but is considered a beta released product. During the time that the WSDL is in beta, changes may be made that are backward incompatible. It is recommended that you use the beta version for testing and development purposes, but wait to upgrade your client applications to the generally available version of the WSDL when it becomes available.

You are able to access the new WSDL during your release preview period. Click [here](#) to view the 2015.2 Release Preview (beta) WSDL.

Araxis Merge Diff File for the 2015.2 Endpoint

The Araxis Merge Diff file will highlight 2015.1 - 2015.2 schema changes. The diff file is not yet available, but this section provides details about the types of information it will include.

About the Araxis Diff File

NetSuite publishes an Araxis Diff File for every new endpoint. Depending on the scope of the release, changes described in the Araxis Diff File may include:

- New record types
- Elements (fields) that have been added, removed, or changed
- New sublists (lists)
- New joins
- New search filters and search return columns
- New platform operations

Using the Araxis Diff File

Important: The figures in this section are for illustrative purposes **only**. They do not reflect changes related to any specific endpoint.

The SuiteTalk diff file is generated using Araxis Merge. When you click the diff file link (when it is available), a page will appear that offers an overview of all schema changes.

The following screenshot shows how this summary page will look. In this example, the rows highlighted in purple indicate that an XSD has been modified. The integers at the center indicate the number of changed lines in the file. To see the exact differences, click the Report link in the column at the right (or click the name of the file, in either column).

activities.scheduling.xsd	3	activities.scheduling.xsd	Report 1
activities.schedulingTypes.xsd	1	activities.schedulingTypes.xsd	Report 2
documents.fileCabinet.xsd	0	documents.fileCabinet.xsd	Report 3
documents.fileCabinetTypes.xsd	2	documents.fileCabinetTypes.xsd	Report 4
general.communication.xsd	3	general.communication.xsd	Report 5
general.communicationTypes.xsd	0	general.communicationTypes.xsd	Report 6
lists.accounting.xsd	32	lists.accounting.xsd	Report 7
lists.accountingTypes.xsd	5	lists.accountingTypes.xsd	Report 8
lists.employees.xsd	0	lists.employees.xsd	Report 9
lists.employeeTypes.xsd	0	lists.employeeTypes.xsd	Report 10
lists.marketing.xsd	0	lists.marketing.xsd	Report 11
lists.marketingTypes.xsd	0	lists.marketingTypes.xsd	Report 12
lists.relationships.xsd	10	lists.relationships.xsd	Report 13
lists.relationshipTypes.xsd	1	lists.relationshipTypes.xsd	Report 14
lists.supplyChain.xsd	0	lists.supplyChain.xsd	Report 15
lists.supplyChainTypes.xsd	0	lists.supplyChainTypes.xsd	Report 16
lists.support.xsd	2	lists.support.xsd	Report 17
lists.supportTypes.xsd	0	lists.supportTypes.xsd	Report 18

When you click any of these links, the diff file shows a page with two columns. The old version of the file is shown at the left, and the new version at the right. Changed elements are highlighted.

For example, the following screenshot shows an excerpt from the report on platform.commonTypes.xsd. Note the addition of two values to the PermissionCode enumeration: `_accessPaymentAuditLog` and `_advancedPDFHTMLTemplates`.

```

500 <!-- PermissionCode/start -->
501 <simpleType name="PermissionCode">
502   <restriction base="xsd:string">
503     <enumeration value="_accountDetail"/>
504     <enumeration value="_accountingBook"/>
505     <enumeration value="_accountingLists"/>
506     <enumeration value="_accounts"/>
507     <enumeration value="_accountsPayable"/>
508     <enumeration value="_accountsPayableGraphing"/>
509     <enumeration value="_accountsPayableRegister"/>
510     <enumeration value="_accountsReceivable"/>
511     <enumeration value="_accountsReceivableGraphing"/>
512     <enumeration value="_accountsReceivableRegister"/>
513     <enumeration value="_accountsReceivableUnbilled"/>
514     <enumeration value="_adjustInventory"/>
515     <enumeration value="_adjustInventoryWorksheet"/>
516     <enumeration value="_admindocs"/>
517     <enumeration value="_adpImportData"/>
518     <enumeration value="_adpSetup"/>
519     <enumeration value="_advancedAnalytics"/>
520     <enumeration value="_allocationSchedules"/>
521     <enumeration value="_allowNonDLChanges"/>
500 <!-- PermissionCode/start -->
501 <simpleType name="PermissionCode">
502   <restriction base="xsd:string">
503     <enumeration value="_accessPaymentAuditLog"/>
504     <enumeration value="_accountDetail"/>
505     <enumeration value="_accountingBook"/>
506     <enumeration value="_accountingLists"/>
507     <enumeration value="_accounts"/>
508     <enumeration value="_accountsPayable"/>
509     <enumeration value="_accountsPayableGraphing"/>
510     <enumeration value="_accountsPayableRegister"/>
511     <enumeration value="_accountsReceivable"/>
512     <enumeration value="_accountsReceivableGraphing"/>
513     <enumeration value="_accountsReceivableRegister"/>
514     <enumeration value="_accountsReceivableUnbilled"/>
515     <enumeration value="_adjustInventory"/>
516     <enumeration value="_adjustInventoryWorksheet"/>
517     <enumeration value="_admindocs"/>
518     <enumeration value="_adpImportData"/>
519     <enumeration value="_adpSetup"/>
520     <enumeration value="_advancedAnalytics"/>
521     <enumeration value="_advancedPDFHTMLTemplates"/>
522     <enumeration value="_allocationSchedules"/>
523     <enumeration value="_allowNonDLChanges"/>
    
```

The diff file also shows changes to data types. The following screenshot is a snapshot of transactions.sales.xsd. Note that the chargeType element changed from an enumeration to a RecordRef.

```

1054 <element name="voiceDelivered" type="xsd:boolean" minOccurs="0"/>
1055 <element name="chargeType" type="platformCommonType:ChargeRuleType" minOccurs="0"/>
1056 <element name="chargeList" type="platformCore:RecordRefList" minOccurs="0"/>
1057 <element name="customFieldList" type="platformCore:CustomFieldList" minOccurs="0"/>
1054 <element name="voiceDelivered" type="xsd:boolean" minOccurs="0"/>
1055 <element name="chargeType" type="platformCore:RecordRef" minOccurs="0"/>
1056 <element name="chargeList" type="platformCore:RecordRefList" minOccurs="0"/>
1057 <element name="customFieldList" type="platformCore:CustomFieldList" minOccurs="0"/>
    
```

2015.2 SuiteTalk Schema Browser

The 2015.2 SuiteTalk Schema Browser is not yet available.

SuiteCloud IDE

SuiteCloud IDE is an Eclipse-based integrated development environment (IDE) that is packaged for NetSuite platform development.

Version 2015 Release 2 includes the following enhancement to SuiteCloud IDE features.

Support for SuiteScript 2.0

As of Version 2015 Release 2, the SuiteCloud IDE includes support for the SuiteScript 2.0 feature. SuiteCloud IDE support for SuiteScript 2.0 includes the following features:

- [Create SuiteScript 2.0 Project](#)
- [Create SuiteScript 2.0 Files](#)
- [Switch SuiteScript Version in Projects](#)

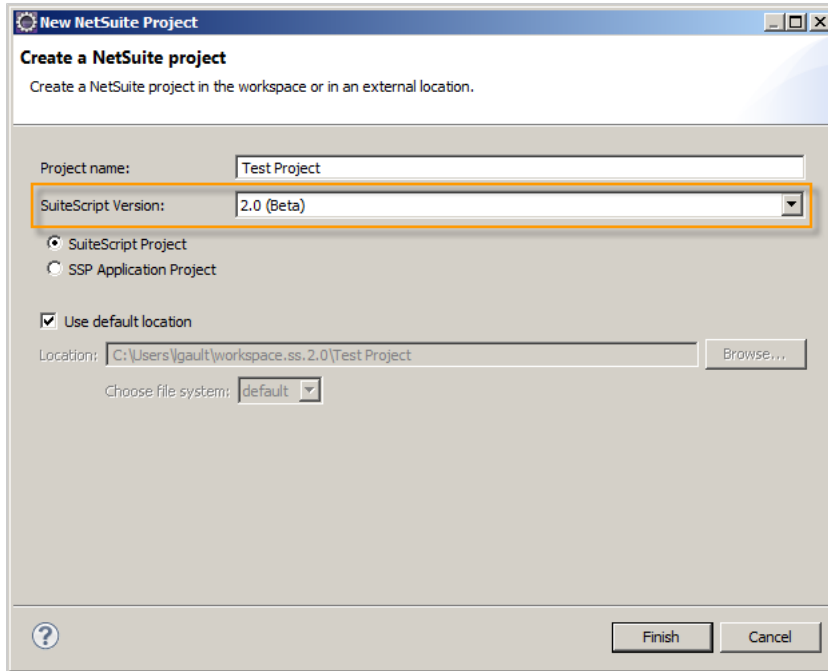
You can also use the debugger, available through the SuiteCloud IDE, to debug server-side SuiteScript 2.0 scripts.

Important: SuiteScript 2.0 is a beta feature in Version 2015 Release 2. As a result, the SuiteCloud IDE does not validate SuiteScript 2.0 scripts. See [Switch SuiteScript Version in Projects](#).

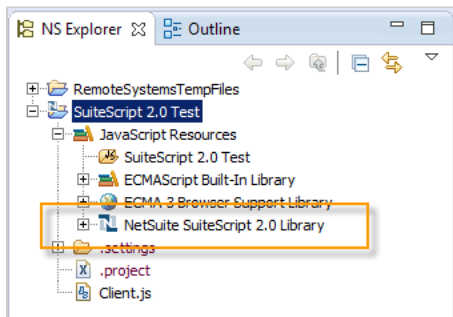
Create SuiteScript 2.0 Project

Go to File > New > NetSuite Project in the SuiteCloud IDE to open the New NetSuite Project window and select **2.0 (Beta)** for the **SuiteScript Version** property.

The following screenshot shows the **SuiteScript Version** property in the New NetSuite Project window:



When you create a SuiteScript 2.0 project, the project includes the SuiteScript 2.0 Library:

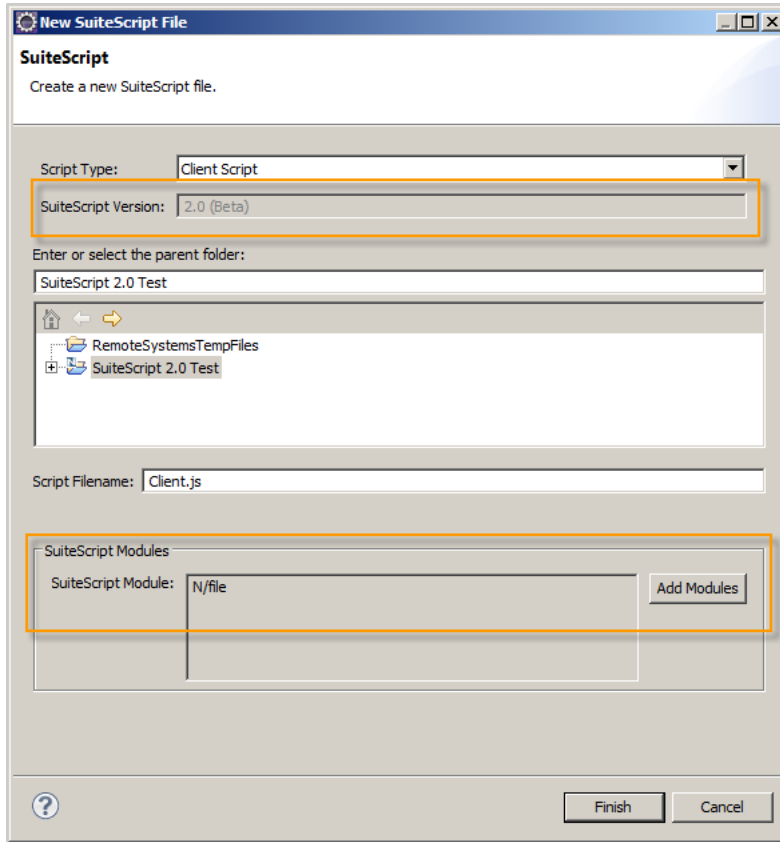


Note: After you create the project, you can use the Change Project Settings window to change the project version after you create the project. See [Switch SuiteScript Version in Projects](#).

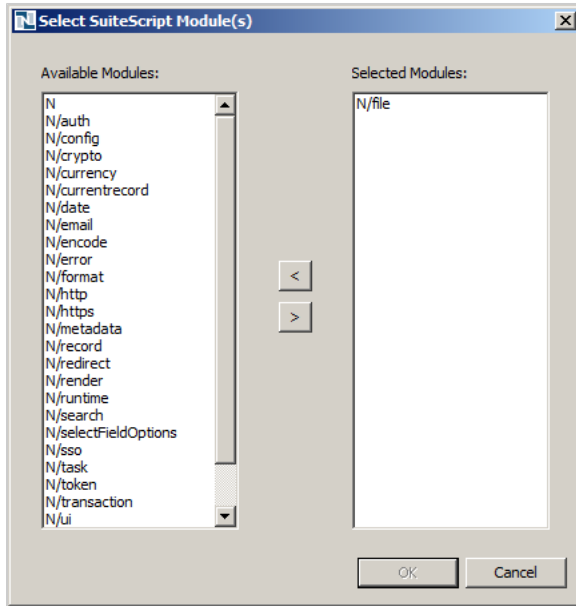
Create SuiteScript 2.0 Files

Go to File > New > SuiteScript File to create script files for all supported SuiteScript 2.0 script types. The SuiteCloud IDE includes SuiteScript 2.0 templates for supported script types. In addition, you can select the modules to include in the `define` statement in the script file.

The following screenshot shows the New SuiteScript File window for a Client Script that includes the file module:



Click **Add Modules** in the New SuiteScript File window to select modules:



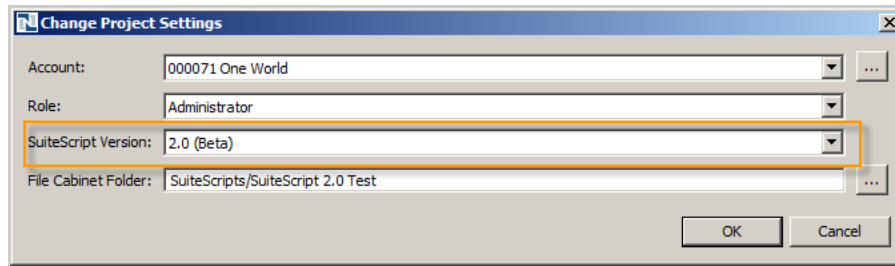
Switch SuiteScript Version in Projects

You can switch project settings between SuiteScript version 1.0 and SuiteScript version 2.0 in the SuiteCloud IDE. Switching the SuiteScript version changes the library used by the project. The SuiteCloud IDE uses the associated library to perform code completion and validation.

The following table lists the library versions and SuiteCloud IDE behavior for each SuiteScript version in Version 2015 Release 2:

SuiteScript Version	SuiteScript Library	SuiteCloud IDE Behavior
1.0	NetSuite SuiteScript 1.0 Library	Provides code completion for the SuiteScript API functions, objects, and internal IDs. Validates SuiteScript 1.0 code.
2.0	NetSuite SuiteScript 2.0 Library	Provides code completion for SuiteScript 2.0 module objects, methods, and enums. Does not provide validation of SuiteScript 2.0 code.

Go to NetSuite > Change Project Settings to change the SuiteScript version and other project settings. The following screenshot shows the **SuiteScript Version** dropdown in the Change Project Settings window:



SuiteApps (Bundles) Released by NetSuite

Version 2015 Release 2 includes the following enhancements to SuiteApps released by NetSuite:

- Recurring Billing Enhancements
- Enhancements to Certificate of Creditable Tax Withheld at Source (Form 2307)
- EU Sales List Support for Triangulation of Goods
- Procurement Dashboard
- Warranty and Repairs Management Enhancements
- Fixed Assets Management Enhancements
- StickyNotes User Experience Improvements
- Basic API Integration on SuiteSocial News Feed
- Email Capture Plugin Support for SuiteSocial
- Enhancements to Payment Gateway Plugin
- Japan-Specific Enhancements and Changes
- Application Performance Management (Beta)

Recurring Billing Enhancements

NetSuite Version 2015 Release 2 includes the following enhancements to the Recurring Billing SuiteApp:

- Enhancements to Multi-Queue Support for Recurring Billing Batch Processes
- Enhancements to Billing Forecast Report
- Rollback Bill Date

Enhancements to Multi-Queue Support for Recurring Billing Batch Processes

In Recurring Billing version 15.2, the charge creation and subscription creation processes are enhanced to support multi-queue implementation.

Charge Creation

Similar to the billing and usage-based rating processes, the charge creation process is implemented using a parent script (SWV_SB_SS_CCScheduler) and a worker script (SWV_SB_SS_CreateCharges).

Subscription Creation

The implementation for the subscription creation process does not use a parent script. Instead, the process creates independent Process Queue Batch records for subscription item creation and charge creation, and deploys the corresponding worker scripts to process the batches.

- SWV_SB_SS_SecondarySubItemGenerator – worker script that processes the independent Process Queue Batch records for secondary subscription item creation
- SWV_SB_SS_CreateCharges – worker script that processes the independent Process Queue Batch records for charge creation

Other Enhancements

Other enhancements to the Recurring Billing batch process implementations and Multi-Queue feature include:

- Changes to Subscription Billing Preferences fields:
 - Processing Mode field – If the Processing Mode preference is set to Data Migration, only the first charge will be created for all subscription items during subscription creation. Previously, a maximum of 12 charges could be generated for each subscription item.
 - Dedicated Queues field – The maximum number of dedicated queues that can be used for all batch processes is total queues minus one. Previously, billing could use a maximum of 12 dedicated queues.
- Manual Runs for Rating and Charge Creation – You can manually run the REScheduler and CCScheduler scripts to process specific subscriptions or subscription items. Use the scheduler script deployment record for manual runs (Status: Not Scheduled) and specify a value for either the Subscription ID or Secondary Sub Item ID parameter.
- Removal of obsolete scripts – The following scripts have been removed from the bundle starting with Recurring Billing version 15.2:
 - SWV_SB_SS_CreateNextSubscriptionCharge – This script is replaced by the SWV_SB_SS_CCScheduler script.

- SWV_SB_SS_InitializeSubscription and SWV_SB_SS_CreateSubscriptionItem – These scripts are replaced by the scheduled scripts in the enhanced subscription creation process.

Note: If you have any customizations around these obsolete scripts, please make sure that they are updated accordingly.

For more information, see the help topic [Multi-Queue Support for Recurring Billing Batch Processes](#).

Enhancements to Billing Forecast Report

Recurring Billing version 15.2 includes enhancements to the Billing Forecast Report.

To generate the billing forecast report, the Advanced PDF/HTML Templates feature must be enabled.

Enhancements to the billing forecast reports include:

- Customer and Subscription fields are added as additional filters for the report.
- Group by Primary Subscription Item and Group by Secondary Subscription Item are added to the report display options.
- The Billing Forecast report is generated in PDF file format.
- An option to save an XML file of the generated report is available.

For more information, see the help topic [Generating Billing Forecast Reports](#).

Rollback Bill Date

This feature enables you to rollback the bill dates of subscriptions that you want to redo billing for.

Starting with Recurring Billing version 15.2, you can perform bill date rollback by mass update or by clicking the Rollback Bill Date button on the subscription record.

Before performing bill date rollback, you must first delete the previous period's billing transactions. Bill date rollback is valid only for billed subscriptions that meet the following conditions:

- Subscription is active.
- Subscription status is Open.
- The subscription's Last Billed Date field is not empty (that is, the subscription has been billed at least one time).

Upon successful completion of the bill date rollback, the following records are updated:

- The subscription's Last Billed Date and Next Billing Date fields will be rolled back by one bill date.
- The subscription item's Last Item Billing Date and Next Item Billing Date fields will be rolled back by one period.
- Non-billable charges with zero amount for the previous period will have its charge stage set back to Ready.

For more information, see the help topic [Rolling Back the Subscription Bill Dates](#).

Enhancements to Certificate of Creditable Tax Withheld at Source (Form 2307)

The Withholding Tax SuiteApp includes the following enhancements to the Philippine Certificate of Creditable Tax Withheld at Source (Form 2307):

- Select Transactions – This button enables you to select which transactions will be included in the report.
- Print Individual 2307 – This button enables you to print individual Form 2307 per transaction.

The Select Transactions and Print Individual 2307 buttons are displayed on the Withholding Tax Report page when you select a Philippine subsidiary and Form 2307 as the report type.

For more information, see the help topic [Withholding Tax Reports - Philippines](#).

EU Sales List Support for Triangulation of Goods

Note: This feature is targeted to be available in August 2015.

This feature enables you to identify goods involved in European Union (EU) triangulation. To mark a transaction as an EU triangulation deal, check the EU Triangulation box on the transaction record.

When you generate a localized or generic EU Sales List report, the default indicator for transactions marked as EU Triangulation will be Triangulated Goods.

The EU Sales List report is provided by the International Tax Reports SuiteApp.

Procurement Dashboard

With Version 2015 Release 2, you can monitor procurement-related transactions using the Procurement Dashboard. It transforms select information to provide status reminders, expenditure reports, vendor performance, and other key business metrics.

By default, the Procurement Dashboard is available in the Classic, Accounting, and Shipping centers, to the following roles:

- Administrator – With the Administrator role, you can set up and publish the dashboard to other roles.
- Full Access – The dashboard is organized and set up in accounts with the Full Access role.

Required features and preferences must be enabled to display portlet content on the dashboard. You can view the components or portlet content related to transactions to which you have access. You can customize the dashboard by adding your own portlets. Your role can be set up to publish your custom Procurement Dashboard to other users within your center.

Warranty and Repairs Management Enhancements

Using the current Warranty and Repairs Management, you can register a warranty only during invoice creation. More options for registration and claim processing are introduced in the latest release, including the following:

- Standalone registration and claim processing
- Import of warranty registration details
- Registration from an item
- Creation of claims from a warranty registration
- Advanced shipping of a replacement item
- Use of Global Search to retrieve a warranty registration record

These enhancements enable you to manually enter a registration and import existing registration details. The enhancements also include settings to automate specific processes related to claims and the return authorization.

Fixed Assets Management Enhancements

- [Enhancements to the Reporting Tool](#)
- [Ability to Propose and Generate Assets from Check Transactions](#)
- [Require Approval for Depreciation Journal Entry](#)
- [Track Asset Proposal Progress on the Transaction Status Page](#)

Enhancements to the Reporting Tool

In Fixed Assets Management version 4.02, the format for FAM reports will be changed from HTML to Advanced PDF. This change will enable users to download the PDF report or export

it to an XML file. Users will also be able to print the reports or send them directly to an email address.

To generate FAM reports, you must enable the **Advanced PDF/HTML Templates** feature in your account.

Ability to Propose and Generate Assets from Check Transactions

Customers that are using check payments to record purchases, can now propose and generate assets from check transactions.

Require Approval for Depreciation Journal Entry

If a user runs asset depreciation, but does not have permission to approve journal entries, the system will require approval for the resulting journal entry before it is posted.

The **Require Approvals on Journal Entries** preference must be enabled in your account.

Track Asset Proposal Progress on the Transaction Status Page

Clicking the **Generate Asset** button on the asset proposal page will redirect users to the transaction status page. From the transaction status page, you will be able to track the progress of the asset proposal and see the number of assets generated from the proposal.

StickyNotes User Experience Improvements

The latest release of StickyNotes includes the following enhancements to the user interface design:

- The note expands when you click on it to display the entire content. Expanding note replies will also expand the size of each reply note to display all the contents.
- Administrators can disable and enable the in-app tutorial for the entire account with the **Suppress tutorial on this Account** preference.

Basic API Integration on SuiteSocial News Feed

The SuiteSocial API lets you integrate SuiteSocial data into third-party applications and services. This API uses a RESTful interface to retrieve data from the SuiteSocial news feed, and view it in another application. With this version of the API, you can retrieve the following information:

- Posts accessible to a SuiteSocial user or users
- List of active SuiteSocial users

- Employees that a specific user is following
- List of a specific user's followers
- List of all the channels in SuiteSocial

Email Capture Plugin Support for SuiteSocial

The email capture feature lets users reply to SuiteSocial notification email messages to respond to a post, comment, or private message in SuiteSocial. Users may include a single attachment to add a file to their post on SuiteSocial. This feature does not apply to notification email messages for subscribed records, status changes, and digests.

Enhancements to Payment Gateway Plugin

The Payment Gateway Plugin introduces the following enhancements:

- Support for pre-authorization for eWAY Rapid payment gateway.
- Support for recurring payments, which enables you to bill a customer's card with a fixed amount on a regular basis for recurring services. This is available to SecurePay and eWAY payment gateways.
- Support for 3D Secure redirect for eWAY, which provides more secure transactions by requiring authentication from a customer's bank before payment is processed.

Japan-Specific Enhancements and Changes

The Japan Localization SuiteApp update includes the following enhancements and changes:

- Item Detail Statement (IDS) Enhancements and Changes
 - Enhanced user Interface for Generating an IDS
 - Ability to Regenerate an IDS for a Customer
 - More Transactions Included in an IDS
 - Tracking Changes in the IDS Record
 - Ability to Define Auto-Generated IDS Numbering
 - IDS Collection Calendar Report
 - Multiple Payment Terms to Manage a Japanese Customer's Billing Cycle
 - Changes to the IDS File Naming and File Cabinet Folder
 - Changes to IDS Template Format and File Cabinet Folder

- Changes to the Japanese Holiday Checking for Payment Due Date
- Changes to the Japanese Custom Address Form and Zip Code Lookup
- Discontinued Japanese Custom Forms

Important: With this SuiteApp update, the bundle ID of the Japan Localization SuiteApp is changed from 15580 to 82168. The SuiteApp is a shared bundle. For new installations, contact Support to get access to the SuiteApp.

Item Detail Statement (IDS) Enhancements and Changes

Following are the item detail statement enhancements and changes included in the Japan Localization SuiteApp.

Enhanced user Interface for Generating an IDS

The new user interface for generating an item detail statement includes a summary page and a drill down page. The summary page shows each customer's total number of transactions and total amounts grouped by transaction type (invoice, sales order, and credit memo). A View Transactions link opens a drill down page that shows the list of the customer's transactions for each transaction type. You can also click a transaction to view or edit it.

See the help topic [Generating a Japanese Item Detail Statement](#).

Ability to Regenerate an IDS for a Customer

Previously, the item detail statement was saved in the system as a PDF file only. Now, the item detail statement is also saved as a custom transaction record. If multiple customers were included in the original IDS PDF document, the system creates multiple IDS records (one record for each customer). You can go to the IDS record to regenerate an IDS for a customer.

Regenerating an item detail statement is recommended when the amount, quantity, or unit price was changed in a customer's transaction after generating the original IDS document. It is also recommended when a new transaction was created for the same period after the original IDS document was already generated.

When you click **Regenerate** on an IDS record, the system creates a new IDS PDF file with updated amounts. It also adds the new IDS PDF link to the **Regenerated Item Detail Statement** field of the IDS record. You can click the download link to save the file to your computer.

See the help topic [Regenerating an Item Detail Statement for a Customer](#).

More Transactions Included in an IDS

Previously, there was a limit to the number of transactions that can be included in an item detail statement, and only invoices were supported.

Now, you can include as many invoices as required. Credit memos and sales orders are automatically included. The system can now also support multiple currencies in one item detail statement. You also have an option to include overdue invoices from previous item detail statements.

Because of this increased scope, the IDS record and PDF file are generated using a scheduled script. When IDS processing is completed, the system sends you an email message that includes a link to the PDF file.

Note that IDS processing time and page loading time on the IDS Generation page depend on the volume of transactions.

Tracking Changes in the IDS Record

As a custom transaction type, the item detail statement is now saved as a record, with a System Notes subtab that tracks changes.

Ability to Define Auto-Generated IDS Numbering

You can now set up your item detail statement to use different numbering systems per subsidiary or location.

Only an Administrator role can define the auto-generated numbering for item detail statements.

1. On the **Document Numbers** subtab of the **Auto-Generated Numbers** page, locate the Item Detail Statement transaction type and enter values in the columns.
2. If you want to continue your current numbering scheme, enter the initial number for the numbering sequence. The initial value cannot be set to a number lower than the number shown in the **Current Number** column. The initial number does not apply if you enable subsidiary or location auto-generated numbering.
3. If you want to set up auto-numbering by subsidiary or location, check the **Use Subsidiary** box or the **Use Location** box.

Clear the **Allow Override** box to ensure that no duplicate transaction numbers are used. By default, this box is checked to enable you to edit the numbers on records.

See the help topic [Setting Up Auto-Generated Item Detail Statement Numbers](#).

IDS Collection Calendar Report

The IDS Collection Calendar Report shows how much funds the company can receive from customers based on all of the item detail statements for the period, so that accountants can see whether sufficient funds will become available for the company's vendor payments. This report is provided as a saved search. Currently, the report does not filter by subsidiary.

See the help topic [IDS Collection Calendar Report](#).

Multiple Payment Terms to Manage a Japanese Customer's Billing Cycle

Previously, you could define only one set of payment terms for a Japanese customer. In the latest version of the Japan Localization SuiteApp, Japanese customer records have a **Payment Terms** subtab where you can set up multiple payment terms.

Upon creation of an invoice, the system will set the nearest closing date and its corresponding payment due date according to the payment terms defined on the customer record. If you want to move the transaction to a future billing period, you can change the closing date to a future date. The system will automatically calculate its corresponding payment due date, based on the available payment terms on the customer record.

Note: The automatic calculation of the payment due date based on the selected closing date does not prevent you from changing the payment due date.

The **Use Item Detail Statement** box is now on the **Payment Terms** subtab of the customer record. With this change, the box is now clear by default. An error message prevents you from saving the customer record with the **Use Item Detail Statement** box checked unless at least one set of payment terms is entered.

See the help topic [Japanese Billing Cycle and Payment Terms](#).

Changes to the IDS File Naming and File Cabinet Folder

IDS PDF file names are system-generated using a new naming convention. The files are automatically saved in the Item Detail Statements folder in the file cabinet. You can go to Documents > Files > File Cabinet and open the Item Detail Statements folder to access an IDS document. A link to the PDF file is also included in the email message that NetSuite sends you when an IDS generation is completed.

Changes to IDS Template Format and File Cabinet Folder

The item detail statement enhancements support only IDS templates created in FreeMarker syntax. This change requires you to enable the Advanced PDF/HTML Templates feature and to upload new custom templates created in FreeMarker syntax.

Templates must be uploaded to the IDS Templates folder inside the Bundle 82168 folder in the file cabinet. Go to Documents > Files > File Cabinet, and click the IDS Templates folder located inside Bundle 82168 in the SuiteBundles folder, and add your template files.

To use your custom template, be sure to select it in the **Default Item Detail Statement Template** field of the Subsidiary or Company Information page for Japanese companies. Otherwise, the default template provided by NetSuite will be automatically selected on the IDS Generation page.

Changes to the Japanese Holiday Checking for Payment Due Date

The web service that checks if the due date you selected is a Japanese holiday is no longer being used in the Japan Localization SuiteApp.

The new implementation of the holiday checking for payment due date introduces a setup page for non-operational days or holidays. Japanese holidays for 2015 are already provided by the SuiteApp. You must create holiday records for subsequent years. You can also define any date as a non-operational day.

The **Use Holiday Checking** box is checked by default on the Company Information page for non-OneWorld accounts, and on Japanese subsidiary records for OneWorld accounts. These defaults ensure that if the payment due date you selected on an invoice is a holiday, the system shows an alert message. Payment due dates are validated by the system according to the holidays that have been set up for your company or subsidiary.

The holiday checking applies only on invoices that have the **Include in Item Detail Statement** box checked.

Before the system can validate the payment due date, an administrator must perform the following steps:

1. Set up holiday display names and their translations:
 - a. Go to Customization > Lists, Records, & Fields > Lists, and click **Holiday Display Name List**.
 - b. To add a holiday name, enter a holiday name in the **Values** column under the **Values** subtab, and click **Add**.
 - c. If the Multi-Language feature is enabled, enter translations in the **Translation** column, and click **Done**. The availability of the language fields for entering translations depends on the languages you have set up on the General Preferences page.
 - d. Click **Save**.
2. Create holiday records by going to Setup > Holidays > Holiday > New. Holiday names defined on the Holiday Display Name List page will be available for selection in the **Display Name** field on the Holiday record. The holiday record defines the date of the holiday (day, month, and year).
3. Assign holidays to Japanese subsidiaries:
 - a. Edit the Japanese Subsidiary page.
 - b. Verify that the **Use Holiday Checking** box is checked.
 - c. Click the **Holidays** subtab, and add the holidays that you want the subsidiary to use for validating the payment due date. You can also remove holidays from the

Holidays subtab. If you do not want a holiday to be available to any subsidiary, you can edit the holiday record, and either delete it or make it inactive.

Note: In non-OneWorld Japanese accounts, a holiday will automatically be added to the **Holidays** subtab of the Company Information page after you save a holiday record. To remove a holiday from a non-OneWorld account, you must delete the holiday record.

For more information and best practice recommendations, see the help topic [Setting Up Japanese Holidays to Validate Payment Due Dates](#).

Changes to the Japanese Custom Address Form and Zip Code Lookup

The Japanese address form provided by the Japan Localization SuiteApp uses the new custom address form format. Its name has also been changed from **Japan 15580 Address Form** to **NS Japan Address**.

An account administrator or a user with the Custom Address Form permission must go to Customization > Forms > Address Forms, edit the custom address form named **NS Japan Address** to assign it to Japan, and click **Save**. This step is required to enable the system to display the correct form on Japanese records.

Note: For SuiteApp updates, you will need to repeat this step. The current system behavior is to revert to the default address template each time the SuiteApp is updated in your account.

In addition to this change to the address form, the feature that auto fills the Japanese address fields based on the zip code you enter is currently not available.

Discontinued Japanese Custom Forms

Japan-specific fields are now shown on standard forms if the customer's country is Japan. As a result of this change, the following custom forms are no longer available in the Japan Localization SuiteApp:

Note: Existing customer and invoice records that were created using the custom jp4030 customer and invoice forms are transitioned to the standard forms automatically during the SuiteApp update.

- jp4030-customer-form (customer form)
- jp4030-invoice-duedate (invoice form)
- JP- Service Invoice
- JP- Standard Sales Order
- JP- Standard Estimate

- JP- Pay bills
- JP- Product Invoice (Tax inclusive)
- JP- Standard Sales Order - Invoice
- JP- Accept Payments
- JP- Standard Estimate (Tax inclusive price)
- JP- Sales Order - Cash Sales (Tax inclusive)
- JP- Standard Sales Order - Invoice (Tax inclusive)
- JP- Standard Sales Order (Tax inclusive)
- JP- Professional Invoice
- JP- Sales Order - Cash Sales
- JP - Expense Reports
- JP- intermediate Invoice
- JP-Receipt Layout
- JP- Transaction Layout
- JP-Disclaimer Transaction Layout
- JP-Payment Slip
- Picking Card Layout (Gothic)
- JP-AR Aging Invoice Layout
- JP- Manager Inventory Item
- JP - Non-inventory Item Form
- JP- Manager Group
- JP- Group Item Form
- JP - Discount Form
- JP- Manager Non-inventory Item
- JP - Inventory Item Form

Application Performance Management (Beta)

The Application Performance Management (Beta) SuiteApp supports monitoring and management of NetSuite performance for business critical operations. Account administrators can use this SuiteApp to review the performance of the NetSuite user interface, including customizations such as scripts and workflows. This SuiteApp compiles information into a

performance dashboard that serves as a starting point for performance troubleshooting. This dashboard can be configured to filter and sort performance data for selected time periods and record pages. Visual indicators on the dashboard highlight performance issues and outliers, with more details available in portlets and quick links. Additional context for performance analysis is provided by a page time summary, script analysis, and performance data visualizations in a variety of chart types.

This SuiteApp supports the following tasks:

- Identification of performance opportunities or degradation, and prioritization of issues based on usage and traffic.
- Viewing of performance metrics for important record pages and assessment of system health and trends.
- Drilling down to performance details for specific record types, operations, and instances.
- Viewing of response times allocated by client, server, and network.
- Performance monitoring of user event scripts, workflows, RESTlets, scheduled scripts, and Suitelets.

Important: The Application Performance Management (Beta) SuiteApp currently is being made available to a limited set of NetSuite customers and partners. If you are interested in obtaining access to this SuiteApp, please contact your Account Manager.