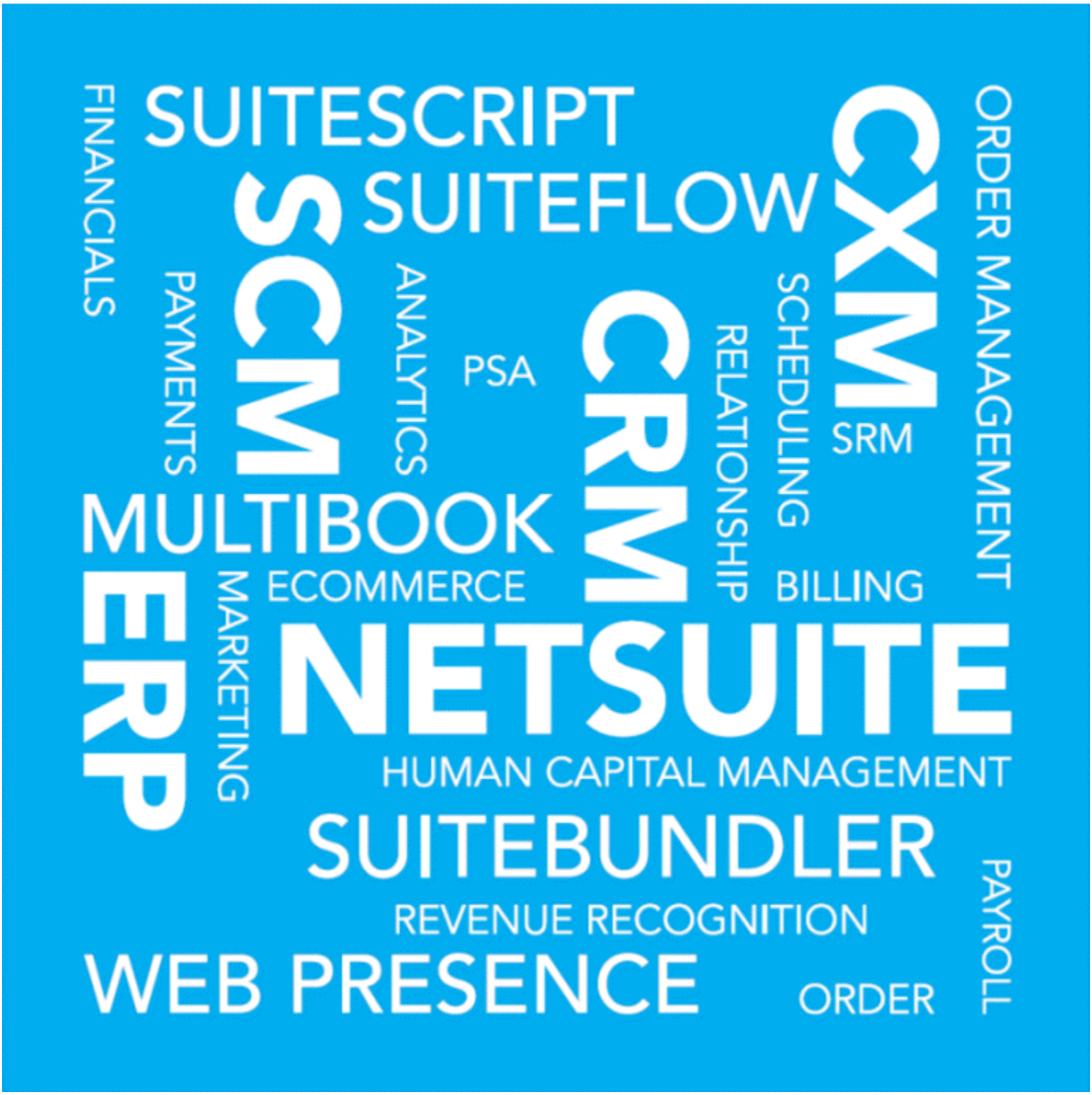
SuiteScript General Reference



Copyright © 2005, 2018, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

If this document is in public or private pre-General Availability status:

This documentation is in pre-General Availability status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.

If this document is in private pre-General Availability status:

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your pre-General Availability trial agreement only. It is not a commitment to deliver any material, code, or functionality,

and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Master Agreement, Oracle License and Services Agreement, Oracle PartnerNetwork Agreement, Oracle distribution agreement, or other license agreement which has been executed by you and Oracle and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Sample Code

Oracle may provide sample code in SuiteAnswers, the Help Center, User Guides, or elsewhere through help links. All such sample code is provided "as is" and “as available”, for use only with an authorized NetSuite Service account, and is made available as a SuiteCloud Technology subject to the SuiteCloud Terms of Service at www.netsuite.com/tos.

Oracle may modify or remove sample code at any time without notice.

No Excessive Use of the Service

As the Service is a multi-tenant service offering on shared databases, Customer may not use the Service in excess of limits or thresholds that Oracle considers commercially reasonable for the Service. If Oracle reasonably concludes that a Customer’s use is excessive and/or will cause immediate or ongoing performance issues for one or more of Oracle’s other customers, Oracle may slow down or throttle Customer’s excess use until such time that Customer’s use stays within reasonable limits. If Customer’s particular usage pattern requires a higher limit or threshold, then the Customer should procure a subscription to the Service that accommodates a higher limit and/or threshold that more effectively aligns with the Customer’s actual usage pattern.

Beta Features

Oracle may make available to Customer certain features that are labeled “beta” that are not yet generally available. To use such features, Customer acknowledges and agrees that such beta features are subject to the terms and conditions accepted by Customer upon activation of the feature, or in the absence of such terms, subject to the limitations for the feature described in the User Guide and as follows: The beta feature is a prototype or beta version only and is not error or bug free and Customer agrees that it will use the beta feature carefully and will not use it in any way which might result in any loss, corruption or unauthorized access of or to its or any third party’s property or information. Customer must promptly report to Oracle any defects, errors or other problems in beta features to support@netsuite.com or other designated contact for the specific beta feature. Oracle cannot guarantee the continued availability of such beta features and may substantially modify or cease providing such beta features without entitling Customer to any refund, credit, or other compensation. Oracle makes no representations or warranties regarding functionality or use of beta features and Oracle shall have no liability for any lost data, incomplete data, re-run time, inaccurate input, work delay, lost profits or adverse effect on the performance of the Service resulting from the use of beta features. Oracle’s standard service levels, warranties and related commitments regarding the Service shall not apply to beta features and they may not be fully supported by Oracle’s customer support. These limitations and exclusions shall apply until the date that Oracle at its sole option makes a beta feature generally available to its customers and partners as part of the Service without a “beta” label.

Table of Contents

SuiteScript Overview 1
- What You Can Do with the SuiteScript API 1
- Using the SuiteScript API with NetSuite Records 2

Setting Up Your SuiteScript Environment 5
- Environment Setup Overview 5
- Configuring NetSuite for SuiteScript 5
  - Enabling SuiteScript 5
  - Showing Record and Field IDs in Your Account 6
  - Setting Roles and Permissions for SuiteScript 10
- Working with IDEs Other Than SuiteCloud IDE 10

SuiteScript Governance 13
- SuiteScript 1.0 API Governance 13
- Script Usage Unit Limits 15
- Monitoring Script Usage 16
- Governance on Script Logging 17
- Search Result Limits 18

SuiteScript Best Practices 19
- General Development Guidelines 19
- Suitelets and UI Object Best Practices 20
- User Event Best Practices 21
- Scheduled Script Best Practices 21
- Client Script Best Practices 22
- Security Considerations 23
- Script Optimization 23

SuiteScript Debugger 25
- SuiteScript Debugger Overview 25
- Before Using the SuiteScript Debugger 25
- Using the SuiteScript Debugger 27
  - On Demand Debugging 27
  - Deployed Debugging 29
- SuiteScript Debugger Interface 33
  - SuiteScript Debugger Buttons 34
  - SuiteScript Debugger Tabs 34
- SuiteScript Debugger Metering and Permissions 38
- SuiteScript Debugger Keyboard Shortcuts 39
- SuiteScript Debugger Glossary 40

SuiteCloud Processors 41
- SuiteCloud Processors – Terminology 41
- SuiteCloud Processors – Basic Architecture 42
- SuiteCloud Processors – Supported Task Types 43
- SuiteCloud Processors – Processor Allotment Per Account 44
- SuiteCloud Processors – Priority Levels 45
  - SuiteCloud Processors – Priority Settings Page 46
  - SuiteCloud Processors – Priority Scheduling Examples 47
- SuiteCloud Processors – Priority Elevation and Processor Reservation (Advanced Settings) 55

SuiteScript Monitoring, Auditing, and Logging 59
- SuiteScript Montoring with the Application Performance Management (APM) 59
- SuiteScript Monitoring with the Script Queue Monitor 59
  - Script Queue Monitor Installation and Access 60
  - Script Queue Monitor User Interface 61

Working with the SuiteScript Records Browser 66
- How to Use SuiteScript Records Help 66
- SuiteScript References Overview 67

Using the SuiteScript Records Browser 67
Finding a Record or Subrecord 68
Understanding the Record Summary 69
Deleted Record Search 71
SuiteScript Supported Records 72
Activities 82
Activity 82
Event 83
Phone Call 83
Project Task 84
Resource Allocation 86
Task 88
Work Calendar 89
Entities 89
Competitor 90
Contact 91
Customer 92
Customer Status 94
Employee 94
Entity 96
Generic Resource 96
Job 97
Lead 98
Other Name 100
Partner 101
Project (Job) 102
Project Status 104
Project Template 105
Project Type 106
Prospect 106
Time-Off Management 108
Vendor 113
Items 114
Using Item Records in SuiteScript 115
Pricing Sublist / Pricing Matrix 116
Assembly Item 130
Description 131
Discount 132
Download Item 132
Gift Certificate Item 133
Inventory Item 134
Item Group 135
Item Search 135
Kit 136
Lot Numbered Assembly Item 136
Lot Numbered Inventory Item 137
Markup 138
Non-Inventory Part 138
Other Charge Item 139
Payment 140
Reallocate Items 140
Serialized Assembly Item 141
Serialized Inventory Item 142
Service 142
Shipping Item 143

Subscription Plan 144
Subtotal 145
Communications 145
Message 145
Note 147
Transactions 148
Advanced Intercompany Journal Entry 150
Assembly Build 151
Assembly Unbuild 152
Bin Putaway Worksheet 153
Bin Transfer 153
Blanket Purchase Order 154
Cash Refund 155
Cash Sale 156
Change Order 158
Charge 159
Check 160
Credit Card Charge 161
Credit Card Refund 162
Credit Memo 163
Customer Deposit 164
Customer Payment 165
Customer Refund 166
Deposit 167
Deposit Application 169
Estimate / Quote 170
Expense Report 172
Fulfillment Request 172
Intercompany Journal Entry 175
Intercompany Transfer Order 177
Inventory Adjustment 178
Inventory Cost Revaluation 179
Inventory Count 180
Inventory Detail 182
Inventory Status Change 182
Inventory Transfer 183
Invoice 184
Item Demand Plan 186
Demand Plan Detail Sublist 188
Item Fulfillment 193
Item Receipt 194
Item Supply Plan 195
Journal Entry 197
Landed Cost 198
Manufacturing Operation Task 200
Manufacturing Planned Time 202
Multi-Book Accounting Transaction 202
Opportunity 204
Order Schedule 205
Paycheck 205
Paycheck Journal 206
Payroll Batch 207
Payroll Batch Employee 209
Price Plan 210
Purchase Contract 210

Purchase Order 211
Requisition 212
Return Authorization 214
Revenue Arrangement 215
Revenue Commitment 217
Revenue Commitment Reversal 217
Sales Order 218
Statistical Journal Entry 220
Store Pickup Fulfillment 222
Subscription 223
Subscription Line 223
Time 224
Transaction Search 224
Transfer Order 225
Usage 226
Vendor Bill 226
Vendor Credit 227
Vendor Payment 228
Vendor Return Authorization 229
Work Order 229
Work Order Close 230
Work Order Completion 232
Work Order Issue 233
Support 234
Case 235
Issue 235
Solution 237
Topic 238
File Cabinet 239
File 239
Folder 239
Lists 240
Account 242
Accounting Book 243
Accounting Context 244
Accounting Period 245
Allocation Schedule 246
Amortization Schedule 247
Amortization Template 248
Billing Account 250
Billing Class 251
Billing Rate Card 252
Billing Schedule 253
Bin 256
BOM 256
BOM Revision 262
Class 265
Consolidated Exchange Rate 265
Contact Category 267
Contact Role 268
Cost Category 268
Currency 269
Customer Category 270
Customer Message 270
Customer-Subsidiary Relationship 271

Department 272
Expense Category 272
Fair Value Price 273
Gift Certificate 274
Global Account Mapping 275
Government-Issued ID Tracking 276
Group 279
Inbound Shipment 280
Intercompany Allocation Schedule 282
Inventory Number 283
Item Account Mapping 284
Item Location Configuration 286
Item Revision 287
Kudos Feature Records 288
Location 289
Manufacturing Cost Template 290
Manufacturing Routing 290
Nexus 291
Note Type 291
Other Name Category 292
Ownership Transfer 293
Partner Category 294
Payment Method 294
Payroll Item 295
Price Book 297
Price Level 298
Price Plan 299
Pricing Group 299
Project Charge Rule 300
Project Expense Type 301
Project Revenue Rule 302
Revenue Recognition Event 303
Revenue Recognition Plan 304
Revenue Recognition Schedule 305
Revenue Recognition Template 306
Role 306
Sales Role 307
Sales Tax Item 308
Subsidiary 308
System Note 309
Tax Control Account 311
Tax Group 314
Tax Period 316
Tax Type 316
Term 317
Unit of Measure 317
Vendor Category 318
Vendor-Subsidiary Relationship 319
Workplace 319
Customization 320
Custom List 320
Custom Record 321
Custom Segment Fields and Values 325
Custom Transaction 327
Scheduled Script Instance 329

Script 331
Script Deployment 334
Marketing 336
Campaign 337
Campaign Response 337
Campaign Template 338
Coupon Code 339
Email Template 340
Promotion 341
Website 343
CMS Content 344
CMS Content Type 345
CMS Page 346
Commerce Category 348
Website Setup 349
Shopping Cart 351
SuiteScript IDs 353
Permission Names and IDs 353
Feature Names and IDs 368
Preference Names and IDs 374
Task IDs 391
Button IDs 479

SuiteScript Overview

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

- What You Can Do with the SuiteScript API
- Using the SuiteScript API with NetSuite Records

What You Can Do with the SuiteScript API

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

SuiteScript is a JavaScript-based API that gives developers the ability to extend NetSuite beyond the capabilities provided through SuiteBuilder point-and-click customization.

The majority of NetSuite forms, records, customization objects and their event/trigger points are programmatically accessible through SuiteScript. What you decide to do with SuiteScript depends on which part of NetSuite you are trying to extend, search, or process.

When you think about using SuiteScript in NetSuite, you must ask yourself:

1. What Do I Want to Do?
2. Which APIs Support What I Want to Do?
3. How Do I Run a Script in NetSuite?

Important: NetSuite currently includes two versions of SuiteScript. For information on the most current version, see the help topic SuiteScript 2.0 API Introduction. For information on how SuiteScript versioning is handled by the system, see the help topic SuiteScript Versioning Guidelines.

What Do I Want to Do?

The following are some of the uses for SuiteScript. Next to each use case is a link to the NetSuite script type you might use to programmatically accomplish the tasks involved.

Using SuiteScript you can:

- Perform custom business processing when NetSuite records are updated, created, deleted (using User Event Scripts).
- Perform custom validations and calculations in the browser client (using Client Scripts).
- Create custom user interfaces (using script types such as Suitelets or User Event Scripts and UI Objects).
- Run batch processes (using Scheduled Scripts).
- Execute NetSuite searches (using script types such as User Event Scripts or Scheduled Scripts).

- Perform various utility processing such as sending email and faxes, creating and uploading files, or working with XML documents (using script types such as User Event Scripts or Suitelets).
- Create custom dashboard portlets (using Portlet Scripts).
- Perform processing in target accounts for bundled solutions as part of bundle installation or update (using Bundle Installation Scripts).

Which APIs Support What I Want to Do?

In the documentation, the SuiteScript API is organized by the types of tasks most developers want to perform. See the help topic SuiteScript API Overview to get started with the SuiteScript API.

See the help topic SuiteScript Functions to see how all APIs are organized. The documentation for each API lists whether the API can be used in client, user event, scheduled, Suitelet, or portlets scripts.

How Do I Run a Script in NetSuite?

The overall process for getting a script to run in NetSuite is fairly basic. This process includes:

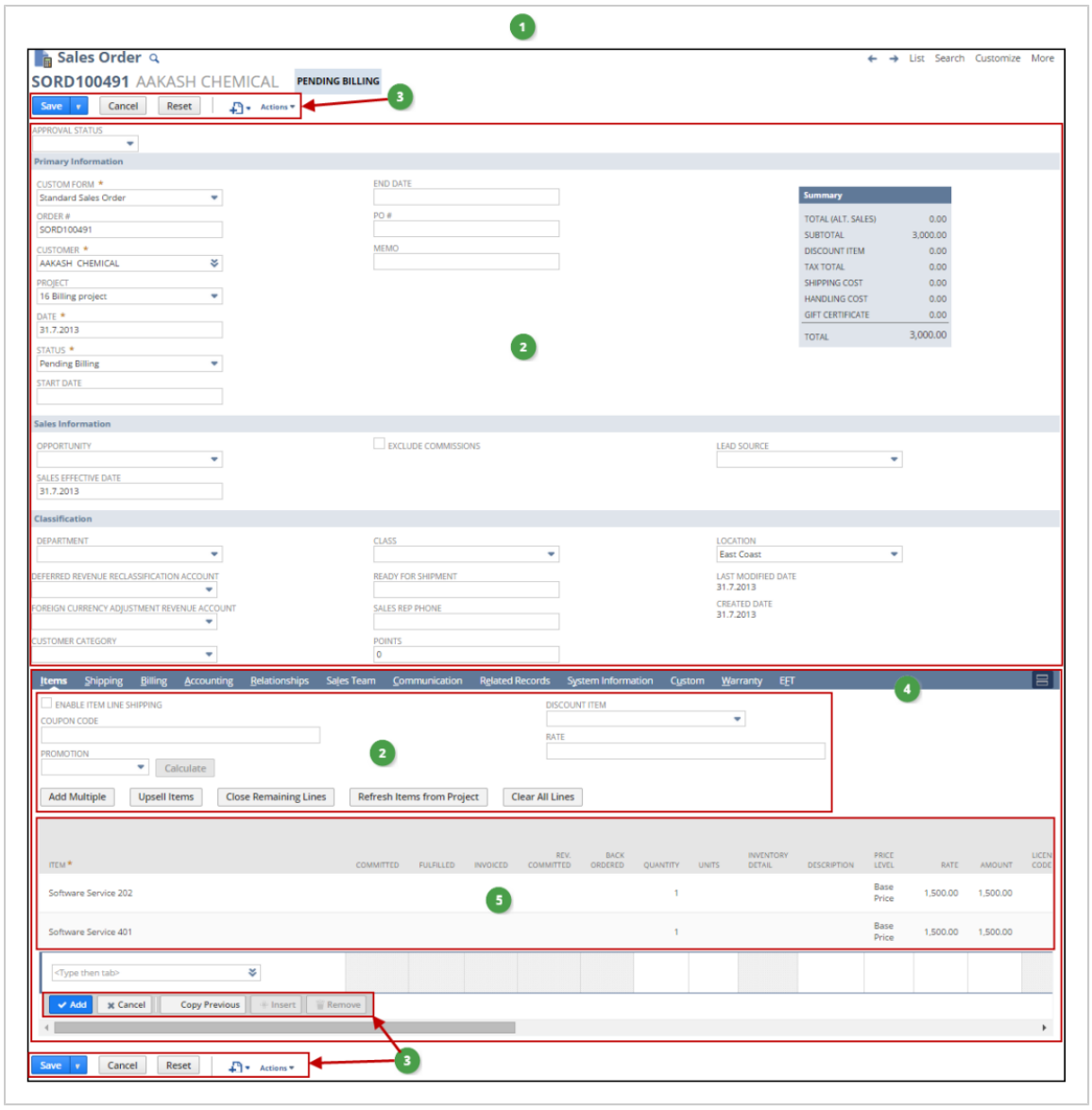
1. Creating a JavaScript file for your script.
2. Uploading the file into NetSuite.
3. Creating a NetSuite "Script" record for your script.
4. Defining script runtime options on a NetSuite Script Deployment page.

For complete details on each step in the process, start with the Running Scripts in NetSuite Overview topic in the NetSuite Help Center.

Using the SuiteScript API with NetSuite Records

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The figure below shows a standard Sales Order record in NetSuite.



The figure outlines the basic components of the record, such as:

1. Record object
2. Body fields
3. Buttons and Actions
4. Subtabs
5. Sublists

Many of the APIs in SuiteScript are used to get and set values on each of these components. You can also use SuiteScript to create these components.

Records

Use Record APIs to interact with the entire record object.

Fields

Use Field APIs to interact with the body fields on the main area of the record. Body fields can also appear on a subtab.

Buttons

The use of SuiteScript on built-in buttons is not currently supported. However, you can add a new button object to a page using the nlobjButton UI object.

Tabs

You can programmatically add fields to NetSuite tabs. You can also add custom tabs using the nlobjTab UI object.

Sublists

Use Sublist APIs to interact with “line item” sublist fields.

Setting Up Your SuiteScript Environment

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

- Environment Setup Overview
- Configuring NetSuite for SuiteScript
- Working with IDEs Other Than SuiteCloud IDE

Environment Setup Overview

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Before working with SuiteScript, you should configure both your NetSuite account and your SuiteScript development environment accordingly. See the following sections:

- Configuring NetSuite for SuiteScript
- Setting Up SuiteCloud IDE

Configuring NetSuite for SuiteScript

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

You must complete all of the following tasks to enable SuiteScript in your account and to access the internal record and fields IDs that may be required as parameter values in your SuiteScript code. These tasks include:

1. Enabling SuiteScript
2. Showing Record and Field IDs in Your Account
3. Setting Roles and Permissions for SuiteScript

Important: After completing these tasks, see the help topic Setting Up SuiteCloud IDE.

Enabling SuiteScript

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

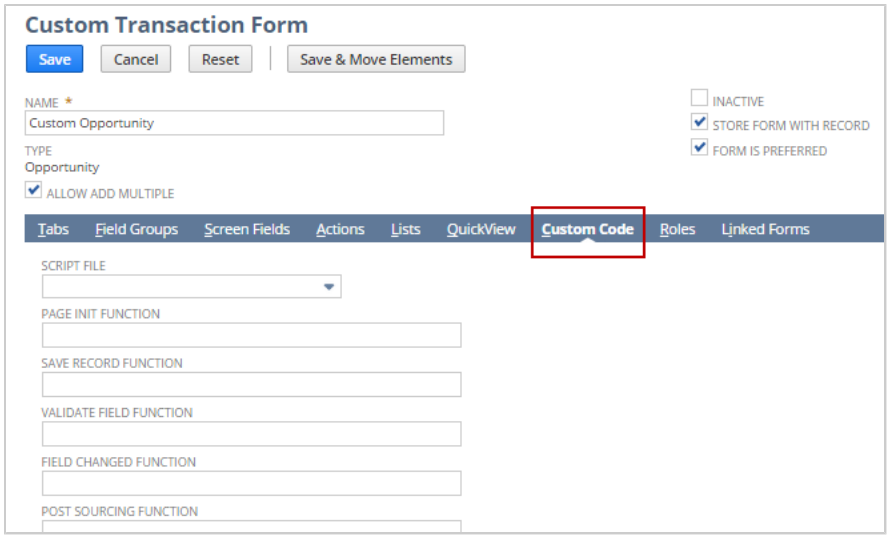
Before you can run SuiteScript in your NetSuite account, you must enable the SuiteScript feature.

To enable SuiteScript:

1. Go to Setup > Company > Enable Features.

2. Click the **SuiteCloud** tab.
3. Under SuiteScript, click the **Client SuiteScript** or **Server SuiteScript** box (or both, depending on the scripts you want to run).
4. Click **Save**.

Important: If Client SuiteScript is enabled, the **Custom Code** tab becomes available on entry and transactions forms (see figure). Here you define which client scripts to associate with the **current form**. For information on attaching client scripts to NetSuite forms, see the help topic Running Scripts in NetSuite Overview. For information on entry and transactions forms, see the help topic Custom Forms in the SuiteBuilder (Customization) Guide.



Showing Record and Field IDs in Your Account

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

After enabling the SuiteScript feature, NetSuite recommends that you enable the **Show Internal IDs** preference. Enabling this preference lets you see the internal IDs for all fields and records in NetSuite.

When referencing a specific record or field in SuiteScript, you use the internal ID. Even if the record or field's UI label is changed, the internal ID will remain constant.

After enabling **Show Internal IDs**, see these sections for steps on viewing the IDs from within NetSuite:

- How do I find a record's internal ID?
- How do I find a field's internal ID?

Important: You obtain the internal ID of a record type (for example, 'salesorder') by going to the SuiteScript Records Browser. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

To show internal NetSuite IDs:

1. Go to Home > Set Preferences.

2. Click the **General** tab and then click the **Show Internal IDs** check box (see figure).
3. Click **Save**.

> **Note:** For examples of how internal IDs are referenced in the SuiteScript API, see the help topic nlapiLoadRecord(type, id, initializeValues) or nlapiSearchRecord(type, id, filters, columns). Also note that when writing SuiteScript, **all record and field IDs must be in lowercase**.

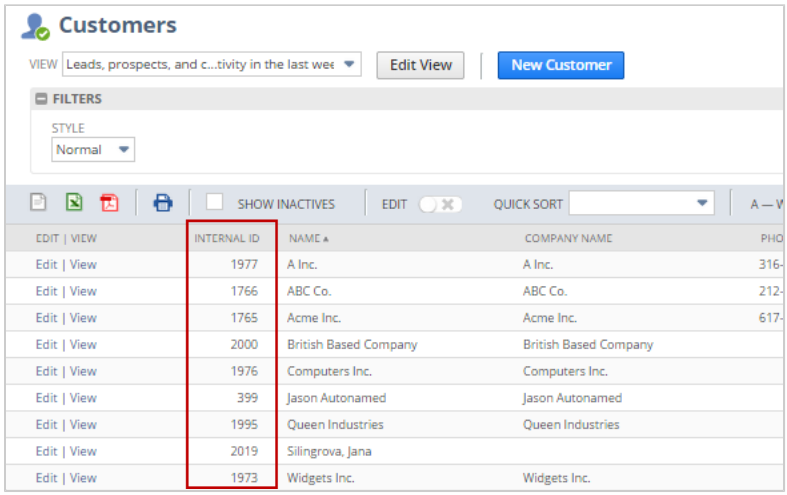
How do I find a record's internal ID?

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

A record's internal ID is unique and associated with the record at the time it is created. After the **Show Internal IDs** preference is enabled, the internal IDs for each record are displayed in the Internal ID column of record lists (see figure).

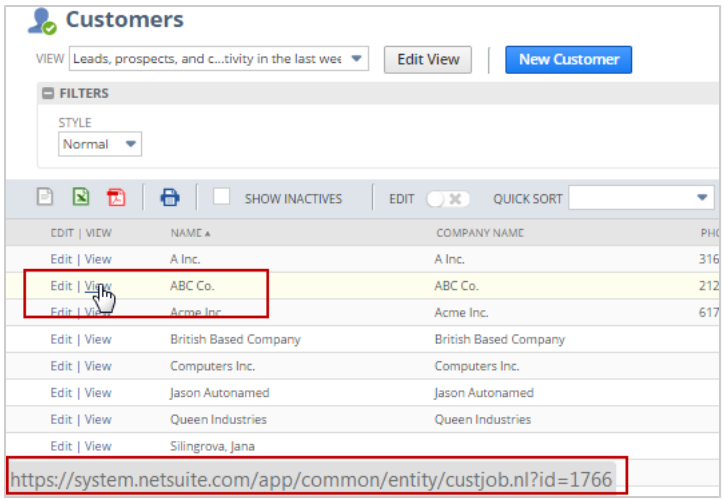
For example, to see an internal ID for a specific customer record, go to Lists > Relationships > Customer. In the Internal ID column, the internal ID appears next to each record in the Customers list (see figure).

> **Note:** See Showing Record and Field IDs in Your Account for steps on enabling the **Show Internal IDs** preference.



If the **Show Internal IDs** preference is NOT enabled, or if the internal IDs are not displayed on a particular page within NetSuite, you can see the internal ID for a record by hovering over a link to that record. The internal ID is displayed as a parameter in the URL in the browser status bar.

The following figure shows that if you hover over a link to the ABC Co. customer record, the internal record ID (**1766**) appears in the browser status bar.



Important: You obtain the internal ID of a record type (for example, 'salesorder') by going to the SuiteScript Records Browser. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

How do I find a field's internal ID?

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Internal field IDs must be used when calling a field from a SuiteScript API.

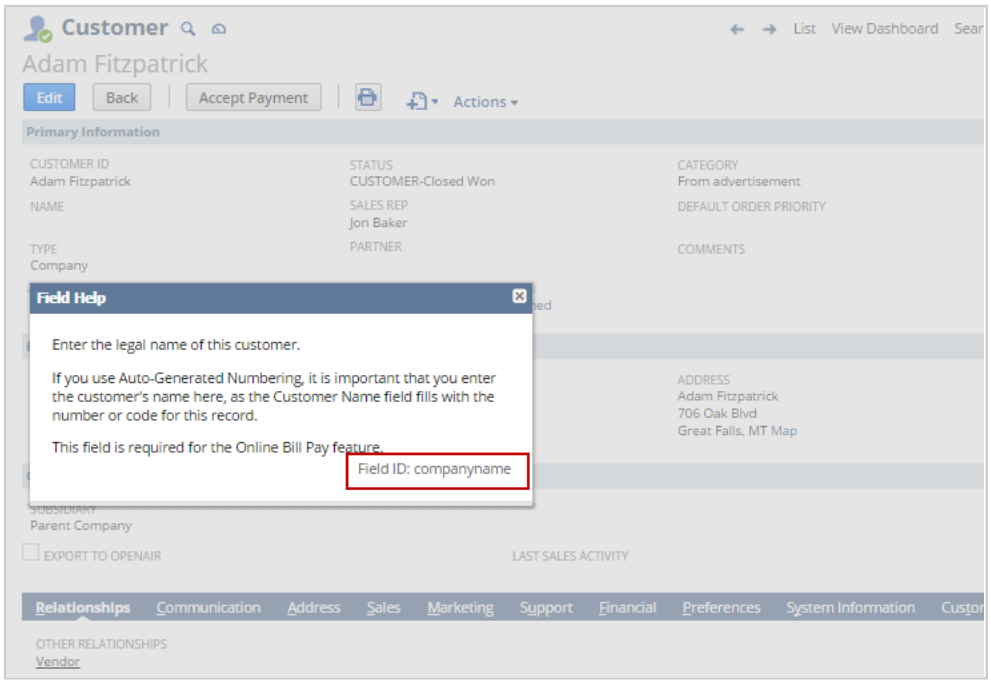
There are multiple ways to view a field's internal ID:

- Using the SuiteScript Records Browser
- Internal IDs for Standard and Custom Fields in Field Level Help Window
- Internal IDs for Custom Fields on List Pages

Some fields are not displayed to users in NetSuite. Use the SuiteScript Records Browser to find internal IDs for those fields.

Internal IDs for Standard and Custom Fields in Field Level Help Window

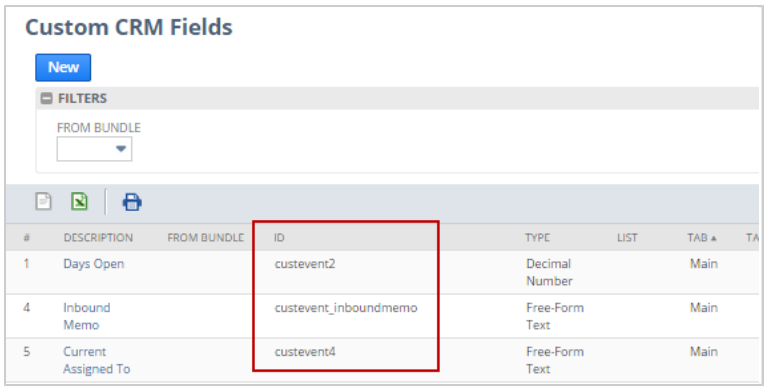
With **Show Internal IDs** enabled, you can also view internal field IDs for both standard and custom fields, by clicking the field label in the UI. The figure below shows the Field Level Help window that opens when a field label is clicked, in this case, the Company label. The internal ID for the Company field is `companyname`, which appears in the bottom-right corner of the Field Level Help window.



Note: When creating custom fields, you can specify your own field ID, or you can accept the default ID assigned by NetSuite. To ensure that the field IDs make sense in the context of your business environment, it is recommended that you define your own custom field IDs. For detailed information on creating custom fields and assigning custom field IDs, refer to Custom Fields in the NetSuite Help Center.

Internal IDs for Custom Fields on List Pages

When the **Show Internal IDs** preference is enabled, internal IDs for each custom field are displayed in the Internal ID column of a custom field page. For example, to see the internal IDs for custom CRM fields, go to Customization > Lists, Records, & Fields > CRM Fields. The ID column appears in the list of custom fields, as shown in the figure.



Setting Roles and Permissions for SuiteScript

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

NetSuite provides many standard roles with predefined permissions. A role is a set of permissions that lets customers, vendors, partners, and employees access specific areas of your data. Each role grants access at a certain level for each permission. For additional information on roles and permissions, see the help topic NetSuite Users & Roles.

Access to the SuiteScript feature is also controlled using roles and permissions. Use the Administrator role for complete access to all SuiteScript functionality, or use the following permissions for more granular access.

Note: If you customize a role to add SuiteScript functionality, you must also include the permissions to customize entry forms and transaction forms.

Action	Permission Required
View script records and script deployment records.	Role with SuiteScript permission (View level)
Create and view script records and script deployment records.	Role with SuiteScript permission (Create level)
Create, view, and edit script records and script deployment records.	Role with SuiteScript permission (Edit level)
Use **Save and Execute** from the script deployment record UI to execute an on demand scheduled script or on demand map/reduce script.	Role with SuiteScript permission (Full level)
Use an API (for example, ScheduledScriptTask.submit()) to execute an on demand scheduled script or on demand map/reduce script.	Role with SuiteScript Scheduling permission (Full level)

Execute a script with one of the following APIs:
- SuiteScript 1.0
  - nlobjContext.getPercentComplete()
  - nlobjContext.setPercentComplete(pct)
- SuiteScript 2.0
  - Script.percentComplete
	Role with SuiteScript Scheduling permission (Full level)

Working with IDEs Other Than SuiteCloud IDE

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Although NetSuite recommends that you use SuiteCloud IDE when writing SuiteScript, you can still use other development tools to create SuiteScript. Note, however, without SuiteCloud IDE, you will not be able to automatically upload SuiteScript files into the NetSuite file cabinet.

Important: For more information about the recommended development environment, see the help topic Setting Up SuiteCloud IDE.

If you choose to use a development tool or IDE other than SuiteCloud IDE, see the following sections:

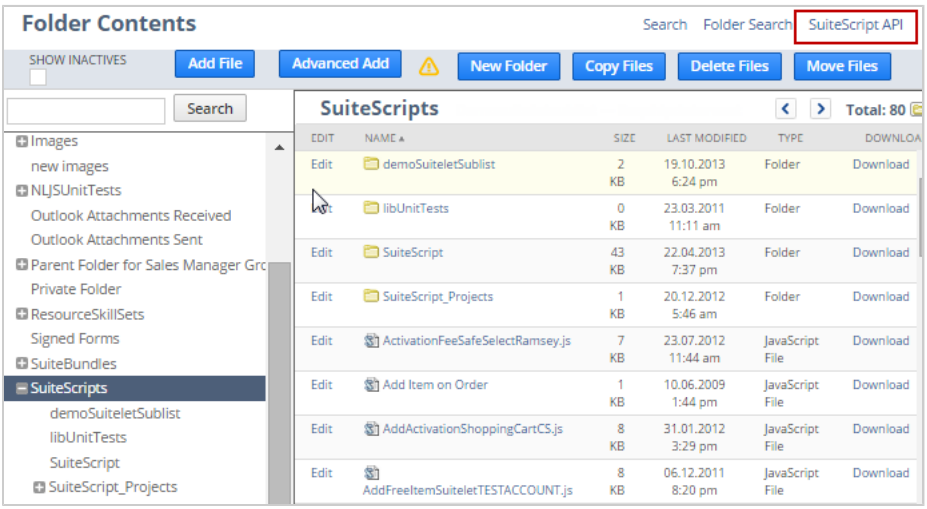
- Adding the SuiteScript Library File to Your IDE
- Uploading SuiteScript into the File Cabinet Without the SuiteCloud IDE

Adding the SuiteScript Library File to Your IDE

If you are working with an IDE other than SuiteCloud IDE, you should still add the SuiteScript library file to your SuiteScript project folder in your IDE.

To add the SuiteScript library file:

1. In NetSuite, go to Documents > Files > SuiteScripts.
2. Next, click the link to the **SuiteScript API** file (see figure).



3. Copy and paste the SuiteScript API file into your IDE.
4. Save the file as a .js file.

Uploading SuiteScript into the File Cabinet Without the SuiteCloud IDE

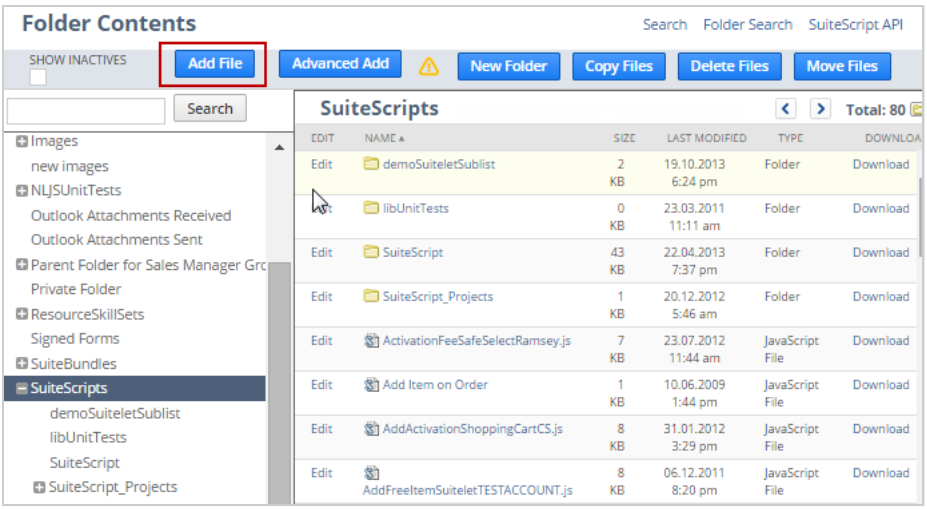
After the SuiteScript feature is enabled, a new SuiteScripts folder is created in the NetSuite file cabinet. The file cabinet is considered as the central repository for all your .js SuiteScript files. Therefore, your SuiteScript files should exist in the file cabinet before they can be executed in your NetSuite account.

- The **SuiteScripts** folder within the file cabinet is provided for convenience, however, you can store the script files in any location.
- For steps on enabling the SuiteScript feature, see Enabling SuiteScript.

If you are not uploading your scripts into the file cabinet using the SuiteCloud IDE, use the following steps.

To upload SuiteScript into the file cabinet:

1. Go to Documents > Files > SuiteScripts.
2. Click the **Add File** button (see figure).



3. In the File Upload window that appears, navigate to the file you want to upload, select the file, and click **Open**.

> **Note:** If you make changes to a SuiteScript file that already exists in the file cabinet, follow steps 1–3 to re-upload the changed file. Click **OK** to overwrite the previous file and load your changes.

SuiteScript Governance

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Important: SuiteScript thresholds are based on the volume of activity that a company's users can manually generate. However, automated functions that generate excessive levels of activity may trigger metering of script execution as referenced in the NetSuite Main Terms of Service (TOS).

To optimize application performance, NetSuite has implemented a SuiteScript governance model based on usage units. If the number of allowable units is exceeded, the script is terminated.

Usage units are tracked on two levels: the **API level** and the **script type level**. Each SuiteScript API consumes a system-defined number of processing units, and each script type can execute a system-defined number of units.

See the following topics to learn about unit governance as it applies to individual **APIs** and specific **script types**.

- SuiteScript 1.0 API Governance (SuiteScript 1.0)
- Script Usage Unit Limits (all versions of SuiteScript)
- Monitoring Script Usage (all versions of SuiteScript)

See the following topics for information on logging limits and search result limits:

- Governance on Script Logging (all versions of SuiteScript)
- Search Result Limits (all versions of SuiteScript)

Important: The governance limits for each SuiteScript 2.0 API are listed in the individual topics describing the SuiteScript 2.0 methods.

Note: NetSuite has put internal mechanisms in place to detect “runaway scripts” that include infinite loops. When they are caught, these scripts will be terminated and an SSS_INSTRUCTION_COUNT_EXCEEDED error message is thrown. Should you receive this error, NetSuite recommends that you examine the *for* loops in your script to ensure that they contain either a terminating condition or a condition that can be met.

SuiteScript 1.0 API Governance

The following table lists each API and the units each consumes.

Important: The content in this help topic pertains to SuiteScript 1.0. The governance limits for each SuiteScript 2.0 API are listed in the individual topics describing SuiteScript 2.0 methods.

Notice the APIs marked with an asterisk consume a different number of units based on the type of record they are running on. This kind of governance model takes into account the NetSuite processing

requirements for three categories of records: custom records, standard transaction records, and standard non-transaction records.

Custom records, for example, require less processing than standard records. Therefore, the unit cost for custom records is lower to be commensurate with the processing required for standard records. Similarly, standard non-transaction records require less processing than standard transaction records. Therefore, the unit cost for standard non-transaction records is lower to be commensurate with the processing required for standard transaction records.

Note: Standard **transaction** records include records such as Cash Refund, Customer Deposit, and Item Fulfillment. Standard **non-transaction** records include records such as Activity, Inventory Item, and Customer. In the section on SuiteScript Supported Records in the NetSuite Help Center, see the "Record Category" column. All record types not categorized as **Transaction** are considered to be standard non-transaction records. Custom List and Custom Record are considered to be custom records.

API	Unit Usage per API	Example
nlapiDeleteFile nlapiInitiateWorkflow nlapiTriggerWorkflow nlapiScheduleScript nlapiSubmitConfiguration nlapiSubmitFile nlobjEmailMerger.merge()	**20**	A user event script on a standard transaction record type (such as Invoice) that includes one call to nlapiDeleteRecord and one call to nlapiSubmitRecord - - consumes 40 units (assuming no other nlapi calls were made). In this case, the user event script consumes 40 units out of a possible 1,000 units available to user event scripts. (See Script Usage Unit Limits for the total units allowed for a user event script.)
nlapiDeleteRecord nlapiSubmitRecord	When used on standard transactions: **20** When used on standard non-transactions: **10** When used on custom records: **4**	
nlapiAttachRecord nlapiDetachRecord nlapiExchangeRate nlapiGetLogin nlapiLoadConfiguration nlapiLoadFile nlapiMergeRecord nlapiRequestURL nlapiRequestURLWithCredentials nlapiSearchGlobal nlapiSearchRecord nlapiSendCampaignEmail nlapiSendEmail nlapiVoidTransaction nlapiXMLToPDF nlobjSearchResultSet.getResults nlobjSearchResultSet.forEachResult	**10**	A scheduled script on a standard non-transaction record type (such as Customer) that includes one call to nlapiLoadRecord, one call to nlapiTransformRecord, one call to nlapiMergeRecord, and one call to nlapiSendEmail consumes 30 units. In this case, the scheduled script consumes 30 units out of a possible 10,000 units available to scheduled scripts. (See Script Usage Unit Limits for the total units allowed for a scheduled script.)

API	Unit Usage per API	Example
nlapiCreateRecord nlapiCopyRecord nlapiLookupField nlapiLoadRecord nlapiSubmitField nlapiTransformRecord	When used on standard transactions: **10** When used on standard non-transactions: **5** When used on custom records: **2**	**Note:** To conserve units, only use nlapiSubmitField on fields that are available through inline edit. For more information, see the help topics Updating a field that is available through inline edit and Updating a field that is not available through inline edit.
nlapiLoadSearch nlobjJobManager.getFuture nlobjSearch.saveSearch	**5**	
nlapiLogExecution	See Governance on Script Logging.	
nlapiSetRecoveryPoint nlapiSubmitCSVImport nlobjJobManager.submit	**100**	
All other SuiteScript APIs	0	

Script Usage Unit Limits

Note: The content in this help topic pertains to all versions of SuiteScript.

The following table lists the maximum units allowed for a particular script type. In SuiteScript 1.0, you can use nlobjGetContext.getRemainingUsage() to see how many units you have remaining for a particular scheduled, user event, portlet, client, or Suitelet script. In SuiteScript 2.0, you can use the Script.getRemainingUsage() method.

Script Type	Total Units Allowed per Script	Notes
Scheduled Scripts	10,000	A scheduled script that includes two calls to nlapiTransformRecord, one call to nlapiMergeRecord, and one call to nlapiSendEmail consumes 40 units out of a possible 10,000 available. **Note:** Scheduled scripts with potentially long execution times should include an nlapiYieldScript() call to avoid exceeding their allowed governance.
User Event Scripts	1,000	Regardless of the 1,000 unit limit for user event scripts, developers should design their scripts so that they are responsive to users, otherwise user experience may be impacted.
Client Scripts	1,000	Be aware that client scripts are metered on a **per-script** basis. If an account has one **form-level** client script attached to a form, and one **record-level** client script deployed to the record (which contains the form), **each** client script can total 1000 units. Usage units are not shared by all the client scripts associated with a form or record.

Script Type	Total Units Allowed per Script	Notes
		Note: For information on form- and record-level client scripts, see the help topic Form-level and Record-level Client Scripts.
SuiteScript 2.0 Map/Reduce Script Type		Map/reduce scripts are available in SuiteScript 2.0 only. Note that the system does not impose any limit on the full duration of a map/reduce script deployment instance. Rather, the system regulates isolated components of the deployment, such as the units used by a single function invocation. For more information, see the help topic SuiteScript 2.0 Map/Reduce Governance.
Suitelets	1,000	For example, a Suitelet that calls nlapiCreateRecord and nlapiRequestURL consumes 20 units out of a possible 1,000 units available. **Note:** Regardless of the 1,000 unit limit for Suitelets, developers should design their Suitelets to be responsive to users, otherwise user experience may be impacted.
RESTlets	5,000	
Portlet Scripts	1,000	
Mass Update Scripts		You can have 1000 units per record/invocation of the script.
Bundle Installation Scripts	10,000	Bundle installation scripts are governed by a maximum of 10,000 units per execution.
Workflow Action Scripts (also referred to as Custom Action in SuiteFlow)	1,000	Use workflow action scripts to create custom actions in your workflow. Note that within one workflow state, all actions combined cannot not exceed 1000 units. Therefore, if you have developed a custom action (using a workflow action script) that consumes 990 units, be aware of the unit consumption of the other actions within that state. See the help topic Workflow Action Usage Units for a list of all workflow actions and the units they consume when executed within a state.
Core Plug-ins	10,000	This is the default limit for core plug-ins that do not have more restrictive limits defined. See the help for each core plug-in for any specific usage unit limits.

Note: NetSuite also enforces a usage limit of 1000 units for SSP application scripts. For more information, see the help topic SSP Application Governance.

Monitoring Script Usage

In SuiteScript 1.0, you can monitor SuiteScript unit usage through the nlobjContext.getRemainingUsage() method. In SuiteScript 2.0, you can use the Script.getRemainingUsage() method.

Note: To access the getRemainingUsage() method, call nlapiGetContext() to instantiate the nlobjContext object.

You can also monitor script units by running the script in the SuiteScript Debugger. After a script completes execution, unit governance details appear on the Execution Log tab in the SuiteScript Debugger console. Note that emailed error messages also include a line item for **Script Usage**. Script Usage shows the number of units that were executed in the script before the error was hit.

Example 1

This sample shows how to instantiate the nlobjContext object, and then call getRemainingUsage() so that the script's remaining usage units appear in the execution log.

```
var context = nlapiGetContext();
nlapiLogExecution('DEBUG', 'remaining usage', context.getRemainingUsage());
```

Example 2

This sample shows how to instantiate the nlobjContext object, and then check for remaining units. If the units remaining are greater than 50, the script will execute a certain set of instructions.

```
var context = nlapiGetContext();
   if (parseInt(context.getRemainingUsage()) > 50)
   {
   //execute code here
   }
```

Note: For a SuiteScript 2.0 sample, see the help topic Script.getRemainingUsage().

Governance on Script Logging

Note: The content in this help topic pertains to all versions of SuiteScript.

NetSuite governs the use of nlapiLogExecution(type, title, details). The governance model is meant to safeguard against unreasonably excessive logging, which can negatively affect performance for other NetSuite customers sharing the same database. The governance model is not meant to impact companies (or scripts) that are using `nlapiLogExecution()` appropriately.

The governance model is as follows:

Within a **60 minute** time period, a company is allowed to make up to 100,000 calls to `nlapiLogExecution()` across **all** of their scripts.

If within a **60 minute** time period NetSuite detects a script is excessively logging (and pushing a company close to the 100,000 `nlapiLogExecution()` call limit), NetSuite will change the offending script's log level to the next level higher. The offending script will continue its execution, however, its log level will go from Debug to Audit, or Audit to Error, or Error to Emergency, depending on the script.

Note: For information on script log levels, see the help topic Setting Script Execution Log Levels.

Changing log levels ensures that the offending script continues to execute and helps prevent an inordinate amount of logging from only one company.

The capacity for script execution logs is shared by customers on the same database. For further protection against excessive logging, script execution logs are governed by a total storage limit on each instance of the NetSuite database. On each NetSuite server, if the database table that stores logs reaches this limit, all logs (across all customers on that server) are purged. For this reason, NetSuite recommends that you store information using custom records.

In SuiteScript 2.0, use the N/log Module to access methods for logging script execution details. The governance model for the N/log module is as follows:

Within a **60 minute** time period, a company is allowed to make up to 100,000 log object method calls across **all** of their scripts.

System errors older than **60 days** and user-generated logs older than **30 days** are purged.

For more information, see the help topic N/log Module Guidelines.

Example

Company ABC has 10 scripts running during a **60 minute** period. If one out of the 10 scripts calls `nlapiLogExecution('DEBUG', 'My log', x.getID())` 70,000 times within only a 20 minute time period, NetSuite will raise the script's log level.

The change to the log level will appear in the **Log Level** field on the script's Script Deployment page (see figure). In the figure below, if the offending script's Log Level was originally set to Debug, NetSuite will increase the log level to Audit. This means the line of code that reads `nlapiLogExecution('DEBUG', 'My log', x.getID())` will continue to execute, however nothing will be logged, as the log level for the script has been raised to Audit.

Script Owners Are Notified

If NetSuite detects that one script is logging excessively, the owner of the script is notified. The script owner is alerted that the script is the primary contributor to his or her company possibly exceeding the 100,000 logging threshold (for a **60 minute** time period).

NetSuite sends notifications through email and adds a log entry to the script's Execution Log. Both the email and the NetSuite-generated log alerts script owners that a script's Log Level has been increased.

Search Result Limits

Search results are limited to 1000 records when you execute SuiteScript searches using nlapiSearchRecord(...) in SuiteScript 1.0 or the N/search module in SuiteScript 2.0. For information on working with SuiteScript searches in NetSuite, see the help topics Searching Overview and Search APIs. For information about working with SuiteScript 2.0 searches, see the help topic N/search Module.

Note that if you load an existing saved search using nlapiLoadSearch(...), and then call nlobjSearch.runSearch() to return a result set of nlobjSearchResultSet objects, you can get up to 4000 results returned. See nlobjSearchResultSet.forEachResult(callback) for details.

SuiteScript Best Practices

- General Development Guidelines
- Suitelets and UI Object Best Practices
- User Event Best Practices
- Scheduled Script Best Practices
- Client Script Best Practices
- Security Considerations
- Script Optimization
- Search Issues and Best Practices for Web Services and SuiteScript

General Development Guidelines

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

This section provides general guidelines for working with SuiteScript.

- Always thoroughly test your code before using it on your live NetSuite data.
- Type all record, field, sublist, tab, and subtab IDs in lowercase in your SuiteScript code.
- Prefix all custom script IDs and deployment IDs with an underscore (_).
- Do not hard-code any passwords in scripts. The password and password2 fields are supported for scripting.

  For additional information, see Security Considerations.
- If the same code is used across multiple forms, ensure that you test any changes in the code for **each** form that the code is associated with.
- Include proper error handling sequences in your script wherever data may be inconsistent, not available, or invalid for certain functions. For example, if your script requires a field value to validate another, ensure that the field value is available.
- Organize your code into reusable chunks. Many functions can be used in a variety of forms. Any reusable functions should be stored in a common library file and then called into specific event functions for the required forms as needed.
- Place all custom code and markup, including third party libraries, in your own namespace.

  Important: Custom code must not be used to access the NetSuite DOM. Developers must use SuiteScript APIs to access NetSuite UI components.
- Use the built in Library functions whenever possible for reading/writing Date/Currency fields and for querying XML documents
- During script development, componentize your scripts, load them individually and then test each one -- inactivating all but the one you are testing when multiple components are tied to a single user event.
- Use **static** ID values in your API calls where applicable because name values can be changed.
- Use custom name spaces or unique prefixes for all your function names.

  When working with the top-level NetSuite functions in Client SuiteScript (for example the Page Init function) it is recommended that the new function name corresponds with the NetSuite name.

For example, a Page Init function can be named pageInit or formAPageInit. If your code is already established, you can wrap it with a top-level function that has the appropriate naming convention.

- Since name values can be changed, ensure that you use **static** ID values in your API calls where applicable.
- Although you can use any desired naming conventions for functions within your code, it is recommended that you use custom name spaces or unique prefixes for all your function names.

  When working with the top-level NetSuite functions in Client SuiteScript (for example the Page Init function) it is recommended that the new function name corresponds with the NetSuite name. For example, a Page Init function can be named pageInit or formAPageInit. If your code is already established, you can wrap it with a top-level function that has the appropriate naming convention.
- You must use the nlobjContext. **getSetting** method on nlapiGetContext to reference script parameters. For example, to obtain the value of a script parameter called *custscript_case_field*, you must use the following code:

```
nlapiGetContext().getSetting('SCRIPT','custscript_case_field')
```

  For additional information, see the help topics nlapiGetContext() and nlobjContext.
- Thoroughly comment your code. This practice helps with debugging and development but and assists NetSuite support in locating problems if necessary.

Suitelets and UI Object Best Practices

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

This section provides best practices for Suitelet development using UI objects and custom UI.

- Suitelets are ideal for generating NetSuite pages (forms, lists), returning data (XML, text), and redirecting requests.
- Limit the number of UI objects on a page (< 100 rows for sublists, < 100 options for on demand select fields, < 200 rows for lists).
- Experiment with inline HTML fields embedded on nlobjForm before going the full custom HTML page route.
- Deploy Suitelets as “Available without Login” only if absolutely necessary (no user context, login performance overhead). (See the help topic Setting Available Without Login.)
- Append “ifrmcntnr=T” to the external URL when embedding in iFrame especially if you are using Firefox. (For more about NetSuite and iFrame, see the help topic Embedding an Online Form in your Web Site Page.)
- When building custom UI outside of the standard NetSuite UI (such as building a custom mobile page using Suitelet), use the User Credentials APIs to help users manage their credentials within the custom UI. For more information, see the help topic User Credentials APIs.
- When the Advanced Employee Permissions feature is enabled keep the following in mind:
  - To avoid inadvertently exposing employee data, use caution when running Suitelets or Restlets as an administrator. A user with a role that has limited access to the employee record can access a Suitelet or Restlet that runs as an administrator. Depending on how the Suitelet or Restlet is written, the user may have access to employee information that they would otherwise not see.
  - Use caution when setting up Suitelets and Restlets to give access to users without having to log in since it could potentially expose employee information in uncontrolled ways.

User Event Best Practices

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

This section provides best practices for developing user event scripts.

- Use the *type* argument and context object to define and limit the scope of your user event logic.
- Limit the amount of script execution in user event scripts (< 5 seconds) since they run often and in-line. You can use the Script Performance Monitor SuiteApp to test the performance of your scripts deployed on a specific record type. See the help topic Application Performance Management (APM)
- Mission critical business logic implemented using user events should be accompanied by a 'Clean up' scheduled script to account for any unexpected errors or mis-fires.
- Any operation that depends on the submitted record being committed to the database should happen in an afterSubmit script.
- Updating the nlobjRecord returned by nlapiGetNewRecord() in a beforeSubmit will affect the values that are written to the database. Updating this object in an afterSubmit script has no effect. This object CANNOT be submitted (because it is already being submitted).
- The nlobjRecord returned by nlapiGetOldRecord() is READ-ONLY.
- Be careful when updating transaction line items in a beforeSubmit script because you have to ensure that the line item totals net taxes and discounts are equal to the *summarytotal, discounttotal*, *shippingtotal*, and *taxtotal* amounts.
- Activities (user events) on a hosted website can trigger server-side SuiteScripts. In addition to Sales Orders, scripts on Case and Customer records will also execute as a consequence of Web activities.

Scheduled Script Best Practices

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

This section provides best practices for working with scheduled scripts. Also see the help topic Best Practice: Handling Server Restarts in Scheduled Scripts (SuiteScript 1.0).

- It is recommended that scheduled scripts are set to run during the hours of 2 AM to 6 AM PST. Scripts set to run during the hours of 6 AM to 6 PM PST may not run as quickly due to high database activity.
- The number of **Not Scheduled** deployments to create should depend on the anticipated number of simultaneous calls you expect to make to this script and the approximate execution time of the script. A general rule of thumb is to create twice as many deployments as the total number of simultaneous calls you anticipate for this script.
- **Scheduled** deployments and **Not Scheduled** deployments are executed from the same processor pool. Keep this in mind as you deploy multiple scheduled scripts because the amount pending script instances is the ultimate bottle-neck for scheduled script execution throughput.
- Although there is no restriction on the number of **Not Scheduled** scripts that can be put into the processing pool, too many pending scripts may create a backlog and compromise system performance. Because only one script can be run in a processor at a time, it is not recommended that you overload the system.
- If you want to deploy scheduled scripts that are **scheduled to run hourly on a 24 hour basis**, the following sample values should be set on the Script Deployment page:

- Deployed = checked
- Daily Event = [radio button enabled]
- Repeat every 1 day
- Start Date = [today's date]
- **Start Time = 12:00 am**
- Repeat = every hour
- End By = [blank]
- No End Date = checked
- Status = Scheduled
- Log Level = Error
- Execute as Role = Set to **Administrator**

If the **Start Time** is set to any other time than 12:00 am (for example it is set to 2:00 pm), the script will start at 2:00 pm, but then finish its hourly execution at 12:00 am. It will not resume until the next day at 2:00 pm.

Client Script Best Practices

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

This section provides best practices for both form-level and record-level client SuiteScript development.

- When testing form-level client scripts, use Ctrl-Refresh to clear your cache and ensure that the latest scripts are being executed.
- Global (record-level) client scripts offer a more flexible deployment model and are easier to port (bundle) than client scripts attached to forms.
- The execution of nlapiSetFieldValue and nlapiSetCurrentLineItemValue is multi-threaded whenever child field values need to be sourced in. Use the postSourcing function to synchronize your logic. Setting the *synchronous* parameter to *true* in both nlapiSetFieldValue and nlapiSetCurrentLineItem will accomplish the same thing.
- Use nlapiSetFieldValue and nlapiSetCurrentLineItemValue instead of nlapiSetFieldText and nlapiSetCurrentLineItemText if the field you are setting sometimes renders as a popup text field. Your script will execute more predictably using the xxx **Value** functions because you are setting an internal ID, which is ultimately a more precise definition. The xxx **Text** functions perform searches that may return duplicates or no results, which effectively disables your script.
- When the Advanced Employee Permissions feature is enabled, the fields and sublists a user has access to can change depending on which employee is being viewed or edited. This is different from other records in NetSuite, where permissions granted to a role determines what the role can see for every instance of that record. In general, scripts should always check to see if a field exists before trying to do something with it. Simply calling functions and methods that interact with fields before checking whether the field is there may result in inconsistent behavior. For example, the department field is permitted on the employee record. When you check to see if this field exists and you do not have access, a null value is returned, and if the field is empty, an empty string is returned.
  - Unsafe Practice Example

```
var employeeRecord = nlapiLoadRecord('employee', '115');
```

```
employeeRecord.setFieldValue('department', 'marketing');
nlapiSubmitRecord(employeeRecord);
```

- Safe Practice Examlpe

To detect if your role has access to a field for a specific employee, load the employee record object and call `getAllFields()`. If the field exists and you do have access a `true` value is returned. In the below example, the user has access to the department field for the employee with ID:115.

```
var accessToDepartment = nlapiLoadRecord('employee', '115').getAllFields().includes('department
');
```

- When the Advanced Employee Permissions feature is enabled, the search columns available to users is also dependent on the permissions assigned to the role.

Security Considerations

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

To prevent users from accessing sensitive information such as password and credit card data, the following fields cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).

Field Internal ID	Field UI Label
password	Password
password2	
ccnumber	Credit Card Number
ccsecuritycode	CSC Code

Script Optimization

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

This section provides best practices for SuiteScript script optimization.

- As a general rule, design your user event scripts to execute in under 5 seconds, your Suitelets and Portlets to execute in under 10 seconds, and your scheduled scripts in under 5 minutes. This gives you a large enough margin of error to handle the outlier use cases (where the volume of work is unusually large, or the overall system is slow due to high load).
- Use nlapiLookupField instead of nlapiLoadRecord for fetching body field values. Note that you can fetch multiple fields at the same time.

  For detailed API information, see the help topics nlapiLookupField(type, id, fields, text) and nlapiLoadRecord(type, id, initializeValues).
- Use nlapiSubmitField instead of nlapiSubmitRecord for updating body field values. Note that you can submit multiple fields at the same time.

For detailed API information, see the help topics nlapiSubmitField(type, id, fields, values, doSourcing) and nlapiSubmitRecord(record, doSourcing, ignoreMandatoryFields).

- Use inline editable child custom records whenever your use case calls for batch processing of multiple related/child records during user events on the parent record. (See the help topic Custom Child Record Sublists in the NetSuite Help Center.)
- Use nlapiScheduleScript to schedule (asynchronously execute) long-running operations from user events and Suitelets.

  For detailed API information, see the help topic nlapiScheduleScript(scriptId, deployId, params).
- Avoid calls to nlapiGetOldRecord() in user event scripts unless absolutely required (field change comparisons in afterSubmit scripts). In a beforeSubmit script you can always do a search to get current field values.
- To minimize execution logging after your script is tested and released, set your script log level to ERROR or EMERGENCY. (See the help topic Setting Script Execution Log Levels.)
- Deploy scripts to run as admin only if absolutely necessary to minimize security risk and to eliminate performance overhead. (See the help topic Executing Scripts Using a Specific Role.)

SuiteScript Debugger

Note: The content in this help topic pertains to all versions of SuiteScript.

- SuiteScript Debugger Overview
- Before Using the SuiteScript Debugger
- Using the SuiteScript Debugger
- SuiteScript Debugger Interface
- SuiteScript Debugger Metering and Permissions
- SuiteScript Debugger Keyboard Shortcuts
- SuiteScript Debugger Glossary

SuiteScript Debugger Overview

Note: The content in this help topic pertains to all versions of SuiteScript.

You can use the SuiteScript Debugger to debug server-side scripts and core plug-in implementations.

To debug **client** scripts, NetSuite recommends using the Chrome DevTools for Chrome, the Firebug debugger for Firefox, and the Microsoft Script Debugger for Internet Explorer. For additional information about these tools, see the documentation provided with each browser.

Note: If you use the SuiteCloud IDE to develop your SuiteScripts, see the help topic SuiteCloud IDE Debugger. The SuiteCloud IDE Debugger enables you to debug server-side SuiteScripts in the same Eclipse-based environment you develop them in.

To work with the SuiteScript Debugger, you will need to learn about the following:

1. Accessing a **Debugger** domain as well as requirements for running the SuiteScript Debugger (see Before Using the SuiteScript Debugger)
2. SuiteScript Debugger metering and permission restrictions (see SuiteScript Debugger Metering and Permissions)
3. How to use the SuiteScript Debugger (see Using the SuiteScript Debugger)
4. SuiteScript Debugger tabs and buttons (see SuiteScript Debugger Interface)

To view script execution details either when using the SuiteScript Debugger, or after the script has been deployed into NetSuite, see the help topic Creating Script Execution Logs.

Before Using the SuiteScript Debugger

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Before using the SuiteScript Debugger, **you must be aware of the following** :

- There are separate Debugger domains, accessible by going to Customization > Scripting > Script Debugger from a production account, a release preview account, or a sandbox account. You can also access a Debugger domain by going directly to the following URLs:

- https://debugger.netsuite.com/pages/login.jsp.

  You must then log in to NetSuite.

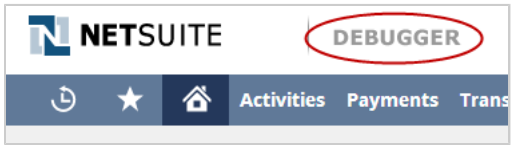
  > **Important:** Any changes you make to your account when on this Debugger domain **will affect the data in your production account**. For example, if you execute a script in the Debugger that creates a new record, that record will appear in your production account.

- https://debugger.sandbox.netsuite.com

  If you have a sandbox account on the sandbox domain, and choose to run the Debugger in this account, go to this Debugger domain.

  > **Note:** If you go to Customization > Scripting > Script Debugger in your sandbox account on the sandbox domain, you will be directed to the Debugger sandbox domain. Any changes you make to your account when on the Debugger sandbox domain will affect your sandbox account only.

When you are logged on to a Debugger domain, you see the Debugger logo at the top of your account.



> **Note:** There may be a decrease in performance when working on Debugger domains.

- The Debugger executes server-side scripts only (these script types include **Suitelets, Portlet** scripts, **Scheduled** scripts, and **User Event** scripts). Client scripts (both form- and record-level) should be tested on the form/record they run against.
- To debug scripts, the following must apply:
  - You must have scripting permission.
  - You must be the assigned owner of the script.
  - If you are debugging a script that already has a defined deployment, that script must be in *Testing* mode before it can be loaded into the Debugger. If you want to debug a script that has already been released into production, you must change the script's status from *Released* to *Testing* on the Script Deployment page.
- If a bundled script has been installed into your account and the script has been marked as *hidden*, you will be unable to debug this script.

Be aware that the SuiteScript Debugger is not any of the following:

- An API test console
- An integrated development environment (IDE)
- A script deployment interface. To deploy SuiteScript to NetSuite, you must still create a script record and define the script's deployment parameters on the Script Deployment page.
- A Client SuiteScript debugger
- A script runner interface (for example, scheduled scripts still need to be put INQUEUE for task completion)

Using the SuiteScript Debugger

Note: The content in this help topic pertains to all versions of SuiteScript.

The SuiteScript Debugger provides two debugging modes, which are based on the type of script you want to debug.

- On Demand Debugging: Enables you to debug code fragments written "on-the-fly." With on demand debugging, you are debugging a new script or code snippet that does not have a defined SuiteScript deployment. Scripts that do not require any form/record-specific interaction are good candidates for on demand debugging.
- Deployed Debugging: Enables you to select an existing script or core plug-in implementation that already has a defined Script Deployment or Plug-in Implementation record. The status of your script or implementation must be set to *Testing* before it can be loaded into the Debugger. You must also be the owner.

On Demand Debugging

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

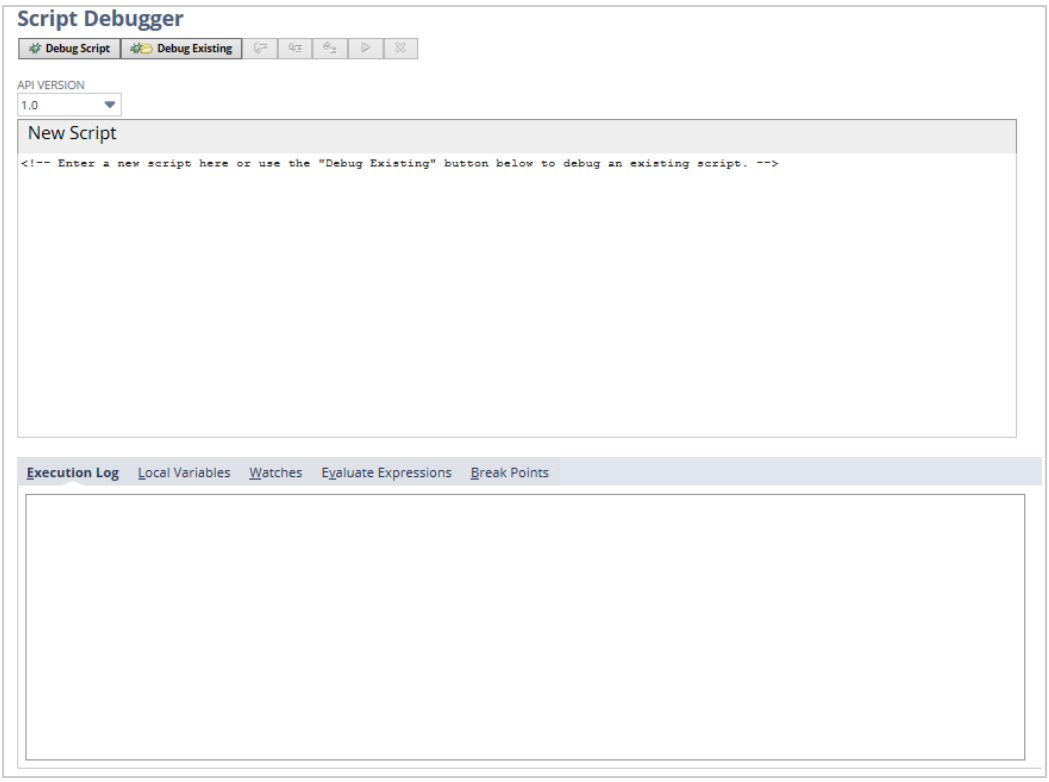
On demand debugging is for testing scripts or code snippets that do not have a defined SuiteScript deployment. On demand debugging is **not** for scripts that have a Script record or defined deployment parameters (set on the Script Deployment page).

To use the Debugger in on demand mode:

1. Go to Customization > Scripting > Script Debugger if you are already logged in to a production, release preview, or sandbox NetSuite account. Or, go directly to one of the following Debugger domains:
   - https://debugger.netsuite.com (if you run the Debugger in your production, release preview, or on your sandbox account on the NetSuite domain)
   - https://debugger.sandbox.netsuite.com (if you have a sandbox account on the sandbox domain, and choose the run the Debugger in this account)

   Important: See Before Using the SuiteScript Debugger to learn about how the changes you make to your account on a Debugger domain can affect your production, release preview, or sandbox account.

   The following figure shows the Script Debugger start page.



2. Type your code snippet in the **New Script** text area.

   If you have already written your code in an IDE, copy and paste the code into the Debugger text area. If you modify your script in the Debugger and you intend to save the changes, you must copy the updated script from the Debugger and paste it into your IDE.

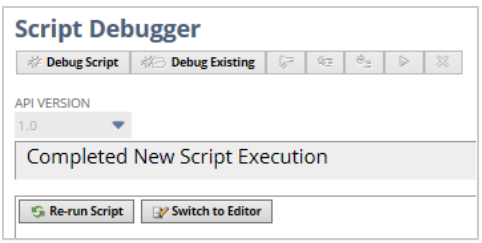
3. Next, click the **Debug Script** button.

   The script is immediately loaded into the Debugger and the program's execution stops before running the first line of executable code.

   > **Important:** Make sure you always have an executable call in your script; otherwise, the script has nothing to execute and return.

4. With the on demand script loaded into the Debugger, you can now step through each line to inspect local variables and object properties. You can also add watches, evaluate expressions, and set break points. See SuiteScript Debugger Interface for information on stepping into/out of functions, adding watches, setting and removing break points, and evaluating expressions.

5. After testing and debugging your on demand script, you can either re-run the script (by clicking the **Re-run Script** button), or put the script into edit mode (by clicking the **Switch to Editor** button) and continue debugging.



Deployed Debugging

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Deployed-mode debugging is for testing and inspecting scripts or core plug-in implementations that have a defined Script Deployment or Plug-in Implementation record.

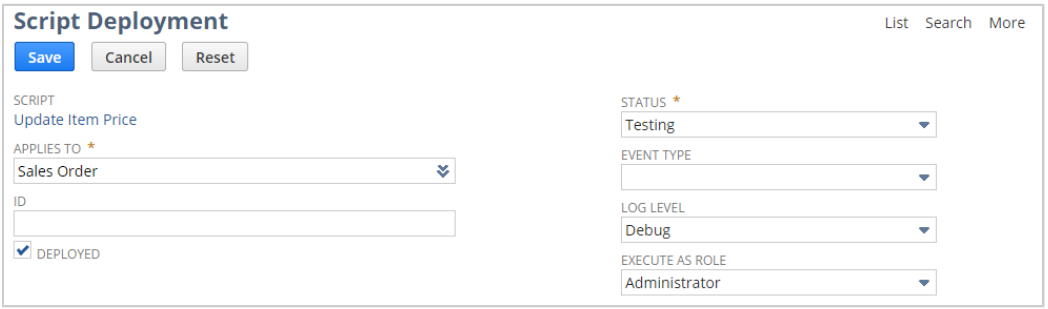
Note: SuiteScript does not support read-only sublists. If you are debugging a script that loads a record with a custom child record as a sublist, make sure the **Allow Child Record Editing** setting is checked for the child record in SuiteBuilder. If this box is not checked, the sublist is read-only and will not load in the parent record. See the help topic Creating Custom Record Types for additional information on creating custom records.

To use the Debugger in deployed-mode:

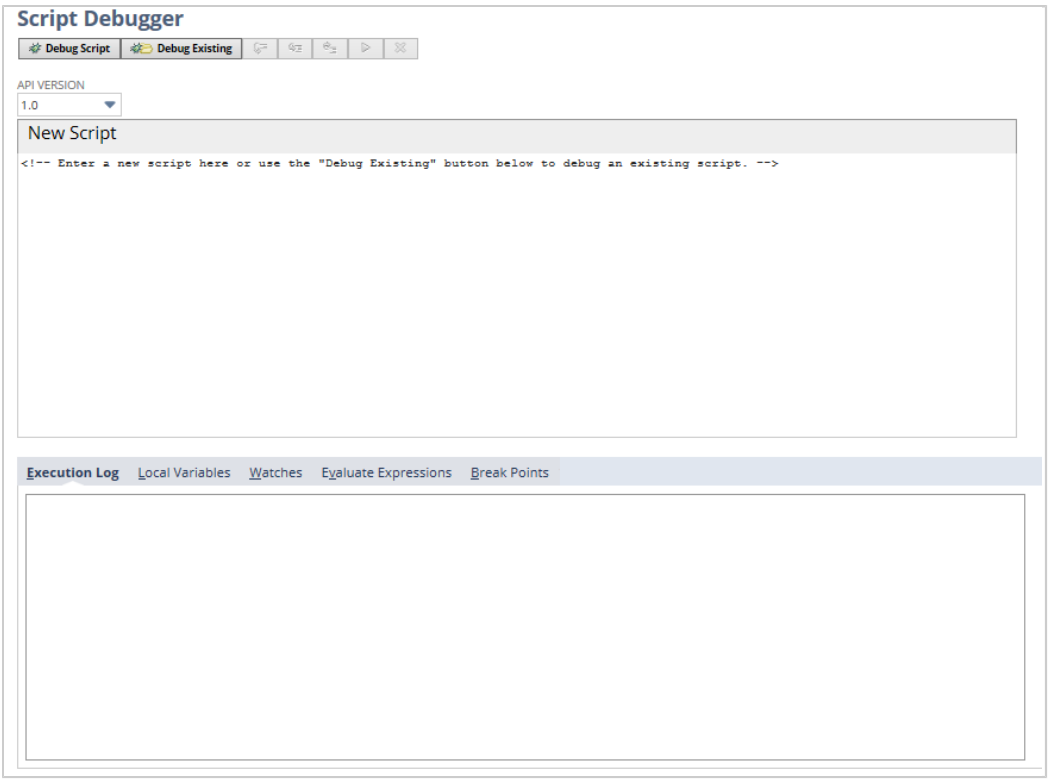
1. Go to Customization > Scripting > Script Debugger if you are already logged in to a production, release preview, or sandbox NetSuite account. Or, go directly to one of the following Debugger domains:
   - https://debugger.netsuite.com (if you run the Debugger in your production, release preview, or on your sandbox account on the NetSuite domain)
   - https://debugger.sandbox.netsuite.com (if you have a sandbox account on the sandbox domain, and choose the run the Debugger in this account)

   Important: See Before Using the SuiteScript Debugger to learn about how the changes you make to your account on a Debugger domain can affect your production, release preview, or sandbox account.

2. When you are on a Debugger domain, navigate to the applicable Script Deployment or Plug-in Implementation record and verify that **Status** is set to **Testing**. If you wish to debug a script or implementation that has already been released into production, you must change the status from **Released** to **Testing**.



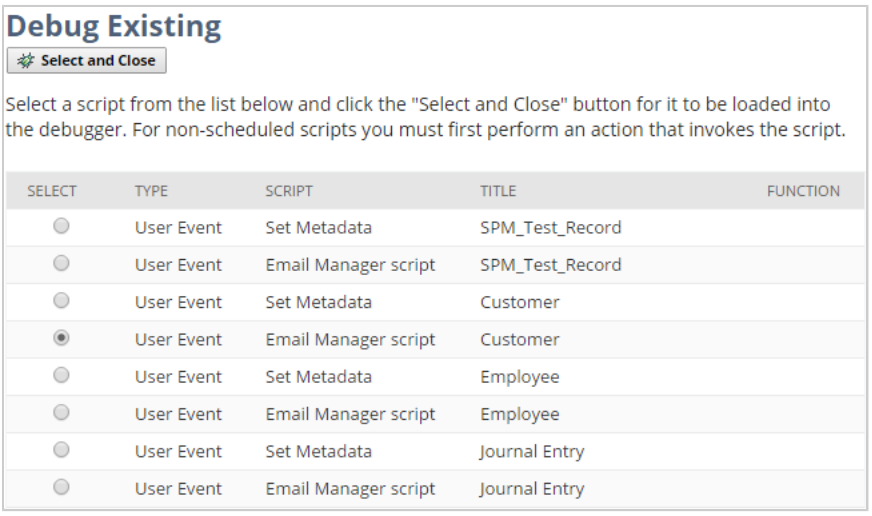
3. Verify that you are the assigned owner of the script or implementation. You can only debug if you are the assigned owner.
4. To get to the Debugger start page, use the appropriate option:
   - If you are on the Script Deployment page in View mode, click the **Debug** button. If you are on the Script Deployment page in Edit mode, click the **Save and Debug** button.
   - If you are not on the Script Deployment page or you are debugging an implementation, you can navigate to the Debugger by going to Customization > Scripting > Script Debugger.
5. When the Script Debugger page opens, click the **Debug Existing** button.



After clicking **Debug Existing**, the Script Debugger popup window opens. The popup shows all the server-side scripts and core plug-in implementations that are available for debugging (see figure below). Note that only scripts and implementations whose statuses are set to **Testing** will appear in the Script Debugger popup.

6. Select the script you want to debug, and then click the **Select and Close** button in the popup window.

   The following figure shows the selection of a beforeLoad User Event script that runs on Customer records. When a new Customer is created, this script sets a custom check box field called **Export to OpenAir** to true.

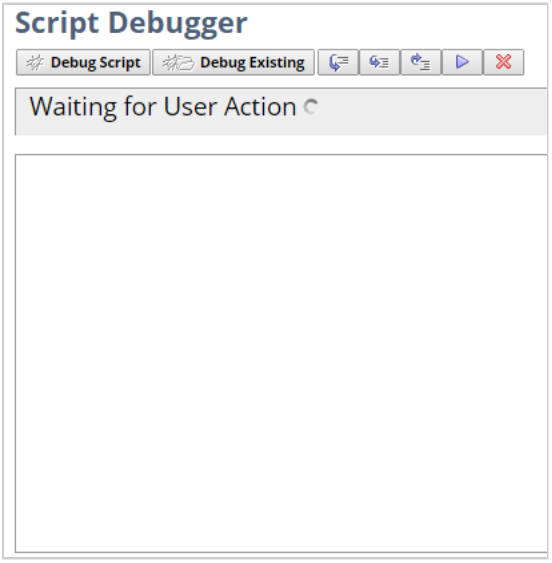


7. After clicking **Select and Close**, the Waiting for User Action screen appears. To load a User Event, Portlet, Suitelet, or Core Plug-in Implementations into the Debugger, you must perform the action that executes its logic.

> **Note:** Adding or updating records using CSV import may not trigger the debugging of scripts because the CSV import may create or update records using a different user role or entity. You can only debug scripts if you are the assigned owner.

> **Note:** You do not have to perform any user actions for Scheduled Scripts to load into the Debugger. Select your Scheduled Script from the Script Debugger popup window, and then click **Save and Close**. The Scheduled Script automatically loads.

In this example, you must create a new Customer for the User Event script to load into the Debugger. Performing the action that actually calls the script creates a "real-time," live context for script debugging.

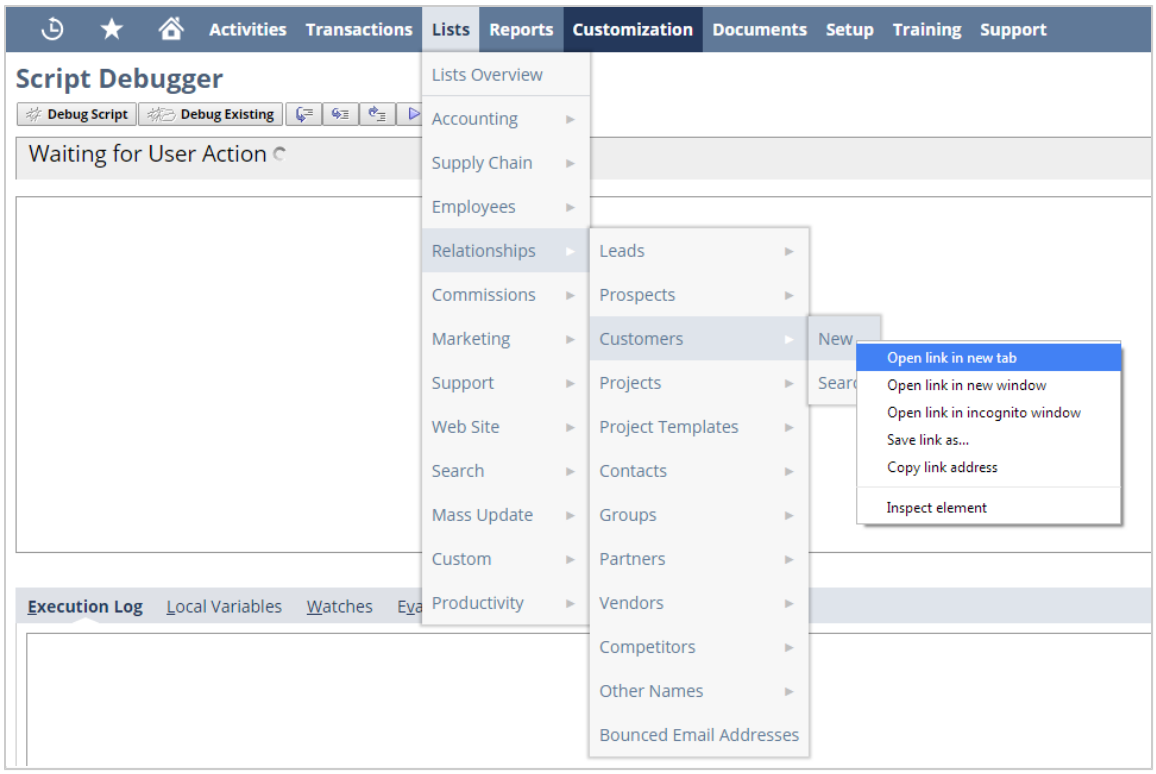


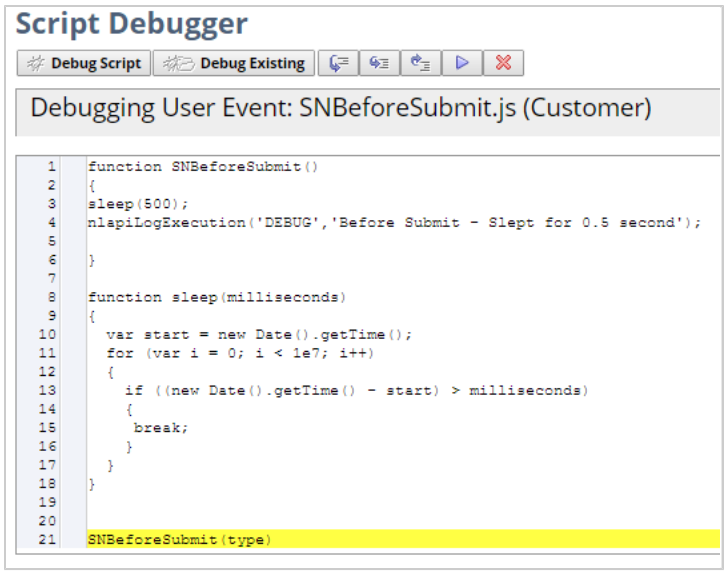
8. Next, create a new Customer to load the User Event script into the Debugger.

> **Important:** It is highly recommended that when performing user actions, you do so in another window or another tab (see figure below). Doing so enables you to keep the Debugger window open so that you can see the script or implementation as it loads.

> **Note:** The Debugger does not caption the user action during bulk processing, but the script is executed and a log is created.

- The first figure shows that a new Customer will be created in a separate window.
- The second figure shows the code in the script file as it loads into the Script Debugger window.

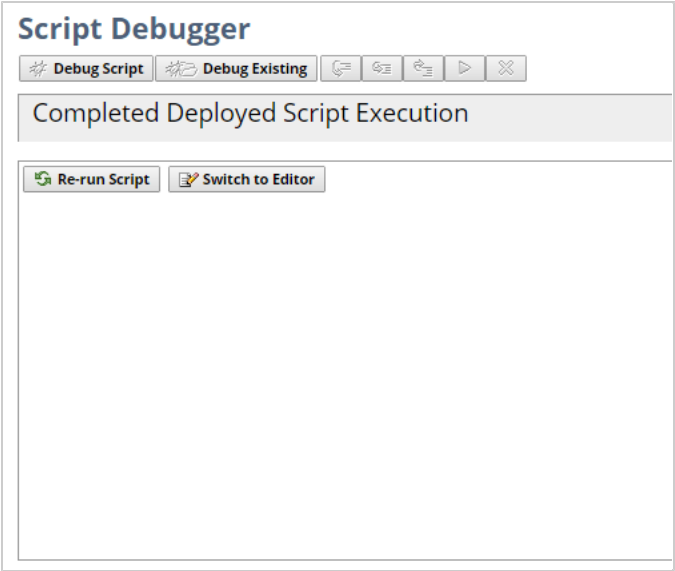




With the code in the Debugger text area, you have the following options:

- Click the Step Over button to begin stepping through each line of code.
- Add watches and evaluate expressions.
- Set break points and click the Run button to run the code. The Debugger will stop code execution at the first break point set.
- Set no break points, click the Run button, and have the Debugger execute the entire piece of code.

After testing or debugging a deployed-mode script or implementation, you can either re-run it (by clicking the Re-run Script button), or put it into edit mode (by clicking the Switch to Editor button) and continue debugging.



Important: If you modify a script or implementation in the Debugger, and you intend to save the changes, you must copy the updated script from the Debugger and paste it into your IDE. You must then re-load the updated .js file into the NetSuite file cabinet.

SuiteScript Debugger Interface

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

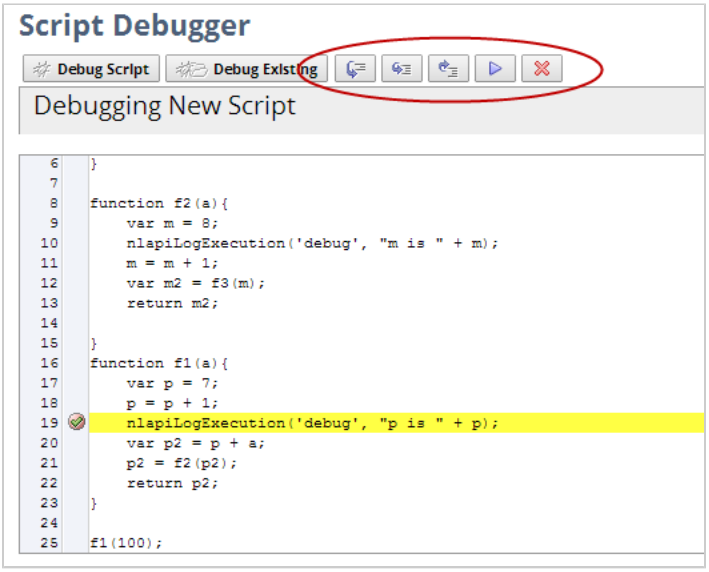
See the following sections to learn about the Debugger UI.

- SuiteScript Debugger Buttons
- SuiteScript Debugger Tabs

Use the Debugger buttons to control the flow of script execution. Use the tabs to inspect all script variables, objects, and properties.

SuiteScript Debugger Buttons

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.



There are five Debugger buttons that can be used to control/resume script execution when the Debugger stops at a particular line:

Note: Keyboard shortcuts are enabled for all five Debugger buttons. See SuiteScript Debugger Keyboard Shortcuts for details.

Icon	Name	Description
	Step Over	Resumes execution from the current line and stops at the next line (even if the current line is a function call).
	Step Into	Resumes execution from the current line and stops at the first line in any function call made from the current line.
	Step Out	Resumes execution from the current line until the end of the current function, and stops at the first line following the line from where this function was called -or- until the next break point -or- until the program terminates (either by error or by normal completion).
	Continue	Resumes program execution from the current line until the next break point -or- until the program terminates.
	Cancel	Aborts execution of the program from the current line.

SuiteScript Debugger Tabs

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The SuiteScript Debugger includes the following tabs. Click the links below to jump to information about each tab.

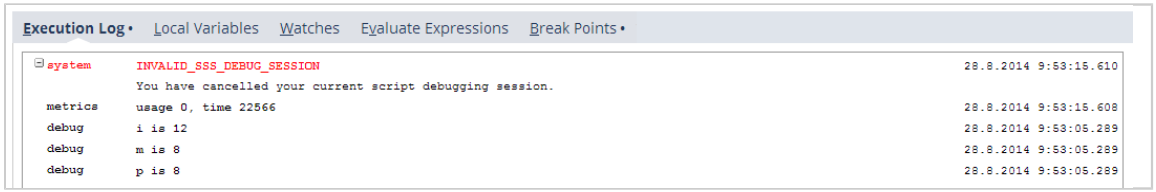
- Execution Log
- Local Variables
- Watches
- Evaluate Expressions
- Break Points

Execution Log

Click the Execution Log tab to view all the execution logs (including errors logged by the system) created by the currently executing program. The execution log details that appear on this tab are the same details that would normally appear in the Execution Log on the Script Deployment page. However, when working in the Debugger, all script execution details appear on the Execution Log tab within the Debugger; these details will NOT appear on the Execution Log tab on the Script Deployment page.

The type, subject, details, and timestamp are displayed in the on the Debugger Execution Log tab. The timestamp is recorded on the server but converted to the current user's time zone for display. The console is automatically cleared at the start of every debugging session.

Log details are collapsed by default, but can be seen by clicking the expand/collapse icon.

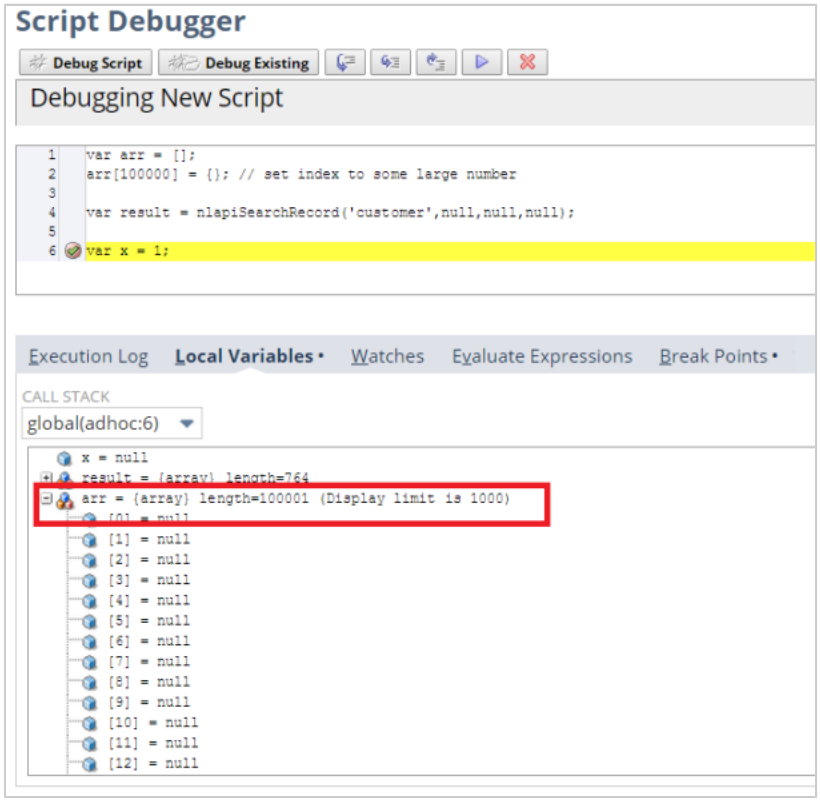


Local Variables

Click the Local Variables to see a browse-able list of all local variables (primitives, objects, and NetSuite objects) currently in scope. Note that for NetSuite (nlobj) objects, all properties are private, even though they can be seen on the Local Variables tab. Do not try to reference these properties directly in your script. Use the appropriate getter/setter functions instead.

Click the Call Stack dropdown list to see a list containing a browse-able view of the current execution stack of the program. The function call and current line number for that function are included in the Call Stack dropdown list. Use the Call Stack dropdown list to switch to different call stacks for local variable observation. In addition, watch expressions and expression evaluations will automatically be performed in the context specified by this field.

Important: Due to performance considerations, the member display limit for all variables is 1000. In addition, only the first 500 characters of a String are displayed. For large variables, use a watch to see the full member display (see Watches for additional information).

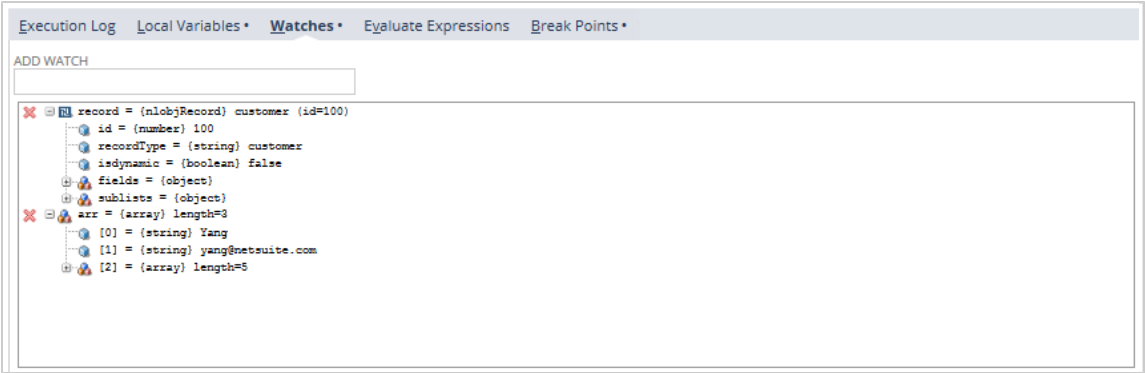


Watches

The Watch tab is where you can add or remove expressions to a list that is maintained and kept up-to-date ("watched") throughout the execution of a script. The expressions are always evaluated in the current call stack. This means that by default they are evaluated at the current line of script execution. However, if you switch to a different function in the call stack, they will be re-evaluated in that context.

- To add an expression, type it into the Add Watch field and press the Enter key.
- To remove a watch expression, click on the **x** icon to the left of the expression.
- To browse sub-properties of a user-defined object, an array, or a NetSuite object (nlobj) expression, click the expand/collapse icon next to the property name.

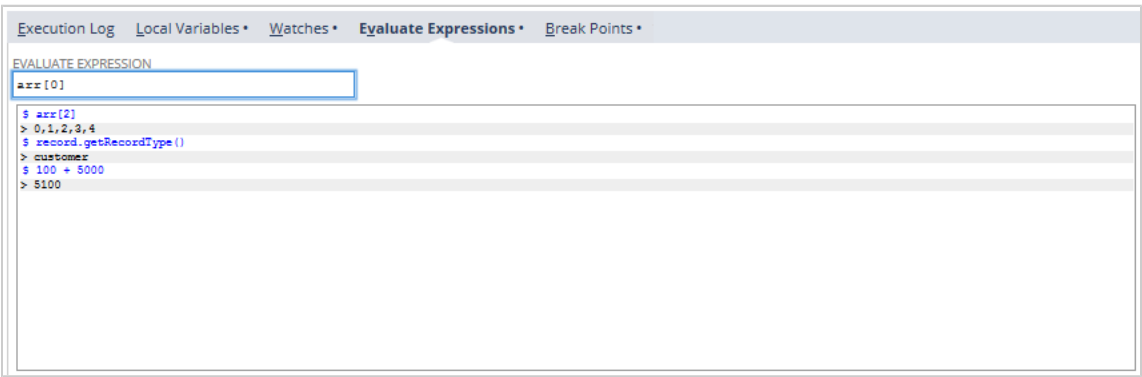
Note that you can use the watch window to navigate to object properties via a command line interface. Any property that is viewable from the property browser can be added as a watch expression by referencing the property using dot (.) notation, even if the property is private in the actual script (for example, you can reference the ID of an nlobjRecord object (referenced by a variable called *record*) by typing record *.id*).



Evaluate Expressions

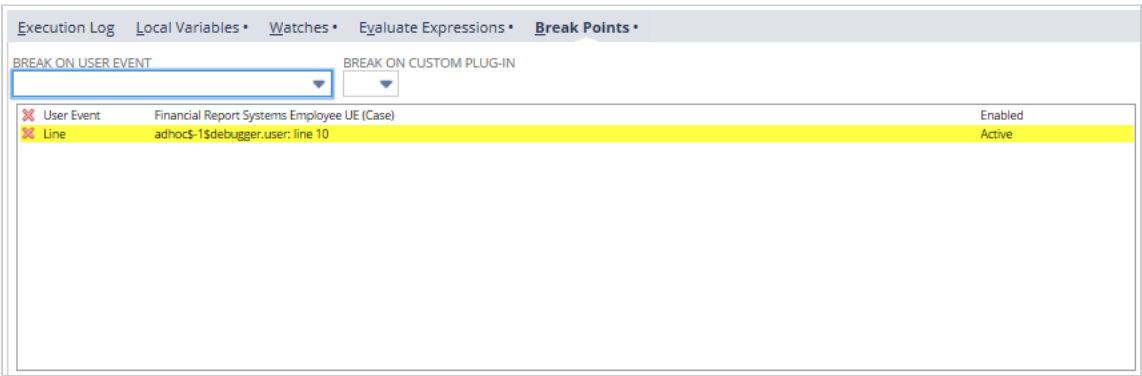
Use the Evaluate Expressions tab to execute code at break points during the current program. Doing so provides access to the program's state, allowing you to modify the state of the program.

Enter an expression in the Evaluate Expression field and press the Enter key to run it at the selected call stack. The results of an evaluated expression (if any) is displayed in the window below. Any changes to the program's state will immediately be reflected in the Local Variables and Watches windows.



Break Points

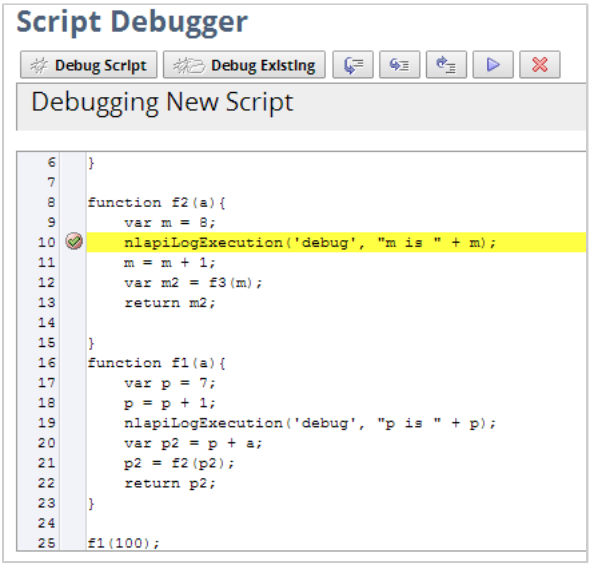
Click the Break Points tab to view all of your instruction-level (line) break points as well as your user event break points (see figure). Note that you can add user event break points by selecting user events from the Break on User Event dropdown list.



By setting breakpoints in your code, you can execute your code up to a certain point, and then halt the execution at the break point and examine the current state of the execution.

To add/remove break points in your code:

1. To add a break point, click between the line number and the actual line of code (see figure below).
2. To remove a break point, click the break point icon as it appears in the code. You can also remove a break point by clicking the **x** icon next to the break point (as it appears on the **Break Points** tab).



SuiteScript Debugger Metering and Permissions

Note: The content in this help topic pertains to all versions of SuiteScript.

The SuiteScript Debugger adheres to the following metering and permission restrictions:

- A user is only allowed to debug a single script at a time. Attempting to debug multiple scripts simultaneously (for example, by opening two different browser windows) will result in the same script/debugging session appearing in both windows.
- Users can debug only their own scripts in their current login session.
- There is a 1000 unit usage limit on all scripts being debugged. This is important to note, particularly for script types such as Scheduled scripts, which are permitted 10,000 units when running in NetSuite. If, for example, you load a 2,000 unit Scheduled script into the Debugger and attempt to step through or execute your code, the Debugger will throw a usage limit error when it reaches 1000 units.
- Email error notification is disabled for scripts being debugged. Additionally, execution log details are displayed on the Execution Log tab in the Debugger rather than in the execution log on the Script Deployment page.
- There is a two minute time limit on scripts sitting idle in the Debugger. If you do not perform some user action in the Debugger within the two minutes, the following error is thrown (see figure):

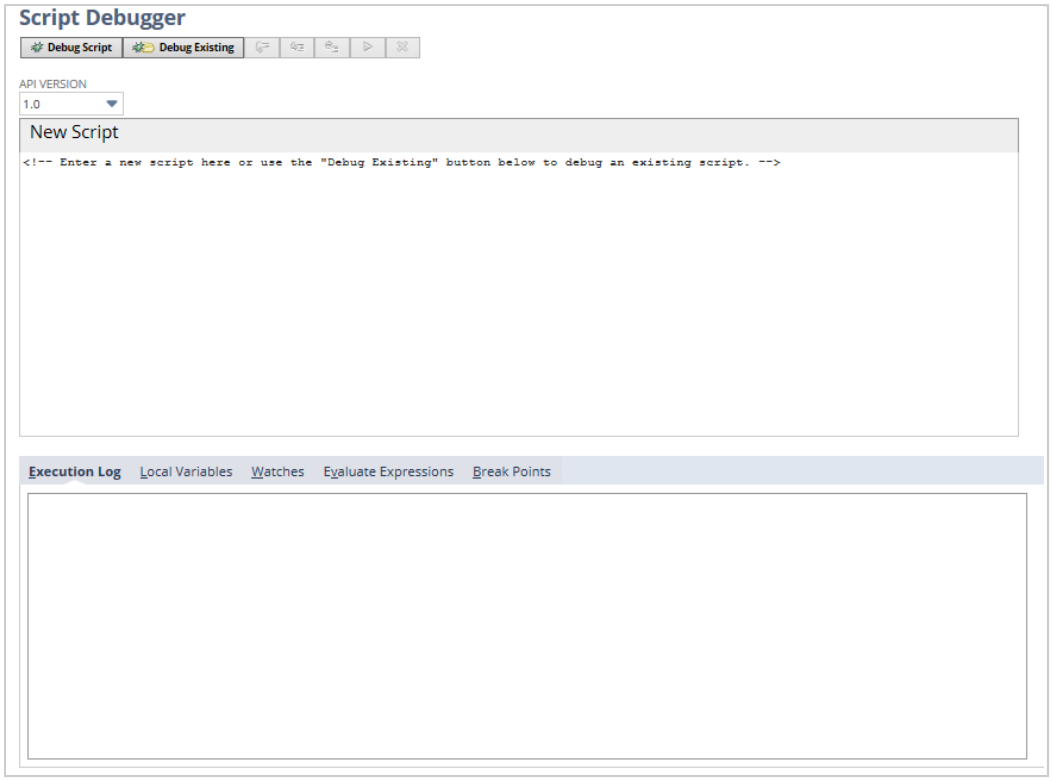
```
You have exceeded the maximum allowable idle time for debugging scripts. To debug another scrip
```

```
t,
simply reload the script debugger page and start a new debugging session.
```

If you receive this message and you are in deployed-debugging mode, click **Go Back** . You must then reload your script by clicking **Debug Existing** (see figure below).

If you are debugging in on demand mode, click the Go Back button, click in the Debugger text area, and re-type your code snippet.

(See Using the SuiteScript Debugger for information on deployed and on demand debugging modes.)



- There is a 10 minute global timeout on all scripts being debugged. This means that even if you are performing user actions within the Debugger every two minutes, the Debugger itself will timeout after 10 minutes.

SuiteScript Debugger Keyboard Shortcuts

Note: The content in this help topic pertains to all versions of SuiteScript.

The following table lists the keyboard shortcuts that are enabled for all five debugger action buttons. Your cursor must be inside the main Debugger text area for the keyboard shortcuts to work.

Key	Command
i	Step Into
space	Step Over
o	Step Out

Key	Command
shift+space	Continue
q	Quit
d	Debug Script (New)
a	Debug Script (Existing)
r	Re-run Script

SuiteScript Debugger Glossary

Note: The content in this help topic pertains to all versions of SuiteScript.

Keyword	Definition
Line Break Point	User-defined line in source code where program halts execution
User Event Break Point	User event possibly invokable during script execution where the program halts execution
Watch	A variable expression that will be monitored throughout the program's execution in the current scope
Call Stack	A stack (most recent on top) of all the active functions (and their local variables) called up until the current line of execution

SuiteCloud Processors

SuiteCloud Processors is the current system used to execute (process) scheduled scripts and map/reduce scripts. Before SuiteCloud Processors was introduced, scheduled scripts and map/reduce scripts were exclusively processed by scheduling queues. All scheduled script and map/reduce script jobs submitted to the same queue were processed on a FIFO (first in, first out) basis, based on the queue submission time stamp. This system had several limitations.

The scheduling queues did not provide automated load balancing or a way to prioritize specific jobs. Users with access to multiple queues (accounts with SuiteCloud Plus) were forced to manually determine the optimal configuration of jobs to queues. For the jobs that needed to be processed in a certain order, this method was useful. But it created unintended dependencies among many of the jobs submitted. If there was a delay in processing one job, a bottleneck would form. The result would be several jobs waiting in one queue when other queues were under utilized or not utilized at all.

SuiteCloud Processors resolves many of the limitations with scheduling queues. A scheduler now automatically determines the order in which jobs start to process. The scheduler uses algorithms that are based on user-defined priority levels, submission time, and user-defined preferences. The result is increased throughput, reduced wait times, and the elimination of most bottlenecks. In addition, SuiteCloud Processors requires less user intervention and enables map/reduce scripts and scheduled scripts to start sooner.

> **Note:** Some features of SuiteCloud Processors are available only to accounts that have one or more SuiteCloud Plus licenses. For more information about SuiteCloud Plus, see the help topic SuiteCloud Plus Settings.

For additional information about SuiteCloud Processors, see:

- SuiteCloud Processors – Terminology
- SuiteCloud Processors – Basic Architecture
- SuiteCloud Processors – Supported Task Types
- SuiteCloud Processors – Processor Allotment Per Account
- SuiteCloud Processors – Priority Levels
- SuiteCloud Processors – Priority Elevation and Processor Reservation (Advanced Settings)

SuiteCloud Processors – Terminology

Term	Definition
Job	A job is a piece of work submitted for processing. Each job is executed by a single processor.
Priority	A priority is a property of a job. The priority of submitted jobs determines the order in which the scheduler sends the jobs to the processor pool. Priorities are set on the deployment record or from the SuiteCloud Processors – Priority Settings Page.
Processor	A processor is a virtual unit of processing power that executes a job. It is not distinguished as an individual physical entity, but as a single processing thread.
Processor Pool	The processor pool represents the number of processors available to a specific account. For accounts without SuiteCloud Plus, the processor pool contains one processor. For accounts with SuiteCloud Plus, the processor pool contains multiple processors. For additional information, see the help topic SuiteCloud Plus Settings.

Term	Definition
Queue	With SuiteCloud Processors, a queue is no longer a separate processing mechanism. On scheduled script deployments, the **Queue** field remains to accommodate deployments that rely on the FIFO (first in, first out) order imposed by an individual queue. However, all jobs that use queues are actually processed by the same processor pool that handles the jobs that do not use queues. All jobs compete with each other using the same common processing algorithm.
Scheduler	The scheduler determines the order in which jobs are sent to the processor pool. The scheduler uses algorithms that are based on user-defined priority levels, submission time, and user-defined preferences.
Submit for Processing	
Task	A task is a script instance that is submitted for processing. Each task is handled by one or more jobs.

SuiteCloud Processors – Basic Architecture

SuiteCloud Processors currently supports the processing of scheduled scripts and map/reduce scripts. Each submitted scheduled script instance (task) is handled by one job. Each submitted map/reduce script task is handled by multiple jobs: one each for the getInput, shuffle, and summarize stages; and a minimum of one each for the map and reduce stages.

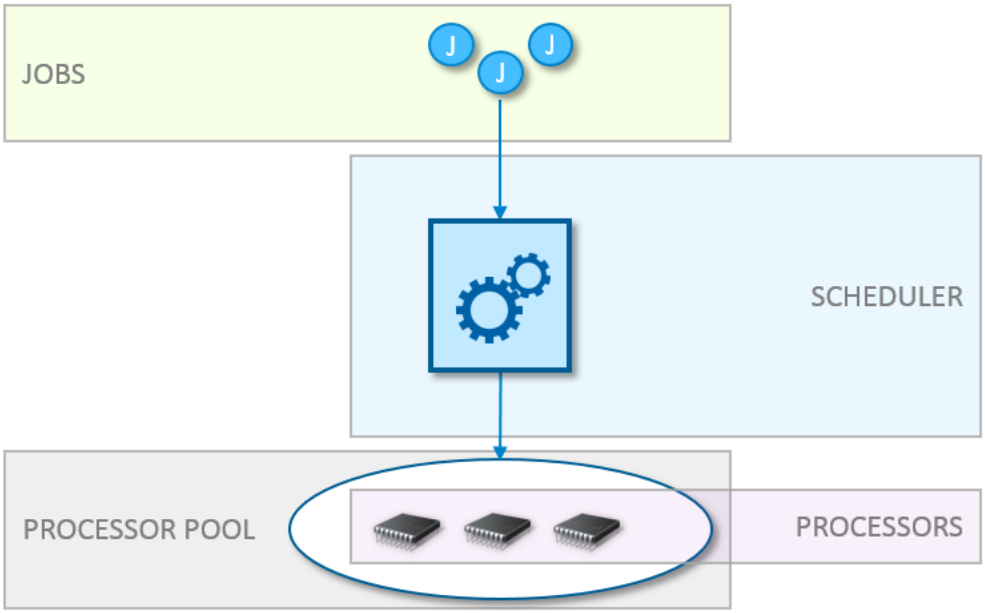
Scheduled script tasks and map/reduce script tasks are submitted for processing in one of three ways:

- By setting a one-time or recurring submission schedule from the script deployment record UI
- By selecting **Save and Execute** from the script deployment record UI to submit an on demand instance of the script
- By using a SuiteScript API to submit an on demand instance of the script

Note: For additional information on submitting a script with the deployment record, see the help topic SuiteScript 2.0 Scheduled Script Type (SuiteScript 2.0), SuiteScript 2.0 Map/Reduce Script Type (SuiteScript 2.0), or Scheduled Scripts (SuiteScript 1.0). For additional information on submitting a script with a SuiteScript API, see the help topic task.ScheduledScriptTask (SuiteScript 2.0), task.MapReduceScriptTask (SuiteScript 2.0), or nlapiScheduleScript(scriptId, deployId, params) (SuiteScript 1.0).

The scheduler sends the resulting jobs to the processor pool in a particular order. This order is determined by the SuiteCloud Processors – Priority Levels and the order of submission. Jobs with a higher priority are sent before jobs with a lower priority. Jobs with the same priority go to the processor pool in the order of submission.

SuiteCloud Processors includes advanced settings that can also impact the order in which jobs are sent to the processor pool. For additional information, see SuiteCloud Processors – Priority Elevation and Processor Reservation (Advanced Settings).



SuiteCloud Processors – Supported Task Types

SuiteCloud Processors currently supports processing for two task types:

- SuiteScript 2.0 Map/Reduce Script Type (SuiteScript 2.0 only)
- SuiteScript 2.0 Scheduled Script Type (SuiteScript 1.0 and SuiteScript 2.0)

SuiteCloud Processors Impact on Map/Reduce Deployments

All map/reduce script deployments include a **Priority** field. For additional information, see SuiteCloud Processors – Priority Levels.

For accounts with SuiteCloud Plus, the **Queues** multi-select field is replaced with the **Concurrency Limit** field. Instead of designating a specific queue, you set the maximum number of processors available to the deployment. This value equates to the number of jobs submitted for the map and reduce stages.

For map/reduce deployments created prior to the introduction of SuiteCloud Processors, the **Concurrency Limit** value is set by default. The default value is equivalent to the number of options selected on the **Queues** field. If **Queues** was previously set to 1, 3, 7, and 9, then **Concurrency Limit** is set to 4. If **Select All** was previously enabled, then **Concurrency Limit** is set to empty (the number of jobs initially created for the map and reduce stages is equivalent to the total number of processors in the processor pool).

For additional information, see the help topic SuiteScript 2.0 Map/Reduce Script Type.

SuiteCloud Processors Impact on Scheduled Script Deployments

All scheduled script deployments include a **Priority** field. For additional information, see SuiteCloud Processors – Priority Levels.

For all scheduled script deployments created **after** the introduction of SuiteCloud Processors, the **Queue** field is removed. These deployments cannot use queues.

For all scheduled script deployments created **prior** to the introduction of SuiteCloud Processors, the **Queue** field remains by default. This applies to accounts with and without SuiteCloud Plus. You have control over whether to stop using queues. The deployment record includes a **Remove Queue** option. After you select this option, the deployment no longer uses a queue **and cannot revert back to using a queue**.

The **Queue** field remains to accommodate deployments that rely on the FIFO order of processing imposed by an individual queue. However, all jobs that use queues are actually processed by the same processor pool that handles the jobs that do not use queues. All jobs compete with each other using the same common processing algorithm.

Note: For deployments that continue to use queues, all jobs assigned to the same queue should have the same priority. In most cases, you can keep the default (standard) priority of these jobs. However, in some cases, you may want to change these jobs to a higher or lower priority. One scenario is if you want to ensure that a specific queue always has a processor available. In that case, designate the jobs assigned to that queue as high priority. Alternatively, if you have a group of lower priority jobs, you can designate them as low priority and assign them to the same queue. That will ensure that only one is processed at a time.

Important: If your existing scheduled script deployments rely on implicit dependencies imposed by queues, you should update and test these scripts before you remove queues. Your scripts may be impacted if they rely on the sequence of FIFO (first in, first out).

One possible solution is to programmatically submit scripts in a certain order. To do this, use task.ScheduledScriptTask (SuiteScript 2.0) or nlapiScheduleScript(scriptId, deployId, params) (SuiteScript 1.0) within the first script to submit the second script. This will ensure that the jobs are submitted to the processor pool in the correct order.

SuiteCloud Processors – Processor Allotment Per Account

Important: SuiteCloud Processors is used to execute (process) all scheduled script and map/reduce script instances. However, this topic does not apply to scheduled script deployments that continue to use queues. See the help topic SuiteCloud Plus Settings to determine the number of queues available to deployments that continue to use queues.

- Processor Allotment for Map/Reduce and Scheduled Scripts
- Accounts Without a SuiteCloud Plus License

Processor Allotment for Map/Reduce and Scheduled Scripts

The following table illustrates SuiteCloud Plus support:

SuiteCloud Plus Licenses	Available Map/Reduce and Scheduled Script Processors
0	2 (see Accounts Without a SuiteCloud Plus License)
1	5

SuiteCloud Plus Licenses	Available Map/Reduce and Scheduled Script Processors
2	10
3	15
4	20
5	25
6	30
7	35
8	40
9	45
10 or more	50

Accounts Without a SuiteCloud Plus License

Accounts without a SuiteCloud Plus license now have access to two processors. The extra processor doubles the processing bandwidth for scheduled scripts and map/ reduce scripts.

If any of your scripts depend on implicit dependencies imposed by having one processor, you may be required to update your existing scripts for this feature. When an account has access to only one processor, all jobs are processed one at a time. If all jobs have the same priority, the order of processing is always first in, first out. Some scripts may depend on this behavior. With the addition of an extra processor, these scripts may no longer process in the correct order.

For example, if you have a script that pre-processes data for a second script, the first script must complete execution before the second script begins. With one processor, you can submit the scripts in the appropriate order and know that the order of processing is as expected. With two processors, there is a possibility that the second script submitted starts execution before the first script completes.

To handle the above scenario, perform the following steps:

1. Audit your scheduled scripts and map/reduce scripts. Look for scripts that depend on a specific order of execution.
2. Use the following SuiteScript APIs within the first script to programmatically submit the dependent script. This action ensures that the scripts are always executed in the correct order.

Script Type	SuiteScript 1.0 API	SuiteScript 2.0 API	Notes
Scheduled Script	nlapiScheduleScript(scriptId, deployId, params)	task.ScheduledScriptTask	Place the code at the end of the script.
Map/Reduce Script	Not Applicable	task.MapReduceScriptTask	Place the code at the end of the summarize stage.

SuiteCloud Processors – Priority Levels

With SuiteCloud Processors, script processing factors in submission time and priority level. This means that jobs with the highest priority are sent to the processor pool first. Jobs of lower priority are sent to the processor pool after all higher priority jobs are sent. The scheduler sends jobs with the same priority to the processor pool in the order of submission. For example, if jobs 1 and 4 are high

priority, jobs 2 and 5 are standard priority, and job 3 is low priority, the scheduler sends the jobs to the processor pool in the following order:

- Job 1
- Job 4
- Job 2
- Job 5
- Job 3

By default, all jobs have a standard priority. When you change the default priority, you change when the scheduler sends the job to the processor pool. You can do this on the deployment record or on the SuiteCloud Processors – Priority Settings Page. For examples that demonstrate how changing the priority can change the order of processing, see SuiteCloud Processors – Priority Scheduling Examples.

The available priority options are:

- High: Use to mark critical jobs that require more immediate processing. The scheduler sends these jobs to the processor pool first.
- Standard: This is the default setting. It is considered to be a medium priority level. The scheduler sends these jobs to the processor pool if there are no high priority jobs waiting.
- Low: Use to mark jobs that can tolerate a longer wait time. The scheduler sends these jobs to the processor pool if there are no high or standard priority jobs waiting.

SuiteCloud Processors – Priority Settings Page

The Priority Settings page lists each scheduled script deployment and map/reduce script deployment created for your account. Each line item corresponds to one deployment record. To access the Priority Settings page, go to Customization > Scripting > Priority Settings.

The primary purpose of this page is to manage the SuiteCloud Processors – Priority Levels for multiple deployments at once. You can also use this page to remove queues for scheduled script deployments.

Buttons

Button	Description
Submit	Saves all changes made to the page
Mark All Queues for Removal	Places a check in the Remove Queue column for all scheduled script deployments that are still using queues. Queues are not shown as removed until you submit the page. If none of your scheduled scripts use queues, this button no longer displays.
Unmark All Queues for Removal	Removes all check marks currently in the Remove Queues column If none of your scheduled scripts use queues, this button no longer displays.
Reset Priorities	Resets all values in the Priority column to Standard (the default setting)

Filters

Filter	Description
Type	Filters for all map/reduce script deployments or all scheduled script deployments. Columns not applicable to the selected script type are hidden.
Status	Filters for the values listed in the Status column

Filter	Description
API Version	Filters for script deployments by SuiteScript 1.0 or SuiteScript 2.0
Script	Filters for all deployments of a specified script record
Page Size	Determines the number of line items shown on each result page
Show Undeployed	If enabled, the results include deployments where the Deployed option is disabled

Columns

Column	Description
Internal ID	The internal ID for the script deployment record, as seen on the Script Deployments list page (for example, 345)
Edit | View	Links to the edit and view mode of the deployment record
ID	The ID of the script deployment record (for example, customdeploy_testscript1)
Script	Corresponds to the Name field value on the script record associated with the deployment record
API Version	Indicates whether the script is SuiteScript 1.0 or SuiteScript 2.0
Status	Indicates how and when a script can be submitted for processing. This value is set with the Status field on the deployment record. Possible values are: ▪ **Testing**: Indicates that you can test the script in the SuiteScript Debugger. The script deployment can be submitted for processing by the script owner only. ▪ **Scheduled**: Indicates that you can schedule a single or recurring instance of the script on the deployment record. You cannot submit an on demand instance of the script when it has this status. ▪ **Not Scheduled**: Indicates that you can submit an on demand instance of the script with the **Save and Execute** button or a SuiteScript API. You can submit an on demand instance of the script only if there is no other unfinished instance of the same script.
Remove Queue	Only applicable to scheduled script deployments. Select this box to stop using queues on an existing scheduled script deployment. You can remove queues for multiple deployments at once with this column. Queues are not shown as removed until you submit the page.
Type	Indicates whether the deployment is for a scheduled script or map/reduce script
Queue	Only applicable to scheduled script deployments. Shows a value if the deployment is still using queues. After you remove queues, this value is empty.
Concurrency Limit	Only applicable to map/reduce script deployments on accounts that use SuiteCloud Plus. The maximum number of processors available to a map/reduce script deployment. You set this value on the map/reduce deployment record.
Priority	The priority setting for the deployment. For additional information, see SuiteCloud Processors – Priority Levels

SuiteCloud Processors – Priority Scheduling Examples

The two examples in this section, Example 1 - Default Priority Scheduling and Example 2 – Varying Priority Scheduling, each show three diagrams. The two sets of diagrams compare the same 18 jobs as they are handled by:

- Pre-2017.2 Scheduling Queues
- 2017.2 SuiteCloud Processors: All queues are removed

- 2017.2 SuiteCloud Processors: Some queues are removed; jobs 1 - 3, 15, and 18 still use queues

This table shows the duration of each job submitted. As demonstrated in the examples, the processing environment has no impact on the amount of time required to complete each job after processing starts.

> **Note:** Each map/reduce stage is handled by a minimum of one job. Therefore, the duration listed for map/reduce jobs is per stage. For example, the reduce stage jobs 11 and 12 have a combined duration of 5. This means that the sum of the duration of jobs 11 and 12 is 5. When one of the jobs is canceled or not created, the combined duration becomes the duration of the remaining job from the pair.

Job	Duration
1	5
2	2
3	2
4	6
5 +6	2
7 + 8	1
9 + 10	5
11 + 12	5
13 + 14	2
15	3
16	3
17	3
18	2

Example 1 - Default Priority Scheduling

This example assumes default priority settings (standard priority) and default preferences. It also assumes that each job executes in full without yielding.

The map/reduce deployment in the first diagram has a **Queues** value of 2 and 3. The map/reduce deployment in the second and third diagrams has a **Concurrency Limit** value of 2.

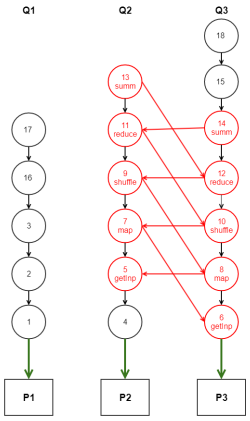
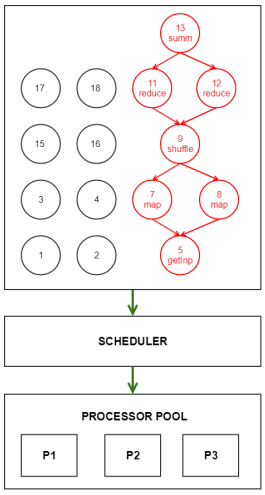
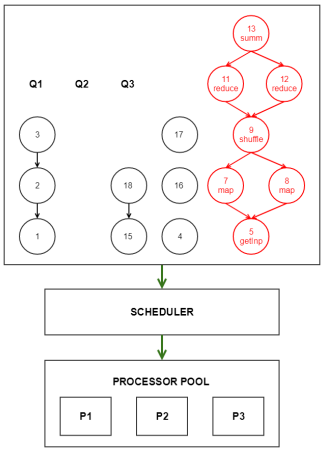
> **Note:** With the introduction of SuiteCloud Processors, the **Queues** field is replaced with the **Concurrency Limit** field on map/reduce deployments.

Diagram Key

- The black circles are scheduled script jobs. Each scheduled script is processed by a single job.
- The red circles are map/reduce jobs. In this example, all of the map/reduce jobs belong to a single map/reduce task. The map/reduce jobs include a label that indicates the map/reduce stage being processed.
- The jobs in columns labeled Q1, Q2, and Q3 are for deployments that still use queues. The jobs in columns without a label are for deployments that no longer use queues.
- P1, P2, and P3 represent the processors that are available in the processor pool.

> **Note:** Although processors are labeled for the purpose of this example, they are not technically individual entities. Also, a real account would never have only three processors available to it. The minimum number of processors available to an account with SuiteCloud Plus is listed at SuiteCloud Plus Settings.

- The numbers within each circle represent the time stamp on the job submission, with 1 being the first job submitted. The example also uses this number as a unique identifier for each job.
- Arrows represent dependencies. For example, in the Scheduling Queues diagram, job 2 cannot start until job 1 is complete.
  - Black arrows represent queue dependencies. Jobs within a queue must be processed in the order they were submitted.
  - Red arrows represent dependencies imposed by the map/reduce algorithm.
  - Green arrows indicate access to processors.

Scheduling Queues Only	SuiteCloud Processors – All Queues Removed	SuiteCloud Processors – Some Queues Removed
		

Time Slot	Scheduling Queues Only	SuiteCloud Processors – All Queues Removed	SuiteCloud Processors – Some Queues Removed
101	Jobs **1**, **4**, and **6** start.	Jobs **1**, **2**, and **3** start.	Jobs **1**, **4**, and **5** start.
102	Job **6** cancels job **5**. The getInput stage cannot be processed by multiple jobs, so the first getInput job to start (job **6**) automatically cancels all others in the queues.		
103	Job **6** completes. The next job in Q3, job **8**, starts.	Jobs **2** and **3** complete. Jobs **4** and **5** start.	Job **5** completes. There is no job **6** with SuiteCloud Processors. In the Scheduling Queues example, jobs **5** and **6** are both getInputstage jobs. The getInput stage cannot be processed by multiple jobs. Since this map/reduce deployment is not using queues, multiple getInput

Time Slot	Scheduling Queues Only	SuiteCloud Processors – All Queues Removed	SuiteCloud Processors – Some Queues Removed
			stage jobs are no longer needed. Job **7** starts.
104	Job **8** completes. The map stage can be processed by multiple jobs. But since the other map stage job, job **7**, is unable to start (the job utilizing its processor, job **4**, is not yet complete), job **8** completes the entire map stage. Job **7** is no longer needed, so it is canceled. The next job in Q3, job **10**, starts.		Job **7** completes. Since job **7** completes the entire map stage, job 8 is no longer needed. Job **8** is canceled. Job **9** starts.
105	Job **10** cancels job **9**. The shuffle stage cannot be processed by multiple jobs, so the first shuffle job to start (job **10**) automatically cancels all others in the queues.	Job **5** completes. There is no job **6** with SuiteCloud Processors. In the Scheduling Queues example, jobs **5** and **6** are both getInputstage jobs. The getInput stage cannot be processed by multiple jobs. Since this map/reduce deployment is not using queues, multiple getInput stage jobs are no longer needed. Job **7** starts.	
106	Job **1** completes. The next job in Q1, job **2**, starts.	Job **7** completes. Since job **7** completes the entire map stage, job **8** is no longer needed. Job **8** is canceled. Job **1** completes. Jobs **9** and **15** start. Jobs **11 - 13** are dependent on job **9**. There is no job **10** or job **14** with SuiteCloud Processors. In the Scheduling Queues example, jobs **9** and **10** are both shuffle stage jobs and jobs **13** and **14** are both summarize stage jobs. The shuffle and summarize stages cannot be processed by multiple jobs. Since this map/reduce deployment is not using queues, multiple shuffle and summarize stage jobs are no longer needed.	Job **1** completes. The next job in Q1, job **2**, starts.
107	Job **4** completes. Q2 is blocked. The next job in Q2, job **11**, cannot start. It is dependent on job **10**, and job **10** is not complete.		Job **4** completes. The first job in Q3, Job **15**, starts.
108	Job **2** completes. The next job in Q1, job **3**, starts.		Job **2** completes. The next job in Q1, job **3**, starts.
109	Job **10** completes. Job **11** can now start and Q2 is no longer blocked. The next job in Q3, job **12**, starts. These are both reduce stage jobs. The reduce stage can be processed by multiple jobs.	Jobs **4** and **15** complete. Jobs **16** and **17** start.	Job **9** completes. There is no job **10** with SuiteCloud Processors. In the Scheduling Queues example, jobs **9** and **10** are both shuffle stage jobs. The shuffle stage cannot be processed by multiple jobs. Since this map/reduce deployment is not using

Time Slot	Scheduling Queues Only	SuiteCloud Processors – All Queues Removed	SuiteCloud Processors – Some Queues Removed
			queues, multiple shuffle stage jobs are no longer needed. Job **11** starts.
110	Job **3** completes. The next job in Q1, job **16**, starts.		Jobs **3** and **15** complete. There are no additional jobs to process in Q1. Jobs **12** and **16** start.
111	Job **11** completes. Q2 is blocked. The next job in Q2, job **13**, cannot start. It is dependent on job **12**, and job **12** is not complete.	Job **9** completes. Job **11** starts.	
112	Job **12** completes. Job **13** can now start and Q2 is no longer blocked. The next job in Q3, job **14**, can also start. But these are both summarize stage jobs, and the summarize stage cannot be processed by multiple jobs. Job **13** starts first, so job **14** is canceled. Since job **14** is canceled, the next job in Q3, job **15**, starts	Jobs **16** and **17** complete. Jobs **12** and **18** start.	Jobs **11** and **12** complete. Jobs **13** and **17** start.
113	Job **16** completes. The next job in Q1, job **17**, starts.		Job **16** completes. The next job in Q3, Job **18**, starts.
114	Job **13** completes. There are no additional jobs to process in Q2.	Jobs **11**, **12**, and **18** complete. Job **13** starts	Job **13** completes. There is no job **14** with SuiteCloud Processors. In the Scheduling Queues example, jobs **13** and **14** are both summarize stage jobs. The summarize stage cannot be processed by multiple jobs. Since this map/reduce deployment is not using queues, multiple summarize stage jobs are no longer needed.
115	Job **15** completes. The next job in Q3, job **18**, starts.		Jobs **17** and **18** complete. There are no additional jobs to process.
116	Job **17** completes. There are no additional jobs to process in Q1.	Job **13** completes. There are no additional jobs to process.	
117	Job **18** completes. There are no additional jobs to process in Q3.		

Example 2 – Varying Priority Scheduling

This example assumes default preferences. It also assumes that each job executes in full without yielding.

The map/reduce deployment in the first diagram has a **Queues** value of 2 and 3. The map/reduce deployment in the second and third diagrams has a **Concurrency Limit** value of 2.

> **Note:** With the introduction of SuiteCloud Processors, the **Queues** field is replaced with the **Concurrency Limit** field on map/reduce deployments.

The second and third diagrams vary from Example 1 - Default Priority Scheduling as follows:

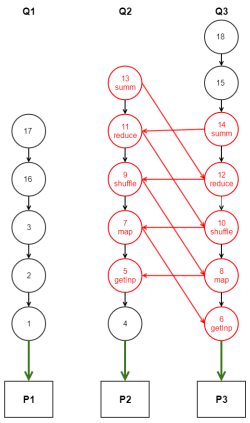
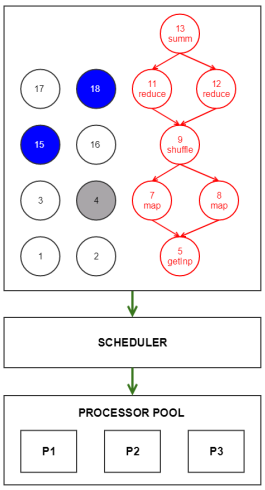
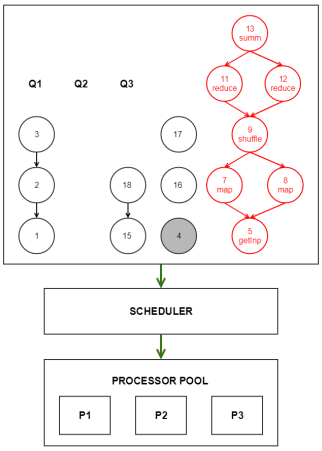
- In the second diagram, jobs 15 and 18 are high priority. Job 4 is low priority. The remaining jobs are standard priority.
- In the third diagram, job 4 is low priority. Jobs 15 and 18 in the third diagram are standard priority.

Diagram Key

- The black circles are scheduled script jobs. Each scheduled script is processed by a single job.
- The red circles are map/reduce jobs. In this example, all of the map/reduce jobs belong to a single map/reduce task. The map/reduce jobs include a label that indicates the map/reduce stage being processed.
- **The circles with blue fill are high priority. The circles with gray fill are low priority. The circles with white fill are standard priority.**
- The jobs in columns labeled Q1, Q2, and Q3 are for deployments that still use queues. The jobs in columns without a label are for deployments that no longer use queues.
- P1, P2, and P3 represent the processors that are available in the processor pool.

  > **Note:** The processors are labeled for the purpose of this example. Although technically, processors are not distinguished as individual entities. Also, a real account would never have only three processors available to it. The minimum number of processors available to an account with SuiteCloud Plus is listed at SuiteCloud Plus Settings.

- The numbers within each circle represent the time stamp on the job submission, with 1 being the first job submitted. The example also uses this number as a unique identifier for each job.
- Arrows represent dependencies. For example, in the Scheduling Queues diagram, job 2 cannot start until job 1 is complete.
  - Black arrows represent queue dependencies. Jobs within a queue must be processed in the order they were submitted.
  - Red arrows represent dependencies imposed by the map/reduce algorithm.
  - Green arrows indicate access to processors.

Scheduling Queues Only	SuiteCloud Processors – All Queues Removed	SuiteCloud Processors – Some Queues Removed
		

Time Slot	Scheduling Queues Only	SuiteCloud Processors – All Queues Removed	SuiteCloud Processors – Some Queues Removed
101	Jobs **1**, **4**, and **6** start.	Jobs **15**, **18**, and **1** start. Jobs **15** and **18** are designated as high priority so they are assigned to processors first (in the order of submission). Job **1** is the first standard priority job submitted so it is the next job assigned to a processer.	Jobs **1**, **5**, and **15** start. Job **4** is designated low priority, so it is processed after all other available standard priority jobs. All other standard priority jobs submitted before job **15** have dependencies that are blocking them.
102	Job **6** cancels job **5**. The getInput stage cannot be processed by multiple jobs, so the first getInput job to start (job **6**) automatically cancels all others in the queues.		
103	Job **6** completes. The next job in Q3, job **8**, starts.	Job **18** completes. Job **2** starts.	Job **5** completes. There is no job **6** with SuiteCloud Processors. In the Scheduling Queues example, jobs **5** and **6** are both getInputstage jobs. The getInput stage cannot be processed by multiple jobs. Since this map/reduce deployment is not using queues, multiple getInput stage jobs are no longer needed. Job **7** starts.
104	Job **8** completes. The map stage can be processed by multiple jobs. But since the other map stage job, job **7**, is unable to start (the job utilizing its processor, job **4**, is not yet complete), job **8** completes the entire map stage. Job **7** is no longer needed, so it is canceled. The next job in Q3, job **10**, starts.	Job **15** completes. Job **3** starts.	Jobs **7** and **15** complete. Since job **7** completes the entire map stage, job 8 is no longer needed. Job **8** is canceled. Jobs **9** and **16** start.

Time Slot	Scheduling Queues Only	SuiteCloud Processors – All Queues Removed	SuiteCloud Processors – Some Queues Removed
105	Job **10** cancels job **9**. The shuffle stage cannot be processed by multiple jobs, so the first shuffle job to start (job **10**) automatically cancels all others in the queues.	Job **2** completes. Job **5** starts. Job **4** is designated low priority, so it is processed after all other available standard priority jobs.	
106	Job **1** completes. The next job in Q1, job **2**, starts.	Jobs **1** and **3** complete. Jobs **16** and **17** start.	Job **1** completes. The next job in Q1, job **2**, starts.
107	Job **4** completes. Q2 is blocked. The next job in Q2, job **11**, cannot start. It is dependent on job **10**, and job **10** is not complete.	Job **5** completes. There is no job **6** with SuiteCloud Processors. In the Scheduling Queues example, jobs **5** and **6** are both getInputstage jobs. The getInput stage cannot be processed by multiple jobs. Since this map/reduce deployment is not using queues, multiple getInput stage jobs are no longer needed. Job **7** starts.	Job **16** completes. Job **17** starts.
108	Job **2** completes. The next job in Q1, job **3**, starts.	Job **7** completes. Since job **7** completes the entire map stage, job 8 is no longer needed. Job **8** is canceled. Job **9** starts.	Job **2** completes. The next job in Q1, job **3**, starts.
109	Job **10** completes. Job **11** can now start and Q2 is no longer blocked. The next job in Q3, job **12**, starts. These are both reduce stage jobs. The reduce stage can be processed by multiple jobs.	Jobs **16** and **17** complete. Job **4** starts. Job **4** is designated as low priority, but all the remaining standard priority jobs are dependent on job **9**.	Job **9** completes. There is no job **10** with SuiteCloud Processors. In the Scheduling Queues example, jobs **9** and **10** are both shuffle stage jobs. The shuffle stage cannot be processed by multiple jobs. Since this map/reduce deployment is not using queues, multiple shuffle stage jobs are no longer needed. Job **11** starts.
110	Job **3** completes. The next job in Q1, job **16**, starts.		Jobs **3** and **17** complete. There are no additional jobs to process in Q1. Job **12** starts. The next job in Q3, Job **18**, starts.
111	Job **11** completes. Q2 is blocked. The next job in Q2, job **13**, cannot start. It is dependent on job **12**, and job **12** is not complete.		
112	Job **12** completes. Job **13** can now start and Q2 is no longer blocked. The next job in Q3, job **14**, can also start. But these are both summarize stage jobs, and the summarize stage cannot be processed by multiple jobs. Job **13** starts first, so job **14** is canceled. Since job **14** is canceled, the next job in Q3, job **15**, starts		Jobs **11**, **12** and **18** complete. There are no additional jobs to process in Q3. Jobs **13** and **4** start.
113	Job **16** completes. The next job in Q1, job **17**, starts.	Job **9** completes. There is no job **10** with SuiteCloud Processors. In	

Time Slot	Scheduling Queues Only	SuiteCloud Processors – All Queues Removed	SuiteCloud Processors – Some Queues Removed
		the Scheduling Queues example, jobs **9** and **10** are both shuffle stage jobs. The shuffle stage cannot be processed by multiple jobs. Since this map/reduce deployment is not using queues, multiple shuffle jobs are no longer needed. Jobs **11** and **12** start.	
114	Job **13** completes. There are no additional jobs to process in Q2.		Job **13** completes. There is no job **14** with SuiteCloud Processors. In the Scheduling Queues example, jobs **13** and **14** are both summarize stage jobs. The summarize stage cannot be processed by multiple jobs. Since this map/reduce deployment is not using queues, multiple summarize stage jobs are no longer needed.
115	Job **15** completes. The next job in Q3, job **18**, starts.	Jobs **4** and **12** complete.	
116	Job **17** completes. There are no additional jobs to process in Q1.	Job **11** completes. Job **13** starts.	
117	Job **18** completes. There are no additional jobs to process in Q3.		
118		Job **13** completes. There is no job **14** with SuiteCloud Processors. In the Scheduling Queues example, jobs **13** and **14** are both summarize stage jobs. The summarize stage cannot be processed by multiple jobs. Since this map/reduce deployment is not using queues, multiple summarize stage jobs are no longer needed. There are no additional jobs to process.	Job **4** completes. There are no additional jobs to process.

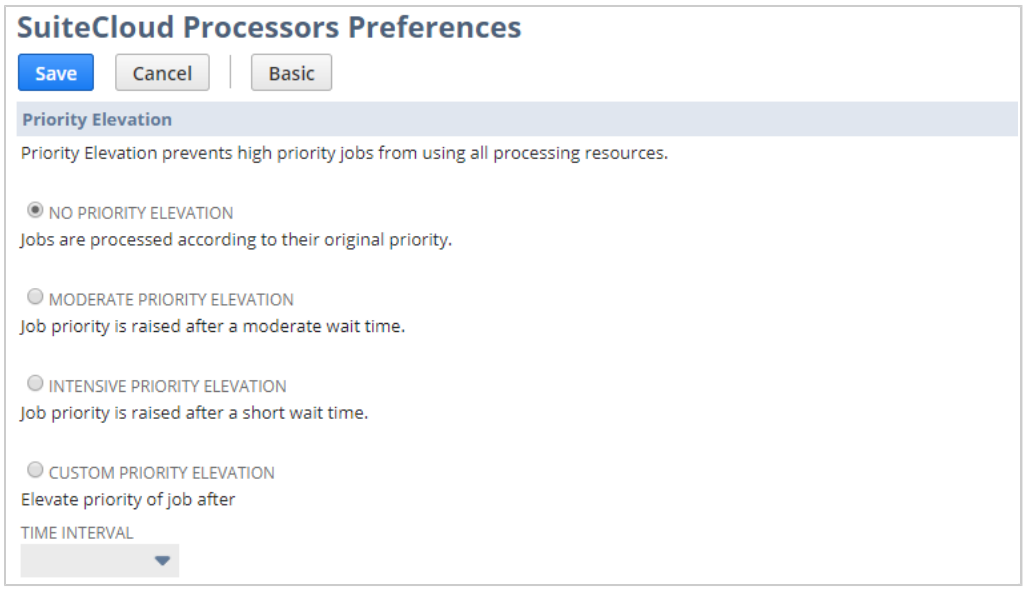
SuiteCloud Processors – Priority Elevation and Processor Reservation (Advanced Settings)

Priority Elevation

Priority elevation enables you to automatically increase the priority of each low or standard priority job after a specific time interval. The time interval starts when the job is submitted. You do not need to use these settings in most cases. Because of this, priority elevation is disabled by default. However, you may need to use them if lower priority jobs experience excessive wait times.

Note: Priority elevation only impacts lower priority jobs with a wait time greater than the time interval indicated. If a lower priority job is sent to the processor pool before the time interval is over, that job is processed with its original priority.

To access the priority elevation settings, go to Setup > Preferences > SuiteCloud Processors.



There are three predetermined settings listed:

- **No Priority Elevation**: The default setting
- **Moderate Priority Elevation**:
  - If a low priority job is still waiting after four hours, it is elevated to standard priority
  - If a standard priority job is still waiting after four hours, it is elevated to high priority

  With this option, if priority elevation is applicable, the system elevates low priority jobs to high priority jobs after eight hours. Specifically, the jobs in this scenario are elevated to standard priority after four hours and then elevated to high priority after another four hours.
- **Intensive Priority Elevation**:
  - If a low priority job is still waiting after one hour, it is elevated to standard priority
  - If a standard priority job is still waiting after one hour, it is elevated to high priority

  With this option, if priority elevation is applicable, the system elevates low priority jobs to high priority jobs after two hours. Specifically, the jobs in this scenario are elevated to standard priority after one hour and then elevated to high priority after another hour.
- **Custom Priority Elevation**: This option enables you to specify a custom time interval for priority elevation from the **Time Interval** dropdown list.

The **Time Interval** field specifies the time interval set for priority elevation. When **Custom Priority Elevation** is selected, this field is enabled for editing. Otherwise, the field displays a value that corresponds with the option selected, but it cannot be edited.

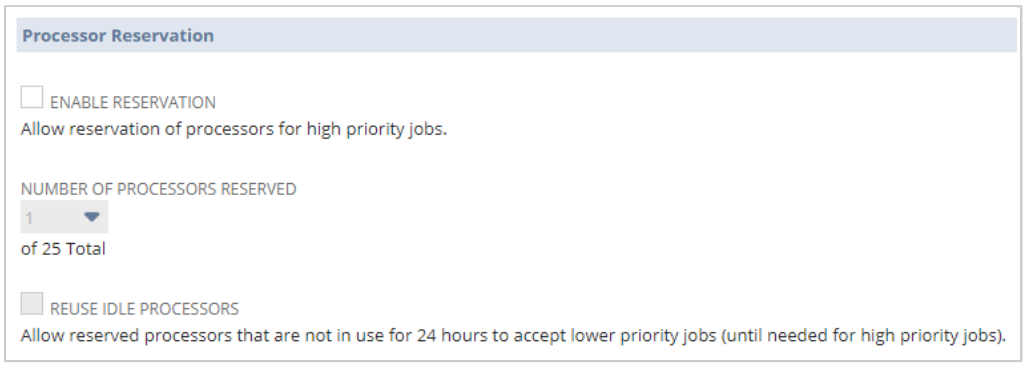
Note: Click **Advanced** at the top of the page to access **Custom Priority Elevation** and **Time Interval**.

Processor Reservation

Processor reservation enables you to reserve a specified number of processors for high priority jobs. You do not need to use these settings in most cases. Because of this, processor reservation is disabled by default.

Important: Processor reservation is available for SuiteCloud Plus accounts only.

To access the priority elevation settings, go to Setup > Preferences > SuiteCloud Processors. Click **Advanced** at the top of the page.



When you select **Enable Reservation**, you can reserve all but one of your available processors from the **Number of Processors Reserved** dropdown list. If a high priority job is submitted, it is sent to the processor pool if there is at least one processor available. If a standard or low priority job is submitted, it is sent to the processor pool only if there are more processors available than the number reserved. For example, you have 10 processors reserved out of 25 total processors. A standard or low priority job is sent to the processor pool if there are at least 11 processors available. If there are 10 processors or fewer available, the lower priority job must wait.

Important: Changes to the Number of Processors Reserved apply to all jobs that have not yet started. This can have an immediate impact on map/reduce scripts since each stage is processed by a minimum of one job. If there is a high priority map/reduce script instance (task) executing and this setting is changed, the new value is applied to all jobs for this task that have not yet started. This includes jobs that may be created from yielding.

Processor reservation decreases the number of processors available for standard and low priority jobs. Therefore, it can reduce the throughput of these jobs. The **Reuse Idle Processors** setting temporarily releases reserved processors that have not been used in the past 24 hours. This increases the number of processors for lower priority jobs.

When **Reuse Idle Processors** is enabled, it initiates an hourly recurring audit. The system uses the data collected to determine whether to release reserved processors. After reserved processors are released, the audit data is also used to determine whether the system needs to increase reserved processors.

Important: If the system decreases or increases the number of reserved processors, additional decreases are not made for 24 hours. However, additional increases (up to the selected limit) can still be made after each hourly audit. This process continues as long as **Reuse Idle Processors** is enabled.

The system analyzes the following data points during this process:

- **a** = total number of processors available
- **b** = the value you set for **Number of Processors Reserved** (the maximum number of reserved processors; this number does not change)
- **c** = maximum number of jobs concurrently processed in the last 24 hours
- **d** = maximum number of high priority jobs that concurrently waited more than a minute in the last hour
- **e** = current number of reserved processors (this number can change if **Reuse Idle Processors** is enabled)

If	Then
c is less than **a** This means that some reserved processors were not used.	The number of processors available to lower priority jobs is increased by **a – c** (up to the value of **a**).
e is less than **b AND d** is more than **0**	The number of reserved processors is increased by **d** (up to the value of **b**) The system cannot increase the number of reserved processors over the limit set for **Number of Processors Reserved**. In other words, **e** cannot be greater than **b**.

SuiteScript Monitoring, Auditing, and Logging

- SuiteScript Montoring with the Application Performance Management (APM)
- SuiteScript Monitoring with the Script Queue Monitor

SuiteScript Montoring with the Application Performance Management (APM)

The Application Performance Management SuiteApp enables you to observe and manage the performance of your NetSuite customizations and business critical operations. You can use the dashboard, data visualizations, page time summary, script analysis, and script queue monitor to review and improve the speed of the NetSuite user interface.

For additional information on the APM, see the help topic Application Performance Management (APM).

SuiteScript Monitoring with the Script Queue Monitor

Note: This help topic is about the Script Queue Monitor (Beta) SuiteApp (Bundle ID: 56125). For information about the Script Queue Monitor dashboard that is integrated in the Application Performance Management SuiteApp (Bundle ID: 67350), see the help topics Application Performance Management (APM) and Using the Script Queue Monitor Dashboard in APM.

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Note: SuiteCloud Processors is the current system used to execute (process) scheduled scripts and map/reduce scripts. All map/reduce script deployments and all scheduled script deployments created after the 2017.2 release no longer use queues. These deployments are now executed with processors. Scheduled script deployments created prior to 2017.2 continue to use queues until the script owner chooses to use processors. This is done by clicking the **Remove Queues** button on the deployment.

The Script Queue Monitor (Beta) SuiteApp can track only queue activity, and not processor activity. Script deployments that no longer use queues cannot be monitored with this SuiteApp..

To monitor scripts that no longer use queues, use the Script Queue Monitor Dashboard that is included with the Application Performance Management SuiteApp. For more information, see the help topic Using the Script Queue Monitor Dashboard in APM

The Script Queue Monitor (Beta) SuiteApp is intended for use in accounts with one or more SuiteCloud Plus licenses. SuiteCloud Plus allows larger accounts to divide their scheduled script work so that

scripts can be processed more efficiently. Generally, companies that run NetSuite are provided a single queue for running their scheduled scripts. Companies can upgrade their number of scheduled script queues from one to five with the purchase of a SuiteCloud Plus license. The purchase of two SuiteCloud Plus licenses provides 10 queues and the purchase of three licenses provides 15 queues. For more information about SuiteCloud Plus, see the help topics Scheduled Scripts on Accounts with Multiple Processors (SuiteCloud Plus) and SuiteCloud Plus Settings.

The Script Queue Monitor provides charts and a list of status details for scheduled script instances running on multiple queues. Account administrators can use the visualizations provided by the Script Queue Monitor to review and manage script queue usage. This information can inform the retargeting of scheduled script deployments to different queues, to maximize the benefit obtained from SuiteCloud Plus.

> **Note:** The Script Queue Monitor is currently considered a beta feature.
>
> The Script Queue Monitor currently is not supported if you are running NetSuite in Internet Explorer 11.

For more information, see the following:

- Script Queue Monitor Installation and Access
- Script Queue Monitor User Interface

Script Queue Monitor Installation and Access

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Script Queue Monitor (Beta) is available as a SuiteApp which can be installed in your account. Its bundle ID is 56125 and it can be installed from production account 3923787.

To install the Script Queue Monitor (Beta) SuiteApp:

1. Go to Customization > SuiteBundler > Search and Install Bundles.
2. Enter the bundle ID, **56125**, in the **Keywords** box and click the Search button.
3. Click the bundle link in the **Name** column to proceed to the Bundle Details Page.
4. On the Bundle Details page, click the **Install** button.
   - The Script Queue Monitor is a managed bundle. After you install it, future updates are automatically pushed to your account.
   - To proceed to the install, click OK in the popup that displays to obtain your consent for these future updates.
5. On the Preview Bundle Install Page, click the **Install Bundle** button.

After you have started the process of installing a bundle, the Installed Bundles page displays. If the installation is not complete, the **Status** column displays the percentage of installation progress. If the installation is complete, the **Status** column displays a green check.

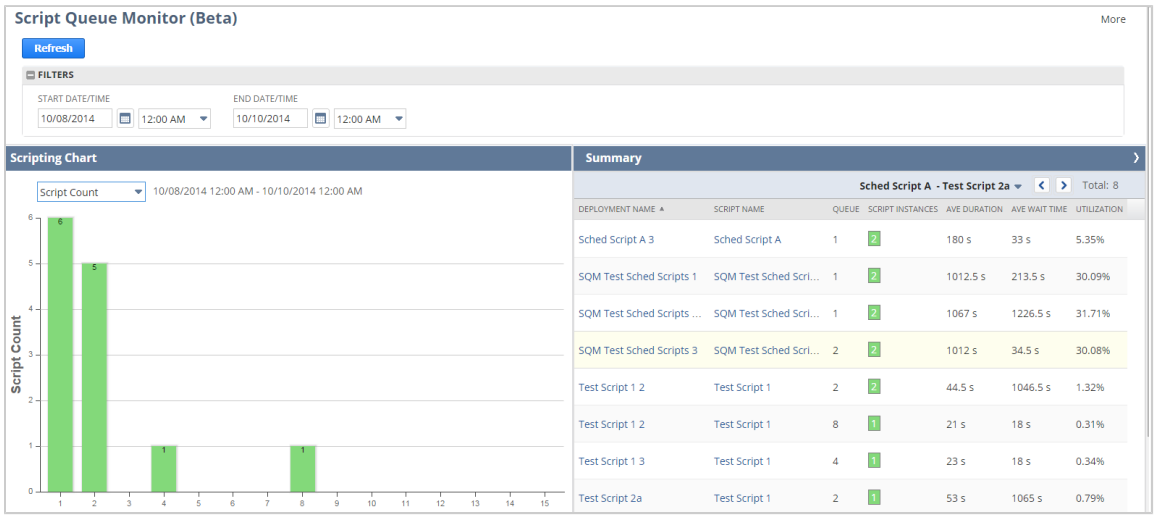
After this SuiteApp has been installed, the Script Queue Monitor is available at Customization > Script Performance > Script Queue Monitor, to account administrators only.

Script Queue Monitor User Interface

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The UI for the Script Queue Monitor includes two panes: on the left, a pane that illustrates script instances for each queue, and on the right, a pane that summarizes relevant information about all script instances, aggregated by deployment name, script name, and queue.

In a Filters section at the top, you can select the time period for scripts to be included in the chart and list. Note that you can collapse and expand the Filters section as needed by clicking on its header bar. In the Scripting Chart pane, you can select the type of chart to display. You click the **Refresh** button to implement these selections in the Script Queue Monitor display.

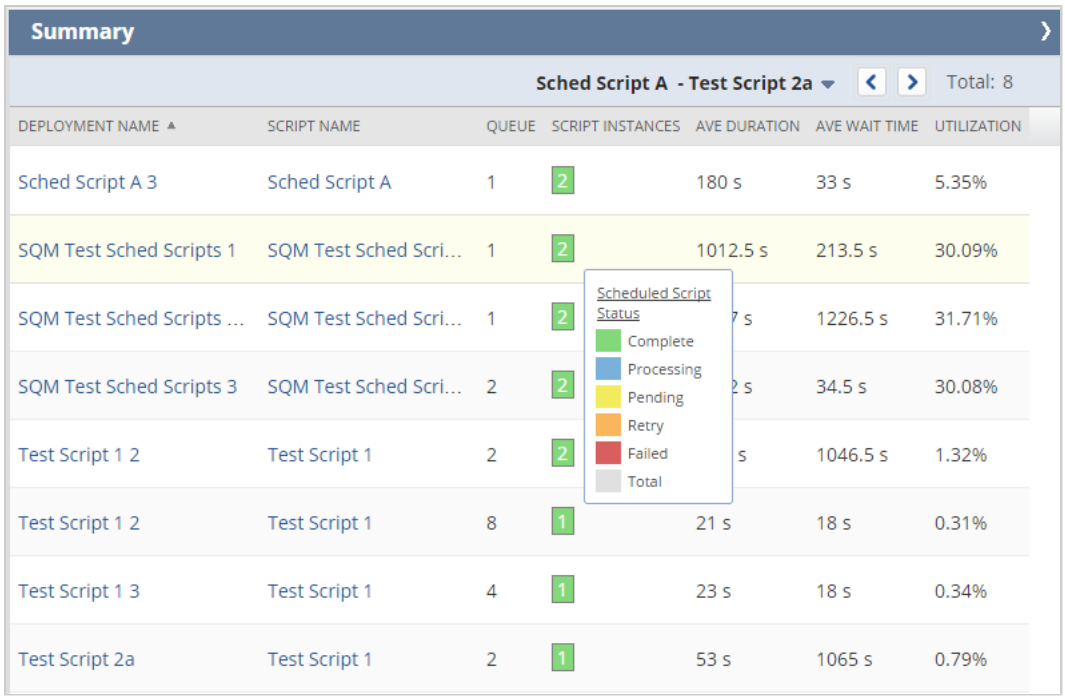


Summary Status Information

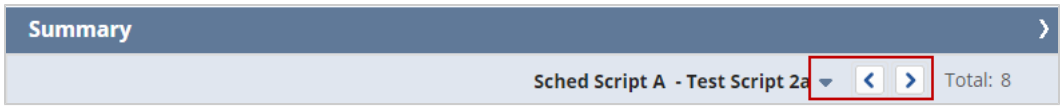
The grid in the right pane of the Script Queue Monitor includes the following details, aggregated by the combination of deployment name, script name, and queue number:

- Instances of that script deployment, on that queue, for the selected time period

  The color of a value in the Script Instances column indicates status, as shown in a key available from a hover over this value.
- Average duration, in seconds, of script execution time
- Average wait time, in seconds, before script execution
- Queue utilization percentage — the percentage of the total duration for all scripts running on that particular date and time

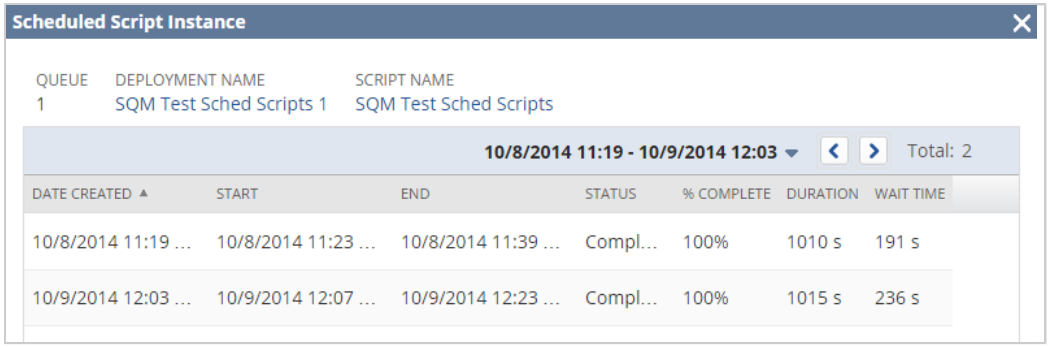


You can click on a column header to sort the summary by that column's values. Click two times to switch the sort from ascending to descending. Summary results are paginated for usability. Use the < and > buttons, or the dropdown arrow, to shift to a different page.



The Summary grid includes links you can click to drill down into more details:

- Click a link in the Deployment Name column to see the script deployment record.
- Click a link in the Script Name column to see the script record.
- Click on a value in the Script Instances column to display a popup window listing details about each script instance:



Click a link in the Deployment Name column to see the script deployment record.

Click a link in the Script Name column to see the script record.

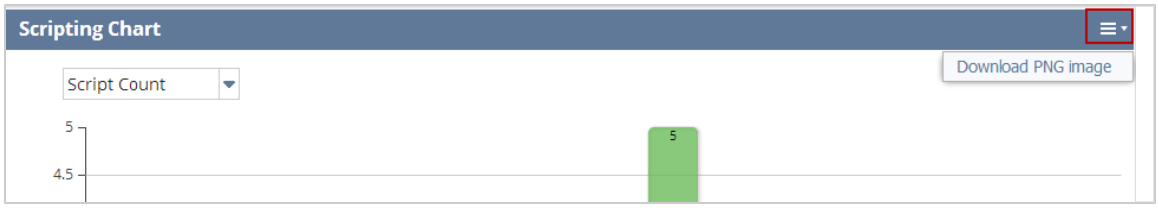
Click on a value in the Script Instances column to display a popup window listing details about each script instance:

Scripting Chart

The Script Queue Monitor supports three types of charts:

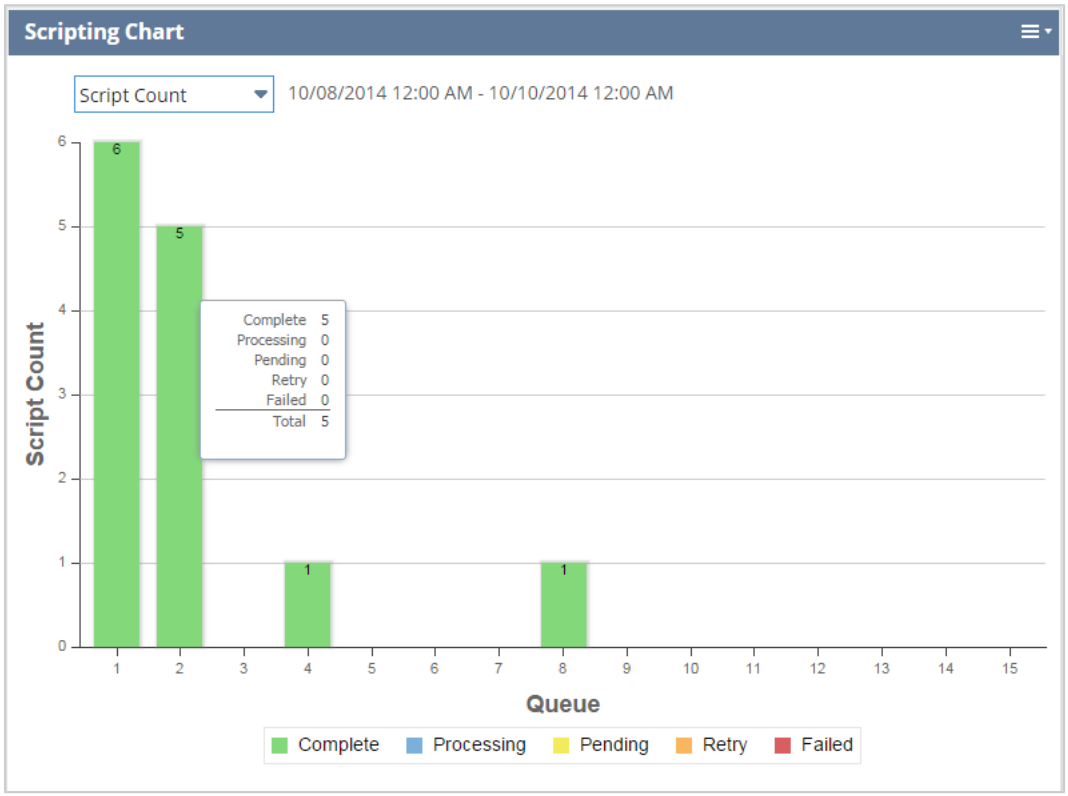
- Script Count Chart
- Utilization Chart
- Timeline Chart

Note that you can click the button available in the chart pane to download the currently displayed chart as a file named chart(<#>).png. (The # variable starts at 1 and is increased each time a chart is downloaded to avoid overwriting previously downloaded charts.)



Script Count Chart

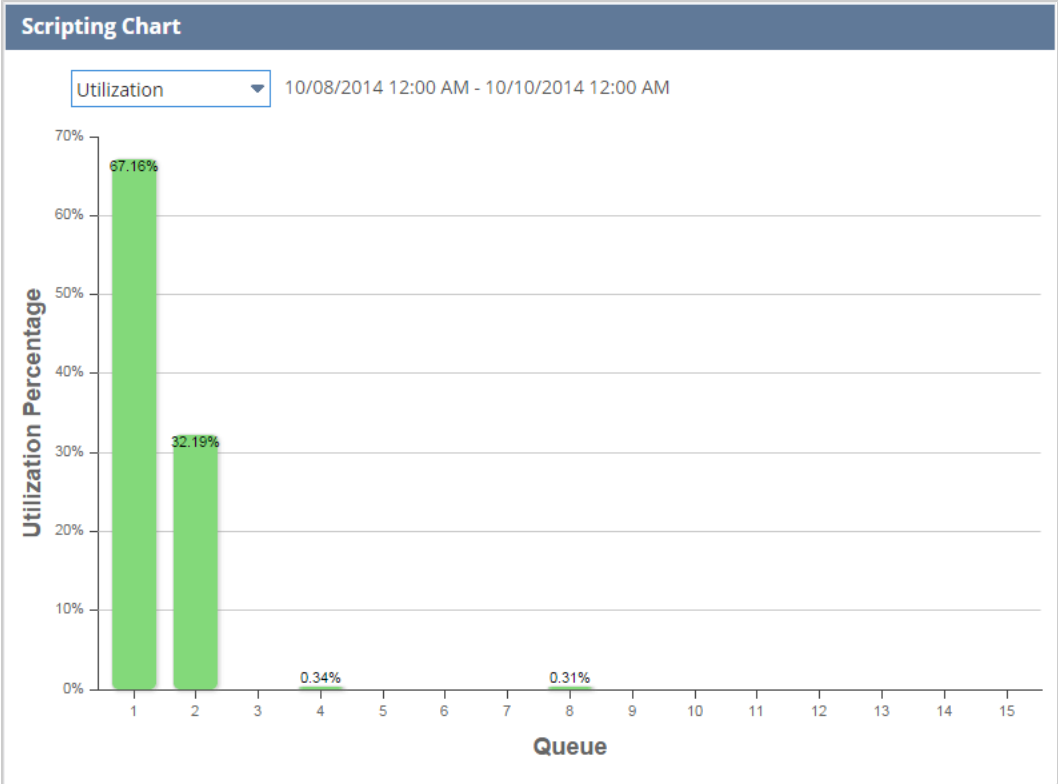
The Script Count chart displays vertical bars representing the number of scripts running on each queue during the selected time period. A hover over the bar for a queue displays a popup listing the number of scripts for each script status.



For each chart, bar color indicates script status, as explained by a legend at the bottom of the pane. You can click on a color in the legend to remove scripts with that status from the chart, and click on the color again to add scripts of that status back in.

Utilization Chart

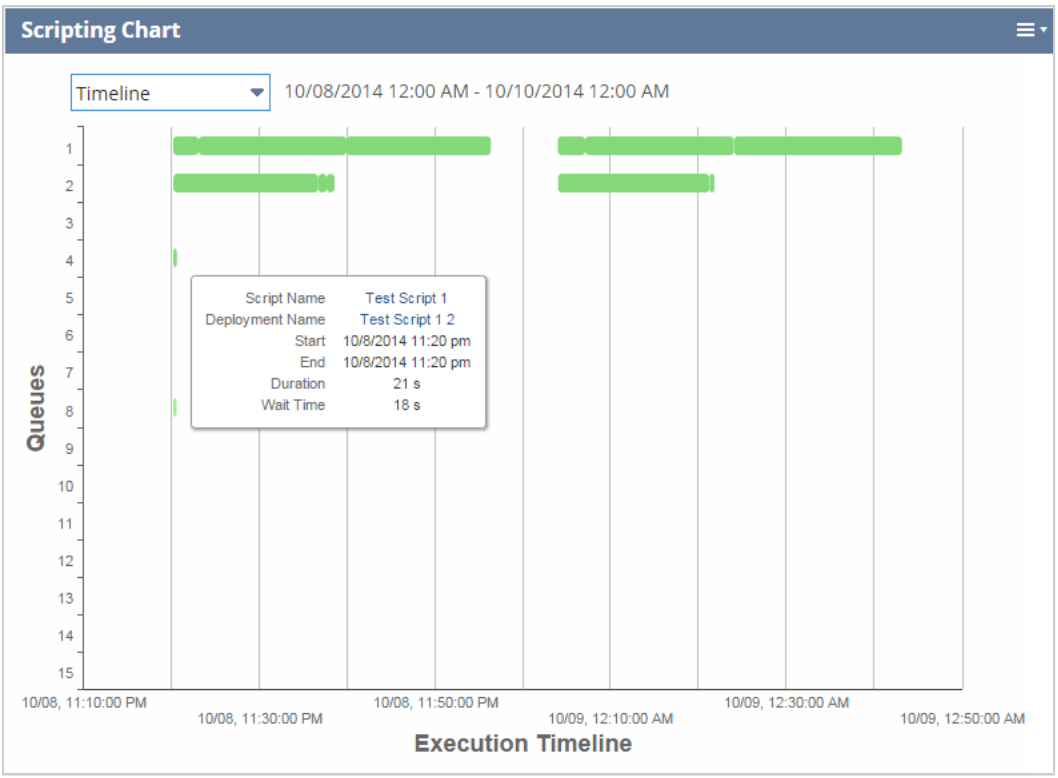
The Utilization chart displays vertical bars representing the percentage of capacity used for each queue during the selected time period. A hover over the bar for a queue displays a popup listing this percentage, the count of scripts run in the queue, and the total runtime of these scripts, in seconds.



Timeline Chart

The Timeline chart displays horizontal bars representing the times during which scheduled scripts were running on each queue during the selected time period. A hover over the bar for a script displays a popup listing this script name, deployment name, start and end times, and total runtime, in seconds.

The Timeline chart displays horizontal bars representing the times during which scheduled scripts were running on each queue during the selected time period. A hover over the bar for a script displays a popup listing this script name, deployment name, start and end times, and total runtime in seconds.



You can grab and highlight a vertical slice of the timeline to zoom in and get a better view of script executions during the highlighted time period. To return to the higher level view, click the **Reset Zoom** button that is available in the lower right corner of the chart after a zoom.

Important: The timeline chart cannot display a time period longer than 3 days. If the selected date range is longer, the timeline chart includes only the first 3 days.

Working with the SuiteScript Records Browser

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The following topics are covered in this section. If you are new to SuiteScript, these topics should be read in order.

- How to Use SuiteScript Records Help
- SuiteScript References Overview
- Using the SuiteScript Records Browser

How to Use SuiteScript Records Help

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The SuiteScript reference guide must be used in conjunction with the SuiteScript Records Browser. Each record listed in this guide includes a link to the Records Browser. Within the Records Browser, you will find all available fields, sublists, and search filter fields for that record. The Records Browser also provides field level help for many fields that appears on the record. Within this guide, the description of each record often includes the following types of information:

- **Supported script types** — a description of whether the record is supported in both client and server SuiteScript, and whether the user events are supported.
- **Supported functions** — a list of all functions that can be used with the record.
- **Field definitions** — provides a link to the Records Browser and, in some cases, additional explanations about how to work with particular fields.
- **Usage notes** — additional details or context about working with the record.
- **Code samples** — examples of how to structure your scripts.

Important: For a list of all NetSuite records that are supported in SuiteTalk, see SuiteScript Supported Records.

Working Online

If you are working online, using either the PDF or online help version of this guide, click the links to the SuiteScript Records Browser that are provided with each record description. The Records Browser will open in your default browser.

Downloading the SuiteScript Records Browser

If you want to use the SuiteScript Records Browser when working offline, you must download the SuiteScript Records Browser .zip file

After downloading the .zip file to your local machine, extract the .zip and navigate to the script directory. To view the Records Browser content, open index.html file in the browser of your choice.

Note: For information on using the Records Browser, see Using the SuiteScript Records Browser.

SuiteScript References Overview

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The NetSuite Help Center provides all the IDs you will need when referencing NetSuite records, fields, sublists, search filters, permissions, features, and so on in SuiteScript.

Important: When writing SuiteScript, you must use the IDs listed in the NetSuite Help Center. Although you can access the IDs by viewing page source on a NetSuite record, there is no guarantee that all IDs in the source code are supported in SuiteScript. If you create a script that references an unsupported or undocumented ID, and the ID is later changed by NetSuite, your script may break.

To find SuiteScript-supported records and IDs:

1. Open the SuiteScript Records Browser. Only records that officially support SuiteScript are listed in the SuiteScript Records Browser. (For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser.)
2. Click the record you want to reference in SuiteScript.

   You will see all IDs currently supported for this record. These IDs can include:

   - field IDs - used when referencing Field APIs
   - sublist IDs - used when referencing Sublist APIs
   - sublist field IDs - used when referencing Sublist APIs
   - search join IDs - used when referencing Search APIs
   - search filter IDs - used when referencing Search APIs
   - search column IDs - used when referencing Search APIs
   - transformation IDs - used when working with nlapiTransformRecord(type, id, transformType, transformValues)

Important: Not every field that appears in the SuiteScript Records Browser is settable through SuiteScript. Some fields are read-only. You will need to look at the NetSuite UI to know whether a field is settable. The general rule is that if you can set a field in the UI, you can set it in SuiteScript. If you cannot set a field in the UI, you cannot set it using SuiteScript. You can, however, still get the field's value using SuiteScript.

In the NetSuite Help Center, also see these links for additional internal ID information. The IDs listed in these sections do not pertain to specific NetSuite records.

Using the SuiteScript Records Browser

Viewing the SuiteScript Records Browser

Go to the SuiteScript Records Browser.

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Important: When writing SuiteScript, you must use the IDs listed in the NetSuite Help Center. Although you can access the IDs by viewing page source on a NetSuite record, there is no guarantee that all IDs in the source code are supported in SuiteScript. If you create a script that references an unsupported or undocumented ID, and the ID is later changed by NetSuite, your script may break.

The SuiteScript Records Browser provides a summary of all records, fields, sublists, search joins, search filters, search columns, and record transformations that are supported in SuiteScript. Information about elements is displayed as a series of tables. For more details, see the following topics:

- Finding a Record or Subrecord
- Understanding the Record Summary
- Deleted Record Search

Note: You can use the Records Browser online, or you can download it. For information about downloading the browser, see Downloading the SuiteScript Records Browser.

Note: For information on governance applied to individual SuiteScript APIs, as well as to various SuiteScript script types, see SuiteScript Governance.

Note: Only the Records Browser for the most recent release of NetSuite is supported. Older version may still be accessible, but they are not supported.

Finding a Record or Subrecord

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

To find a record or subrecord in the SuiteScript Records Browser, use the A-Z index at the top of the browser window.

To find a record or subrecord:

1. Click the appropriate letter at the top of the browser window.

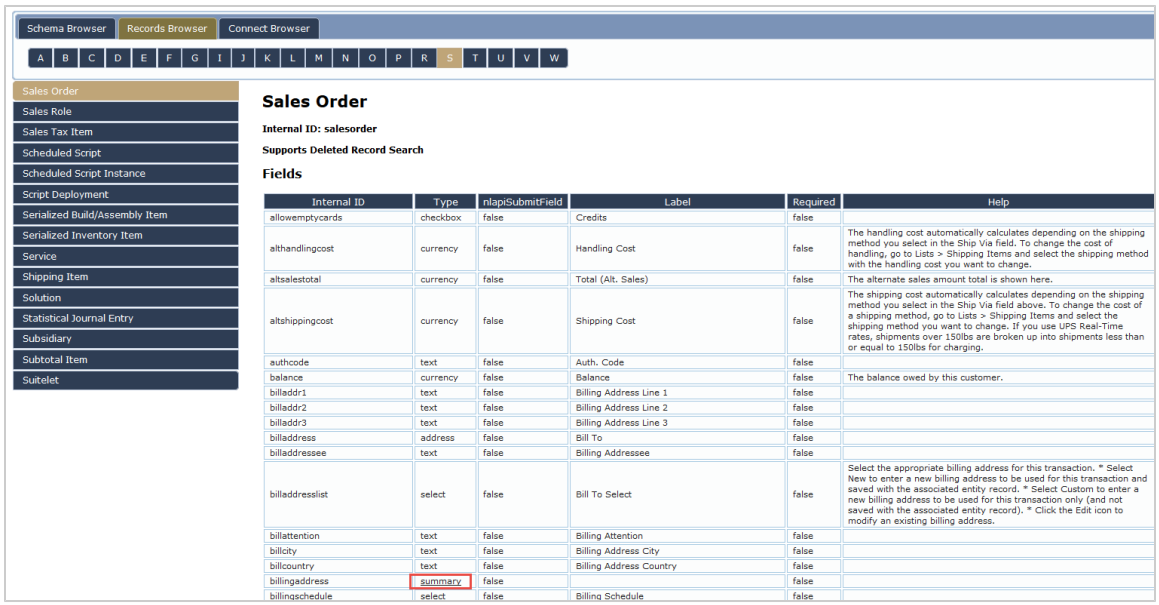
   The pane at the left updates to include a list of all records whose names begin with the letter you selected. The center pane updates to show details of the first record in the list.

2. In the left-hand pane, click the name of the record you are interested in.

   The center pane updates to show details of the record.

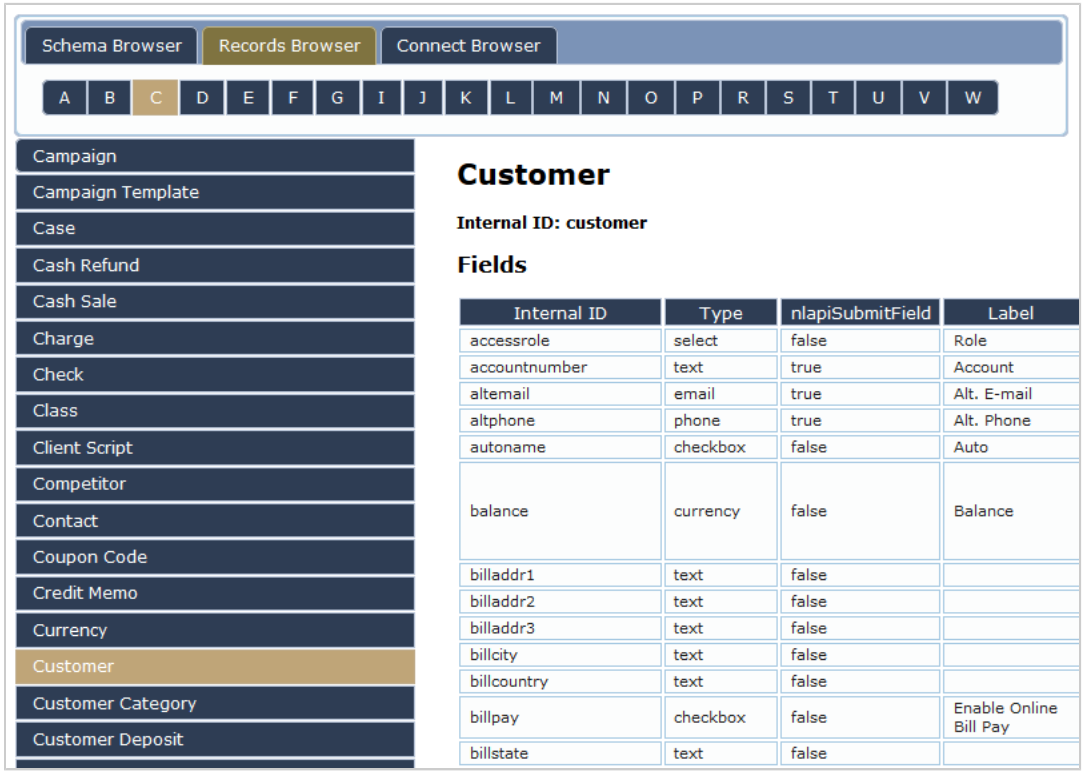
You can also get to a subrecord page from one of its parent records. Each subrecord summary field on a record page now includes a link to the subrecord page. For example, on the Sales Order page of the

Records Browser, the billingaddress and shippingaddress fields include links to the address subrecord page. Click the **summary** link in the **Type** column to go to the subrecord page.



Understanding the Record Summary

In the SuiteScript Records Browser, for each record exposed to SuiteScript, you can find a reference page, such as the one in the following illustration.



Summary of the Record

Browser pages include the following designations for applicable records:

- **Search Only** — appears on pages for records that do not support SuiteScript actions other than search.
- **Supports Deleted Record Search** — appears on pages for records that can be used as filters for the deleted record search.

For each record, the browser displays a series of tables summarizing the following:

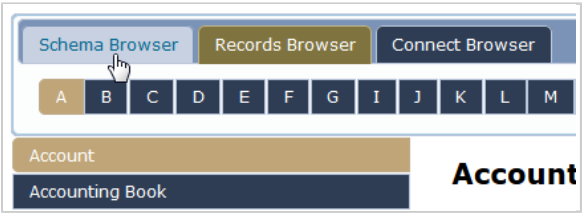
- Fields — The record's fields.
- Sublists — A table representing each supported sublist.
- Tabs — A list of tabs available on the record in the user interface.
- Search Joins — A list of other searches you can access when searching this record.
- Search Filters — Fields in the record that you can use as search criteria.
- Search Columns — Fields that you can include search results.
- Transform Types — A list of records that this record can be transformed into using nlapiTransformRecord.

Some of the column names used by these tables are described below.

Column Label	Description
Name	The internal ID of the field.
Type	The data type of the field.
nlapiSubmitField	A boolean value indicating whether the field supports the nlapiSubmitField function, which enables you to do the equivalent of direct line editing. For details on this function, see the help topic Inline Editing Using nlapiSubmitField.
Label	The label for the field as shown in the user interface.
Help	Additional details about working with the field.

Comparing SuiteScript, SuiteTalk, and SuiteAnalytics Connect Exposure

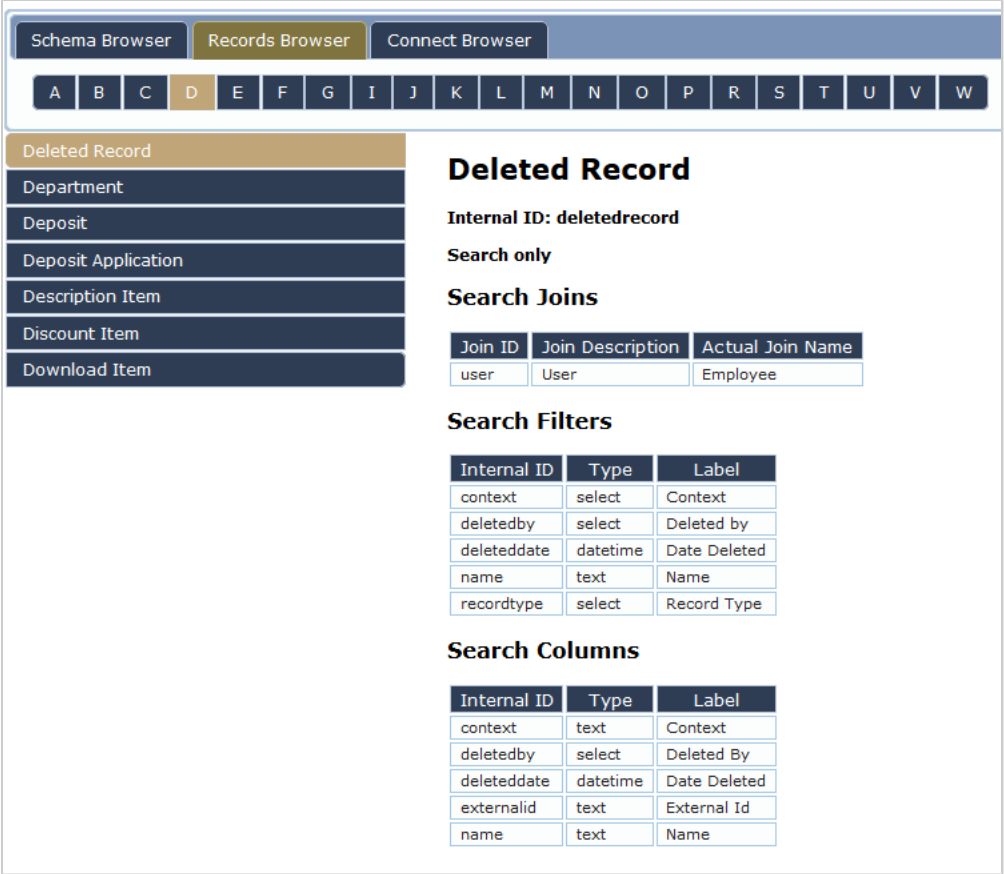
To check whether the record you are currently viewing is also supported in SuiteTalk or SuiteAnalytics Connect, click the Schema Browser tab or the Connect Browser tab at the top of the page.



If the record is supported in SuiteTalk or SuiteAnalytics Connect, you are directed to the corresponding page in the SuiteTalk Schema Browser or SuiteAnalytics Connect Browser. Otherwise, you are directed to the first page of the respective browser. To compare record types support across SuiteScript, SuiteTalk, and SuiteAnalytics Connect, see the help topic SuiteCloud Supported Records.

Deleted Record Search

The Records Browser includes a Deleted Record page, listed under **D**. This page lists the columns and filters available for deleted record searches in SuiteScript. You can determine whether a record can be used as a filter deleted records searches by looking at its page in the Records Browser. The **Supports Deleted Record Search** designation appears on pages for records that can be used as filters.



For information about running deleted record searches in the UI, see the help topic Searching for Deleted Records.

The retrieval of deleted records in web services is a bit different. You can use the getDeleted operation rather than running a search. For a complete list of supported record types for this operation, refer to the DeletedRecordType enumeration in the coreTypes xsd.

SuiteScript Supported Records

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The following table lists all NetSuite records that support SuiteScript. Also provided are record IDs, which are often referenced in SuiteScript APIs.

Note that a scripting level defined as **Full** means that the record can be created, updated, copied, deleted, and searched using SuiteScript.

All subrecords are scriptable from the line item level, unless stated otherwise.

Memorized transactions do not support SuiteScript.

Record Name	Record ID	Record Category	Scripting Level	Scriptable in Client SuiteScript	Scriptable in Server SuiteScript
Account	account	List	Full	Record-level Client SuiteScript only	X
Accounting Book	accountingbook	List	Copy Not Supported		X
Accounting Context	accountingcontext	List	Read, Create, Update, Copy, Delete, and Search	X	X
Accounting Period	accountingperiod	List	Read and Search Only	X	X
Activity	activity	Activity	Search Only	X	X
Address See the help topic Using SuiteScript with Address Subrecords.	addressbookaddress	Subrecord	See the help topic Using SuiteScript with Address Subrecords.	X	Server-side scripts must access through the parent record.
Advanced Intercompany Journal Entry	advintercompanyjournalentry	Transaction	Full	X	X
Allocation Schedule	allocationschedule	List	Create, Read, Update, and Delete	X	X
Amortization Schedule	amortizationschedule	List	Copy, Create, and Delete Not Allowed		X
Amortization Template	amortizationtemplate	List	Full		X
Assembly Build	assemblybuild	Transaction	Full		X
Assembly Item	assemblyitem	Item	Full		X
Assembly Unbuild	assemblyunbuild	Transaction	Full		X
Billing Account	billlingaccount	List	Full	X	X
Billing Class	billingclass	List	Full		X

Record Name	Record ID	Record Category	Scripting Level	Scriptable in Client SuiteScript	Scriptable in Server SuiteScript
Billing Rate Card	billingratecard	List	Read, Create, Update, Delete, and Search Copy and Transform Not Supported	X	X
Billing Schedule	billingschedule	List	Full		X
Bin	bin	List	Full		X
Bin Putaway Worksheet	binworksheet	Transaction	Copy and Update Not Supported		Dynamic Mode Only
Bin Transfer	bintransfer	Transaction	Full		X
Blanket Purchase Order	blanketpurchaseorder	Transaction	Full	X	X
BOM	bom	List	Read, Create, Update, Copy, Delete, and Search	X	X
BOM Revision	bomrevision	List	Read, Create, Update, Copy, Delete, and Search		X
Campaign	campaign	Marketing	Full	X	X
Campaign Response	campaignresponse	Marketing	Full	X	X
Campaign Template	campaigntemplate	Marketing	Search Not Supported		X
Case	supportcase	Support	Full	X	X
Cash Refund	cashrefund	Transaction	Full	X	X
Cash Sale	cashsale	Transaction	Full	X	X
Change Order	subscriptionchangeorder	Transaction	Full	X	X
Charge	charge	Transaction	Full		X
Check	check	Transaction	Full	X	X
Class	classification	List	Full	X	X
CMS Content	cmscontent	Web Site	Create, Update, and Search Only Attach, Transform, and Copy Not Supported		X
CMS Content Type	cmscontenttype	Web Site	Create, Update, and Search Only Attach, Transform, and Copy Not Supported		X
CMS Page	cmspage	Web Site	Create, Update, and Search Only Attach, Transform, and Copy Not Supported		X

Record Name	Record ID	Record Category	Scripting Level	Scriptable in Client SuiteScript	Scriptable in Server SuiteScript
Commerce Category	commercecategory	Website	Attach, Transform, and Copy Not Supported		X
Competitor	competitor	Entity	Full	X	X
Consolidated Exchange Rate	consolidatedexchangerate	List	Read, Update, and Search Only Create and Delete Not Supported	X	X
Contact	contact	Entity	Full	X	X
Contact Category	contactcategory	List	Full	X	X
Contact Role	contactrole	List	Full	X	X
Cost Category	costcategory	List	Full	X	X
Coupon Code	couponcode	Marketing	Full	X	X
Credit Card Charge	creditcardcharge	Transaction	Full		X
Credit Card Refund	creditcardrefund	Transaction	Full		X
Credit Memo	creditmemo	Transaction	Full	X	X
Currency	currency	List	Full — with Multiple Currencies Feature Enabled Read Only — without Multiple Currencies Feature		X
Custom List	See Custom List IDs	Custom	Search Only	X	X
Custom Record	See Custom Record IDs	Custom	Full	X	X
Customer	customer	Entity	Full	X	X
Customer Category	customercategory	List	Search Not Supported	X	X
Customer Deposit	customerdeposit	Transaction	Full	X	X
Customer Message	customermessage	List	Full	X	X
Customer Payment	customerpayment	Transaction	Copy and Create Not Allowed		X
Customer Refund	customerrefund	Transaction	Full		X
Customer Status	customerstatus	Entity	Full	X	X
Customer-Subsidiary Relationship	customersubsidiaryrelationship	List	Full	X	X
Department	department	List	Full	X	X
Deposit	deposit	Transaction	Copy Not Supported	X	X

Record Name	Record ID	Record Category	Scripting Level	Scriptable in Client SuiteScript	Scriptable in Server SuiteScript
			Dynamic Mode Not Supported		
Deposit Application	depositapplication	Transaction	Create Not Allowed	X	X
Description	descriptionitem	Item	Full	X	X
Discount	discountitem	Item	Full	X	X
Download Item	downloaditem	Item	Full	X	X
Driver's License	driverslicense	List	Full	X	X User event scripts are not supported.
Email Template	emailtemplate	Marketing	Search Not Supported		X
Employee	employee	Entity	Full	X	X
Entity	entity	Entity	Search Only		
Estimate / Quote	estimate	Transaction	Full	X	X
Event	calendarevent	Activity	Full	X	X
Expense Category	expensecategory	List	Full		X
Expense Report	expensereport	Transaction	Full	X	X
Fair Value Price	fairvaluepricelist	List	Create, Delete, and Search Only Copy Not Allowed		X
File	file	File Cabinet	Search Only		X
Folder	folder	File Cabinet	Full		X
Fulfillment Request	fulfillmentrequest	Transaction	Create and Copy Not Supported	X	X
Generic Resource	genericresource	Entity	Copy Not Supported	X	X
Gift Certificate	giftcertificate	List	Full	X	X
Gift Certificate Item	giftcertificateitem	Item	Full	X	X
Global Account Mapping	globalaccountmapping	List	Full		X
Group	entitygroup	List	Full		X
Inbound Shipment	inboundshipment	List	Copy and Transform Not Supported	X	X
Intercompany Allocation Schedule	intercompallocationschedule	List	Create, Read, Update, and Delete	X	X
Intercompany Journal Entry	intercompanyjournalentry	Transaction	Full	X	X

Record Name	Record ID	Record Category	Scripting Level	Scriptable in Client SuiteScript	Scriptable in Server SuiteScript
Intercompany Transfer Order	intercompanytransferorder	Transaction	Full	X	X
Inventory Adjustment	inventoryadjustment	Transaction	Full	X	X
Inventory Cost Revaluation	inventorycostrevaluation	Transaction	Full	X	X
Inventory Count	inventorycount	Transaction	Full	X	X
Inventory Detail	inventorydetail	Subrecord	See the help topic Scripting the Inventory Detail Subrecord.	X	Server-side scripts must access through the parent record
Inventory Item (Also referred to in the UI as Inventory Part)	inventoryitem	Item	Full	X	X
Inventory Number	inventorynumber	List	Copy, Create, and Delete Not Supported		X
Inventory Status	inventorystatus	List	Full	X	X
Inventory Status Change	inventorystatuschange	Transaction	Full	X	X
Inventory Transfer	inventorytransfer	Transaction	Full		X
Invoice	invoice	Transaction	Full	X	X
Issue	issue	Support	Full	X	X
Issue Product	issueproduct	Support	Copy, transform and search not supported.	X	X
Issue Product Version	productversion	Support	Copy, transform and search not supported.	X	X
Item Account Mapping	itemaccountmapping	List	Full		X
Item Demand Plan	itemdemandplan	Transaction	Copy Not Supported, No Available Transforms	X	X
Item Fulfillment	itemfulfillment	Transaction	Copy and Create Not Supported	X	X
Item Group	itemgroup	Item	Full	X	X
Item Location Configuration	itemlocationconfiguration	List	Read, Create, Edit, Delete, and Search	X	X
Item Receipt	itemreceipt	Transaction	Copy and Create Not Supported	X	
Item Revision	itemrevision	List	Full		X
Item Search	item	Item	Search Only		

Record Name	Record ID	Record Category	Scripting Level	Scriptable in Client SuiteScript	Scriptable in Server SuiteScript
Item Supply Plan	itemsupplyplan	Transaction	Full	X	X
Job	job	Entity	Copy and Transform Not Supported	X	X
Job Requisition	jobrequisition	Entity	Copy and Transform Not Supported	X	X
Journal Entry	journalentry	Transaction	Full	X	X
Kit	kititem	Item	Full		X
Kudos Feature Records	kudos	List	Full Only the inactive field can be edited.		X
Landed Cost	landedcost	Subrecord	Full		X
Lead	lead	Entity	Full	X	X
Location	location	List	Full		X
Lot Numbered Assembly Item	lotnumberedassemblyitem	Item	Full	X	X
Lot Numbered Inventory Item	lotnumberedinventoryitem	Item	Full	X	X
Manufacturing Cost Template	manufacturingcosttemplate	List	Full		X
Manufacturing Planned Time	mfgplannedtime	Transaction	Search Only	X	X
Manufacturing Operation Task	manufacturingoperationtask	Transaction	Copy Not Supported	X	X
Manufacturing Routing	manufacturingrouting	List	Full		X
Markup	markupitem	Item	Full	X	X
Message	message	Communication	For details, see Message.		X
Multi-Book Accounting Transaction	accountingtransaction	Transaction	Search Only	X	X
Nexus	nexus	List	Copy Not Supported		X
Non-Inventory Part	noninventoryitem	Item	Full	X	X
Note	note	Communication	Full		X
Note Type	notetype	List	Full	X	X
Opportunity	opportunity	Transaction	Full	X	X
Order Schedule	orderschedule	Subrecord	See the help topic Working with Subrecords in SuiteScript.	X	Server-side scripts must access through the parent record

Record Name	Record ID	Record Category	Scripting Level	Scriptable in Client SuiteScript	Scriptable in Server SuiteScript
Organization Value	organizationvlaue	List	Full		X
Other Charge Item	otherchargeitem	Item	Full		
Other Government-Issued ID	othergovernmentissuedid	List	Full	X	X User event scripts are not supported.
Other Name	othername	Entity	Full		X
Other Name Category	othernamecategory	List	Full	X	X
Partner	partner	Entity	Full	X	X
Partner Category	partnercategory	List	Full	X	X
Passport	passport	List	Full	X	X User event scripts are not supported.
Paycheck	paycheck	Transactions	Read and Update Only	X	X
Paycheck Journal	paycheckjournal	Transaction	Full	X	X
Payment	paymentitem	Item	Full	X	X
Payment Method	paymentmethod	List	Full	X	X
Payroll Batch	payroll batch	Transactions	Full	X	X
Payroll Batch Employee	payrollbatchaddemployees	Transaction	Read and Update Only	X	X
Payroll Item	payrollitem	List	Full	X	X
Phone Call	phonecall	Activity	Full	X	X
Price Book	pricebook	List	Full	X	X
Price Level	pricelevel	List	Search Not Supported	X	X
Price Plan	priceplan	List	Full	X	X
Pricing Group	pricinggroup	List	Full	X	X
Project (Job)	job	Entity	Full	X	X
Project Charge Rule	chargerule	List	Transform Not Supported	X	X
Project Expense Type	projectexpensetype	List	Full		X
Project Revenue Rule	laborbasedprojectrevenuerule, fixedamountprojectrevenuerule, pctcompleteprojectrevenuerule	List	Copy and Transform Not Supported	X	X

Record Name	Record ID	Record Category	Scripting Level	Scriptable in Client SuiteScript	Scriptable in Server SuiteScript
Project Status	jobstatus	Entity	Full	X	X
Project Task	projecttask	Event	Full	X	X
Project Template	projecttemplate	Entity	Copy Not Supported	X	X
Project Type	jobtype	Entity	Full	X	X
Promotion	promotioncode	Marketing	Full	X	X
Prospect	prospect	Entity	Full	X	X
Purchase Contract	purchasecontract	Transaction	Full	X	X
Purchase Order	purchaseorder	Transaction	Full	X	X
Reallocate Items	reallocateItem	Item	Exposed for user event scripts only. This record can not be created, loaded, or deleted using scripts.		
Requisition	purchaserequisition	Transaction	Full	X	X
Resource Allocation	resourceallocation	Activities	Full	X	X
Return Authorization	returnauthorization	Transaction	Full	X	X
Revenue Arrangement	revenuearrangement	Transaction	Full		X
Revenue Commitment	revenuecommitment	Transaction	Create Not Supported See Usage Notes.	X	X
Revenue Commitment Reversal	revenuecommitmentreversal	Transaction	Create Not Supported See Usage Notes.	X	X
Revenue Recognition Event	billingrevenueevent	List	Create, Red, Update, Delete, and Search	X	X User Event Scripts Not Supported
Revenue Recognition Plan	revrecplan	List	Copy, Create, and Delete Not Supported		X User Event Scripts Not Supported
Revenue Recognition Schedule	revrecschedule	List	Copy, Create, and Delete Not Supported		X (User Event Scripts Not Supported)
Revenue Recognition Template	revrectemplate	List	Full		X (User Event Scripts Not Supported)
Role	role	List	Search Only		X
Sales Order	salesorder	Transaction	Full	X	X

Record Name	Record ID	Record Category	Scripting Level	Scriptable in Client SuiteScript	Scriptable in Server SuiteScript
Sales Role	salesrole	List	Full	X	X
Sales Tax Item	salestaxitem	List	Full		X
Scheduled Script Instance		Customization	Search Only		X
Script	Need Internal ID. See Script description	Customization	Copy, Create, and Delete Not Supported See Usage Notes		X
Script Deployment	Need Internal ID. See Script Deployment description	Customization	See Usage Notes		X
Serialized Assembly Item	serializedassemblyitem	Item	Full	X	X
Serialized Inventory Item	serializedinventoryitem	Item	Full	X	X
Service	serviceitem	Item	Full	X	X
Shipping Item	shipitem	Item	Copy and Create Not Supported		X
Shopping Cart	shoppingcart	Website	Search Only		X
Solution	solution	Support	Full	X	X
Statistical Journal Entry	statisticaljournalentry	Transaction	Full	X	X
Store Pickup Fulfillment	storepickupfulfillment	Transaction	Copy and Create Not Supported	X	X
Subscription	subscription	Transaction	Full	X	X
Subscription Line	subscriptionline	Transaction	Create and Delete Not Supported	X	X
Subscription Plan	subscriptionplan	Item	Full	X	X
Subsidiary	subsidiary	List	Full	X	X
Subtotal	subtotalitem	Item	Full	X	X
System Note	systemnote (SYSTEM_NOTE for SuiteScript 2.0 search enum)	List	Search Only	X	X
Task	task	Activity	Full	X	X
Tax Control Account	taxacct	List	Delete and Search Not Supported	X	X
Tax Group	taxgroup	List	Full		X
Tax Period	taxperiod	List	Full		X
Tax Type	taxtype	List	Full		X
Term	term	List	Search Not Supported	X	X
Time	timebill	Transaction	Full	X	X
Time-Off Change	timeoffchange	Entity	Copy and Transform Not Supported		X User Event Triggering

Record Name	Record ID	Record Category	Scripting Level	Scriptable in Client SuiteScript	Scriptable in Server SuiteScript
					not supported.
Time-Off Plan	timeoffplan	Entity	Copy and Transform Not Supported		X User Event Triggering not supported.
Time-Off Request	timeoffrequest	Entity	Copy and Transform Not Supported		X User Event Triggering not supported.
Time-Off Rule	timeoffrule	Entity	Copy and Transform Not Supported		X User Event Triggering is not supported.
Time-Off Type	timeofftype	Entity	Copy and Transform Not Supported		X User Event Triggering not supported.
Topic	topic	Support	Full		X
Transaction Search	transaction	Transaction	Search Only	X	X
Transfer Order	transferorder	Transaction	Full	X	X
Unit of Measure	unitstype	List	Full		X
Usage	usage	Transaction	Full		
Vendor	vendor	Entity	Full	X	X
Vendor Bill	vendorbill	Transaction	Full	X	X
Vendor Category	vendorcategory	List	Search Not Supported	X	X
Vendor Credit	vendorcredit	Transaction	Full	X	X
Vendor Payment	vendorpayment	Transaction	Full	X	X
Vendor Return Authorization	vendorreturnauthorization	Transaction	Full	X	X
Vendor-Subsidiary Relationship	vendorsubsidiaryrelationship	List	Full	X	X
Website Setup	website	Website	Full		X Not supported in beforeLoad user event scripts.

Record Name	Record ID	Record Category	Scripting Level	Scriptable in Client SuiteScript	Scriptable in Server SuiteScript
Work Order	workorder	Transaction	Full	X	X
Work Order Close	workorderclose	Transaction	Copy and Create Not Supported.		X
Work Order Completion	workordercompletion	Transaction	Copy and Create Not Supported		X
Work Order Issue	workorderissue	Transaction	Copy and Create Not Supported		X
Workplace	workplace	List	Full	X	X

Activities

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The following activity records are scriptable in SuiteScript:

- Activity
- Event
- Phone Call
- Project Task
- Resource Allocation
- Task
- Work Calendar

Activity

Note: The content in this help topic pertains to all versions of SuiteScript.

The internal ID for this record is `activity`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The activity record is scriptable in both server and client SuiteScript.

Supported Functions

The activity record can only be searched using SuiteScript.

Event

Note: The content in this help topic pertains to all versions of SuiteScript.

Events are scheduled activities that are automatically added to your calendar when created.

The internal ID for this record is `calendarevent`. Note that setting recurring events in SuiteScript is not currently supported.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The event record is scriptable in both server and client SuiteScript.

Supported Functions

The event record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Usage Notes

When the Time Tracking feature is enabled, the TimeItemList sublist is available. This list is used to track employee time associated with the phone call, including payroll, billing, and project fields. It is an inline editor sublist.

Important: If any case, task, or event record has more than 9500 time entries in the Time Tracking sublist, all cases include a static list of time entries. These static lists are not accessible to scripting. Attempts to script on Time Tracking sublists that are static lists of time entries will not return accurate results.

Phone Call

Note: The content in this help topic pertains to all versions of SuiteScript.

Phone calls records are used to document phone call activity. All information submitted for a phone call record is stored on a record in the phone call list, on the customer record who calls and on any contact's records referenced in the call contact list.

The internal ID for this record is `phonecall`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The phone call record is scriptable in both server and client SuiteScript.

Supported Functions

The phone call record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Usage Notes

When the Time Tracking feature is enabled, the TimeItemList sublist is available. This list is used to track employee time associated with the phone call, including payroll, billing, and project fields. It is an inline editor sublist.

> **Important:** If any case, task, or event record has more than 9500 time entries in the Time Tracking sublist, all cases include a static list of time entries. These static lists are not accessible to scripting. Attempts to script on Time Tracking sublists that are static lists of time entries will not return accurate results.

Project Task

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The project task record can be used to keep track of specific activities and milestones associated with a project.

The internal ID for this record is `projecttask`.

The project task record is available when the Project Management feature is enabled at Setup > Company > Enable Features, on the Company subtab. When the feature is enabled, you can access the project task record in the UI by navigating to an existing project and clicking the New Project Task or New Milestone button. For details on working with this record in the UI, see the help topic Project Tasks.

Project task records cannot be created as standalone records. Rather, you create a project task for a specific project record, and the task remains attached to that record. For information on working with the project record in SuiteScript, see Project (Job).

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Project Tasks Versus Milestone Tasks

Every project task record is designated as either a project task or a milestone task. A project task is used to represent an activity, whereas a milestone task is used to represent a checkpoint in the overall progress of the project.

Although they are the same type of record, a milestone task cannot have values in the estimatedwork body or sublist fields. Therefore, if you create a project task record and do not include any estimated work, the record is automatically saved as a milestone task. If you do include estimated work, the record is saved as a project task.

These same rules apply to the updating of records as well. In other words:

- To convert a project task into a milestone task, clear the estimatedwork body field and all values from the Assignees sublist.
- To convert a milestone task into a project task, add a value to the estimatedwork body field or add at least one record to the Assignees sublist, with a positive value of estimated work.

Note that the estimatedwork body field is populated by the sum of estimated work listed for Assignees. If you explicitly set a value for the estimatedwork body field, and you also include Assignees, the value you specify for the body field is overwritten based on the sublist's estimatedwork values. If you do not include Assignees, you can explicitly assign a value to the estimatedwork body field.

Supported Script Types

The project record is scriptable in server SuiteScript only.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

The project record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Field Definitions

When creating new project tasks you must set the Project (*company*) field to the project/job ID. Project tasks are not standalone records, and therefore must be associated with a specific project.

For other details on body fields and sublist fields, See the SuiteScript Records Browser, which lists all internal IDs associated with this record. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Code Sample

The following example shows how you might create both a project task and a milestone task.

```
// create a project
var project = nlapiCreateRecord('job');
project.setFieldValue('companyname', 'Reconstruction');
var projectId = nlapiSubmitRecord(project);

// create a project task
var task = nlapiCreateRecord('projecttask');
task.setFieldValue('estimatedwork', 2);
task.setFieldValue('title', 'Remove old furniture');
task.setFieldValue('company', projectId);
var task1Id = nlapiSubmitRecord(task);

// create another task depending on the first one with Finish-To-Start dependency
task = nlapiCreateRecord('projecttask');
task.setFieldValue('estimatedwork', 5);
task.setFieldValue('title', 'Paint walls');
task.setFieldValue('company', projectId);
task.selectNewLineItem('predecessor');
task.setCurrentLineItemValue('predecessor', 'task', task1Id);
task.setCurrentLineItemValue('predecessor', 'type', 'FS');
task.commitLineItem('predecessor');
var task2Id = nlapiSubmitRecord(task);

// create a milestone
task = nlapiCreateRecord('projecttask');
task.setFieldValue('estimatedwork', 0);
task.setFieldValue('title', 'Verify painting after the walls dry out');
task.setFieldValue('company', projectId);
task.selectNewLineItem('predecessor');
task.setCurrentLineItemValue('predecessor', 'task', task2Id);
task.setCurrentLineItemValue('predecessor', 'type', 'FS');
task.commitLineItem('predecessor');
var milestoneId = nlapiSubmitRecord(task);
```

Resource Allocation

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Important: For information on the availability of the Resource Allocations feature, please contact your account representative.

The resource allocation record supports reserving an employee's time for a particular project.

The internal ID for this record is `resourceallocation`.

In the UI, you access this record by going to Activities > Scheduling > Resource Allocations. Alternatively, you can view the resource allocations for a specific project through the project record's Resources subtab (Lists > Relationships > Projects, or Lists > Relationships > Jobs).

This record is available only if the Resource Allocations feature has been enabled at Setup > Company > Enable Features, on the Company tab. Note that this option will not be visible unless your account has been provisioned for this feature. For more details, contact your account representative.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

This record is scriptable in both client SuiteScript and server SuiteScript

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

The resource allocation record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Field Definitions

To create a new resource allocation record, you must reference two existing NetSuite records, as follows:

- project — A reference to a project record defined at Lists > Relationships > Projects. Note that in some NetSuite accounts, projects are referred to as jobs.
- allocationresource — A reference to an employee record, defined at Lists > Employees > Employees, for which the Project Record option has been checked. The Project Resource box is located on the Human Resources tab of the employee record.

You are also required to provide values for several other fields. Refer to the SuiteScript Records Browser for the internal IDs of all fields associated with this record.

Note also that numberhours and percentoffime are read-only fields. They are returned when you load the record, but they cannot be modified.

For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser.

Code Samples

The following example shows how to create a resource allocation record:

```
var AMOUNT = "10.0";
var PROJECT1 = "43";
var RESOURCE1 = "45";
var ALLOCATIONTYPE1 = "1";   // Hard
var ALLOCATIONTYPE2 = "2";   // Soft
```

```
var ALLOCATIONUNIT1 = "H";   // Hours
var ALLOCATIONUNIT2 = "P";   // Percent of Time
var NOTES1 = "My notes 1";
var STARTDATE1 = "4/20/2013";
var ENDDATE1 = "4/28/2013";

var record = nlapiCreateRecord('resourceallocation');
record.setFieldValue('allocationamount', AMOUNT);
record.setFieldValue('allocationresource', RESOURCE1);
record.setFieldValue('allocationtype',ALLOCATIONTYPE1 );
record.setFieldValue('allocationunit', ALLOCATIONUNIT1);
record.setFieldValue('startdate', STARTDATE1);
record.setFieldValue('notes', NOTES1);
record.setFieldValue('project', PROJECT1);
record.setFieldValue('enddate', ENDDATE1);
var recId = nlapiSubmitRecord(record);
```

Task

Note: The content in this help topic pertains to all versions of SuiteScript.

Tasks are activities that need to be completed. Use the task record to add new tasks for individuals, companies or contacts and to modify those records.

The internal ID for this record is `task`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The task record scriptable in both client SuiteScript and server SuiteScript.

Supported Functions

The resource allocation record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

When the Time Tracking feature is enabled, the TimeItemList sublist is available. This list is used to track employee time associated with the task, including payroll, billing, and project fields. It is an inline editor sublist.

Important: If any case, task, or event record has more than 9500 time entries in the Time Tracking sublist, all cases include a static list of time entries. These static lists are not accessible to scripting. Attempts to script on Time Tracking sublists that are static lists of time entries will not return accurate results.

Work Calendar

Note: The content in this help topic pertains to all versions of SuiteScript.

You can set up work calendars to track and manage the work capacity for employees and vendors you assign as resources on projects. Knowing the work capacity for each employee helps you to schedule resources for project tasks.

Also, any employee or vendor you plan to assign as a project resource must have a work calendar assigned on their record.

For information about working with work calendar records in the UI, see the following topics:

- Project Resource Work Calendars
- Setting Up a Work Calendar
- Assigning a Resource Work Calendar

The internal ID for this record is `workcalendar`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Entities

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Entity Record Restrictions for Scripting on Administrator Role

This section describes entity record restrictions for scripting on the Administrator and Full Access roles. Note that these restrictions apply to all versions of SuiteScript and all NetSuite environments (see the help topic Understanding NetSuite Account Types).

You cannot perform these actions with SuiteScript:

- Create an entity record where any role is set to Administrator or Full Access

- Delete an entity record where any role is set to Administrator or Full Access
- Edit an entity record so that the entity gains an Administrator or Full Access role
- Edit an entity record so that the entity loses an Administrator or Full Access role
- Edit the password or email field value on an entity record where any role is set to Administrator or Full Access

These restrictions contribute to improved security. Scripts that violate them throw an error message: `Script Security Violation: Unauthorized attempt to <operation> entity with <role> role by SuiteScript!`

Entity Record Types

The following entity records are scriptable in SuiteScript:

- Competitor
- Contact
- Customer
- Customer Status
- Employee
- Entity
- Generic Resource
- Job
- Job Requisition
- Lead
- Other Name
- Partner
- Project (Job)
- Project Status
- Project Template
- Project Type
- Prospect
- Time-Off Management
  - Time-Off Change
  - Time-Off Plan
  - Time-Off Request
  - Time-Off Rule
  - Time-Off Type
- Vendor

Competitor

Note: The content in this help topic pertains to all versions of SuiteScript.

You can create competitor records to track how other businesses in your industry impact your sales.

If you use NetSuite or NetCRM and have the Opportunities feature enabled, selecting a competitor on an opportunity record lets everyone working with that opportunity know who they are competing with. By being aware of the strengths and weaknesses of the competition, sales reps can make educated offers and close more deals.

For details on working with the competitor record in the user interface, see the help topics Competitors and Opportunity Records.

The internal ID for this record is `competitor`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The competitor record is scriptable in both client and server SuiteScript.

Supported Functions

The competitor record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Contact

Contacts represent people or companies that you deal with in the daily activity of your business. Use the contact record to create, modify, or delete contacts and associate a contact to a parent record.

> **Note:** The content in this help topic pertains to all versions of SuiteScript.

The internal ID for this record is `contact`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The contact record is scriptable in both client and server SuiteScript.

Supported Functions

The contact record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Customer

Customer records allow you to track all the following types of information on your customers in one place, depending on the features you have enabled.

For details about working with the customer record in the user interface, see the help topic Customers.

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The internal ID for this record is `customer`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The customer record is scriptable in both client and server SuiteScript.

Supported Functions

The customer record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Usage Notes

Notes on Scripting Customer Fields

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
Body Fields		
datecreated	Date Created	This is a system-generated field that marks the date the record was created in NetSuite. You cannot change or override this field. Tip: If you need to capture “date created” information that is not related to the date the record was created in NetSuite, create a custom field and set it to auto-default to today's date.

Field Internal ID	Field UI Label	Note
password	Password	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).
password2	Confirm Password	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).
Search Filters and Search Columns		
ccnumber	Credit Card Number	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).

The Multi-Partner Management feature must be enabled in your account for the Partners sublist to appear.

The Access / contactroles sublist is included in the customer record on the Access subtab. It is an inline editor subtab. The Access / contract roles sublist is related to the Contact / contactroles sublist. For details, see the contactroles sublist in the SuiteScript Records Browser.

The Time Tracking sublist is included with the customer record. It is an inline editor sublist.

Important: If any case, task, or event record has more than 9500 time entries in the Time Tracking sublist, all cases include a static list of time entries. These static lists are not accessible to scripting. Attempts to script on Time Tracking sublists that are static lists of time entries will not return accurate results.

Notes on Scripting Customer Sublists

You can update the contactaccessroles sublist to provide Customer Center access to contacts. You can provide access to contacts that already exist in NetSuite and that have already been attached to a customer that already exists in NetSuite. The workflow is as follows: 1) Add customer. 2) Add contacts. 3) Attach contacts to customer. 4) Update customer with contact access information.

The fields in this sublist map to the fields on the System Information, Access subtab in the UI. These fields include: a Boolean field that indicates whether a contact has access to NetSuite, contact name key field, email address used to log in to NetSuite, password used to log in to NetSuite, NetSuite role (Customer Center), and a Boolean field that indicates whether the contact should receive a notification email when access changes are made. If this Notify field is set to true, an email is sent.

Transform Types

In the NetSuite Help Center, see the help topic nlapiTransformRecord(type, id, transformType, transformValues) for examples on how to transform records.

Target Record Type	Target Record Internal ID	Field Defaults
Cash Sale	cashsale	billdate
Customer Payment	customerpayment	
Quote	estimate	

Target Record Type	Target Record Internal ID	Field Defaults
Invoice	invoice	billdate
Opportunity	opportunity	
Sales Order	salesorder	

Customer Status

Note: The content in this help topic pertains to all versions of SuiteScript.

The customer status record describes a lead, prospect, or a customer's stage in the sales cycle. To create a new customer status record, go to Setup > Sales > Setup Tasks > Customer Statuses > New.

The internal ID for this record is `customerstatus`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The customer status record is scriptable in both client and server SuiteScript.

Supported Functions

The customer status record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Employee

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The internal ID for this record is `employee`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module

- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The employee record is scriptable in both client and server SuiteScript.

Supported Functions

The employee record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Usage Notes

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
Body Fields		
password	Password	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).
password2	Confirm Password	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).

Scripting when Advanced Employee Permissions is Enabled

When the Advanced Employee Permissions feature is enabled keep the following in mind:

- Before submit scripts can be run, either as the role that has manipulated the record or as an administrator, the script only has access to the information that has been submitted from the browser. In the case of an employee record being edited, the before submit script only has access to the fields or sublists the role editing the record is permitted to edit. Therefore, you cannot assume that all fields and sublists on the employee record are available in a before submit script.
- After submit scripts can be run as administrator, which means complete access to an employee record. When the Advanced Employee Permissions feature is enabled, perform actions, such as setting the value of a field, as a role that is known to have the appropriate level of access or as an administrator.

For information about this feature, see the help topic Advanced Employee Permissions.

Transform Types

In the NetSuite Help Center, see the help topic nlapiTransformRecord(type, id, transformType, transformValues) for examples on how to transform records.

Target Record Type	Target Record Internal ID	Field Defaults
Expense Report	expensereport	
Time	timebill	

Entity

Note: The content in this help topic pertains to all versions of SuiteScript.

The internal ID for this record is `entity`.

See the SuiteScript Records Browser for all internal IDs associated with this record. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The entity record is only exposed in search.

Supported Functions

Only search is supported for the entity record.

Generic Resource

Note: The content in this help topic pertains to all versions of SuiteScript.

Generic resource records can be used as placeholders when project managers and resource managers are planning a project in NetSuite. This feature enables resource allocations and project task assignments to be made when a specific resource may not yet be identified.

For details about working with generic resources in the user interface, see the help topic Generic Resources.

The internal ID for this record is `genericresource`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Use a generic resource as a placeholder for resource allocation and project task assignment. This record is primarily used when a specific resource is not available. See the help topic Generic Resources for additional information.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The generic resource record is supported in all client and server-side scripts.

Supported Functions

The following SuiteScript functionality is supported:

- Read
- Create
- Edit
- Delete
- Search

Note: Copy is not supported.

Job

The Job Management feature must be enabled to script with this record.

The internal ID for this record is `job`. For more details about this record and how to work with it in the user interface, see the help topic Managing Jobs.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The job record is scriptable in both client and server SuiteScript.

Supported Functions

The job record can be read, created, updated, searched, and deleted using SuiteScript. It cannot be copied or transformed.

Usage Notes

The Assigned Employees sublist and Job Requisition sublist are not available to SuiteScript.

Code Sample

```
function beforeLoad(type, form, request)
{
  nlapiSetFieldValue('title' , 'abc');
  nlapiSetFieldValue('description' , 'test description');
  nlapiSetFieldValue('isinactive' , 'F');
```

```
}
```

Job Requisition

The Job Requisitions feature must be enabled to script with this record.

The internal ID for this record is `jobrequisition`. For more details about this record and how to work with it in the user interface, see the help topic Job Requisitions.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The job requisition record is scriptable in both client and server SuiteScript.

Supported Functions

The job requisition record can be read, created, updated, searched, and deleted using SuiteScript. It cannot be copied or transformed.

Code Samples

```
function beforeLoad(type, form, request)
{
  nlapiSetFieldValue('title' , 'abc');
  nlapiSetFieldValue('description' , 'test description');
  nlapiSetFieldValue('isinactive' , 'F');

}

function beforeSubmit(type, form, request)
{
  nlapiSetFieldValue('subsidiary' , '1');
  nlapiSetFieldValue('department' , '2');
  nlapiSetFieldValue('location' , '3');
}
```

Lead

Leads are companies or individuals who represent potential customers. In NetSuite, leads are the first step in the sales cycle that progresses to prospect and then to customer.

For more details about the lead record and working with it in the user interface, see the help topic Lead Management Overview.

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The internal ID for this record is `lead`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The lead record is scriptable in both client and server SuiteScript.

Supported Functions

The lead record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Usage Notes

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
Body Fields		
datecreated	Date Created	This is a system-generated field that marks the date the record was created in NetSuite. You cannot change or override this field. Tip: If you need to capture “date created” information that is not related to the date the record was created in NetSuite, create a custom field and set it to auto-default to today's date.
password	Password	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).
password2	Confirm Password	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).
Search Filters and Search Columns		
ccnumber	Credit Card Number	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).

The Multi-Partner Management feature must be enabled in your account for the Partners sublist to appear.

The Access / contactroles sublist is included in the customer record on the Access subtab. It is an inline editor subtab. The Access / contract roles sublist is related to the Contact / contactroles sublist. For details, see the contactroles sublist in the SuiteScript Records Browser.

The Time Tracking sublist is included with the lead record. It is an inline editor sublist.

Important: If any case, task, or event record has more than 9500 time entries in the Time Tracking sublist, all cases include a static list of time entries. These static lists are not accessible to scripting. Attempts to script on Time Tracking sublists that are static lists of time entries will not return accurate results.

Transform Types

In the NetSuite Help Center, see the help topic nlapiTransformRecord(type, id, transformType, transformValues) for examples on how to transform records.

Target Record Name	Target Record Internal ID	Field Defaults
Opportunity	opportunity	

Other Name

Note: The content in this help topic pertains to all versions of SuiteScript.

The list of other name records is a collection of records for people or companies who are not vendors, customers or employees. This enables you to keep records for other people or companies if you write checks to or receive deposits from them.

For example, your company might donate money to a favorite charity, so you create an other name record for the charity. You might also list your owners and partners here if they contribute or withdraw equity.

For more details about the other name record and working with it in the user interface, see the help topic Other Name Records

The internal ID for this record is `othername`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The other name record is scriptable in server SuiteScript only.

Supported Functions

The lead record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Partner

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

A partner is a company you have a business agreement with who isn't a customer or a vendor.

For more details about the partner record and working with it in the user interface, see the help topic Managing Partners.

The internal ID for this record is `partner`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The partner record is scriptable in both client and server SuiteScript.

Supported Functions

The partner record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Usage Notes

Notes on Scripting Partner Fields

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
Body Fields		
password	Password	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).

Field Internal ID	Field UI Label	Note
password2	Confirm Password	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).

Notes on Scripting Partner Sublists

You can update the contactaccessroles sublist to provide Partner Center access to contacts. You can provide access to contacts that already exist in NetSuite and that have already been attached to a partner that already exists in NetSuite. The workflow is as follows: 1) Add partner. 2) Add contacts. 3) Attach contacts to partner. 4) Update partner with contact access information.

The fields in this sublist map to the fields on the System Information, Access subtab in the UI. These fields include: a Boolean field that indicates whether a contact has access to NetSuite, contact name key field, email address used to log in to NetSuite, password used to log in to NetSuite, NetSuite role (Partner Center), and a Boolean field that indicates whether the contact should receive a notification email when access changes are made. If this Notify field is set to true, an email is sent.

The Access / contactroles sublist is included in the customer record on the Access subtab. It is an inline editor subtab. The Access / contract roles sublist is related to the Contact / contactroles sublist. For details, see the contactroles sublist in the SuiteScript Records Browser.

Project (Job)

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Use the project record to manage company initiatives.

To use the project record, you must have the Projects feature enabled at Setup > Company > Enable Features, on the Company tab. If you plan to do advanced project tracking, you must also enable Project Management. If you do not see the Project Management check box, your company must first purchase the Project Management add-on from NetSuite.

To access the project record in the UI, choose Lists > Relationships > Projects (or Jobs). For help working with projects manually, see the help topic Projects.

The internal ID for this record is `job`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The project record is scriptable in both client and server SuiteScript.

Supported Functions

The project record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Field Definitions

Note that the datecreated body field is a system-generated field that marks the date the record was created in NetSuite. You cannot change or override this field. If you need to capture “date created” information that is not related to the date the record was created in NetSuite, create a custom field and set it to auto-default to today's date.

For more details on available fields, see the SuiteScript Records Browser, which lists all internal IDs associated with this record. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

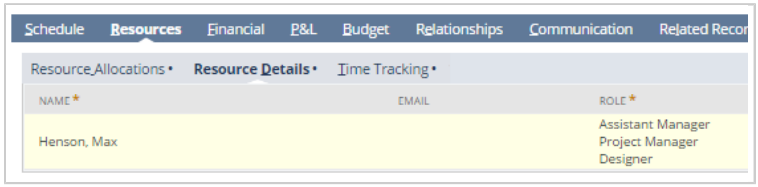
Usage Notes

The Time Tracking sublist is included with the project (job) record. It is an inline editor sublist.

Important: If any case, task, or event record has more than 9500 time entries in the Time Tracking sublist, all cases include a static list of time entries. These static lists are not accessible to scripting. Attempts to script on Time Tracking sublists that are static lists of time entries will not return accurate results.

Adding a Resource With Multiple Roles

When using SuiteScript, the behavior of the Resources sublist differs slightly from the behavior in the UI. Specifically, in the UI, each line must have a unique value in the Name field. Further, in the UI, you can specify more than one role for each resource.



With SuiteScript, if you want to add a resource that has two different roles, you set up your code as if you are adding two sublist records for that resource -- one for each role. The jobresource values are not required to be unique. The following example illustrates this technique. In this example, only two unique resources are being added, but because one resource has two roles, that resource is represented twice.

```
proj.setFieldValue('companyname', 'Launch');
proj.setFieldValue('subsidiary', '1');

proj.selectNewLineItem('jobresources');
proj.setCurrentLineItemText('jobresources', 'jobresource', 'Employee Resource 1');
proj.setCurrentLineItemText('jobresources', 'role', 'Staff');
proj.commitLineItem('jobresources');

proj.selectNewLineItem('jobresources');
proj.setCurrentLineItemText('jobresources', 'jobresource', 'Employee Resource 1');
```

```
proj.setCurrentLineItemText('jobresources', 'role', 'Project Manager');
proj.commitLineItem('jobresources');

var id = nlapiSubmitRecord(proj);
var savedProj = nlapiLoadRecord('job', id);
savedProj.selectLineItem('jobresources', '1');
savedProj.setCurrentLineItemText('jobresources', 'jobresource', 'Employee Resource 2');
savedProj.commitLineItem('jobresources');
nlapiSubmitRecord(savedProj);
```

Note that after you complete the add operation, the Resources sublist in the UI looks the same as it would if you had manually added one line for the resource, with multiple roles specified on that line, as shown in the illustration above.

Scripting Projects with Advanced Revenue Management

The following table lists the scriptable fields associated with the Project record and advanced revenue management.

Field	Type	Internal ID
Accounting Period	List/Record	`accountingperiod`
Comments	Text Area	`comments`
Percentage Complete	Percent	`percent`

Before you begin working with advanced revenue management programmatically, see the help topic Setup for Advanced Revenue Management.

For help working with this record in the user interface, see the help topic Advanced Revenue Management for Projects.

Project Status

Note: The content in this help topic pertains to all versions of SuiteScript.

The project status record indicates the progress of the project.

You can create new statuses at Setup > Accounting > Accounting Lists > New > Project Status.

To use the project status and Project (Job) records, you must have the Projects feature enabled at Setup > Company > Enable Features, on the Company tab. If you plan to do advanced project tracking, you must also enable Project Management. If you do not see the Project Management check box, your company must first purchase the Project Management add-on from NetSuite.

For details about how the project status record is used with the Project (Job) record, see the help topic Creating a Project Record.

The internal ID for this record is `jobstatus`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The project status record is scriptable in both client and server SuiteScript.

Supported Functions

The project status record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Project Template

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Project Templates enable you to create project records in NetSuite for projects your business performs repeatedly.

To use the project template and Project (Job) records, you must have the Projects feature enabled at Setup > Company > Enable Features, on the Company tab. If you plan to do advanced project tracking, you must also enable Project Management. If you do not see the Project Management check box, your company must first purchase the Project Management add-on from NetSuite.

For more details about the project template record and working with it in the user interface, see the help topic Project Templates.

The internal ID for this record is `projecttemplate`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Use a project template as a standard starting point for projects and project items. Each record instance is reusable. See the help topic Project Templates for information on using project templates in the UI.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The project template record is supported in all client and server-side scripts.

Supported Functions

The following SuiteScript functionality is supported:

- Read
- Create
- Edit
- Delete
- Search

Note: Copy is not supported.

Project Type

Note: The content in this help topic pertains to all versions of SuiteScript.

To use the project type and Project (Job) records, you must have the Projects feature enabled at Setup > Company > Enable Features, on the Company tab. If you plan to do advanced project tracking, you must also enable Project Management. If you do not see the Project Management check box, your company must first purchase the Project Management add-on from NetSuite.

The internal ID for this record is `jobtype`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The project type record is scriptable in both client and server SuiteScript.

Supported Functions

The project type record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Prospect

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Prospect records enable you to track all the information you need to convert a prospect into a customer.

For more details about the prospect record and working with it in the user interface, see the help topic Prospects.

The internal ID for this record is `prospect`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The prospect record is scriptable in both client and server SuiteScript.

Supported Functions

The prospect record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Usage Notes

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
Body Fields		
password	Password	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).
password2	Confirm Password	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).
Search Filters and Search Columns		
ccnumber	Credit Card Number	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).

The Multi-Partner Management feature must be enabled in your account for the Partners sublist to appear.

The Access / contactroles sublist is included in the customer record on the Access subtab. It is an inline editor subtab. The Access / contract roles sublist is related to the Contact / contactroles sublist. For details, see the contactroles sublist in the SuiteScript Records Browser.

The Time Tracking sublist is included with the prospect record. It is an inline editor sublist.

Important: If any case, task, or event record has more than 9500 time entries in the Time Tracking sublist, all cases include a static list of time entries. These static lists are not accessible to scripting. Attempts to script on Time Tracking sublists that are static lists of time entries will not return accurate results.

Transform Types

In the NetSuite Help Center, see the help topic nlapiTransformRecord(type, id, transformType, transformValues) for examples on how to transform records.

Target Record Name	Target Record Internal ID	Field Defaults
Estimate/Quote	estimate	
Opportunity	opportunity	
Sales Order	salesorder	

Time-Off Management

The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The following record types are related to the Time-Off Management feature:

- Time-Off Change
- Time-Off Plan
- Time-Off Request
- Time-Off Rule
- Time-Off Type

Time-Off Change

The internal ID for this record is `timeoffchange`. For more details about this record and how to work with it in the user interface, see the help topic Viewing an Employee's Time-Off Balance.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The time-off change record is scriptable in server SuiteScript only.

Note that triggering of user events is not supported for this record.

Supported Functions

The time-off change record is not fully scriptable. Transform and copy are not permitted.

Code Samples

The following example shows how to create a time-off change.

```
<code>

var record = nlapiCreateRecord('timeoffchange');
record.setFieldValue('timeofftype','1');//ID of time-off type. In this example, ID of 1 is used
.
record.setFieldValue('description','Time-Off Type Name');
record.setFieldValue('dateapplied','12/25/2016');
record.selectFieldValue('amount', '1');
record.selectFieldValue('timeoffunit', '1');//ID of unit from drop-down.
var id = nlapiSubmitRecord(record);

</code>
```

Time-Off Plan

The internal ID for this record is `timeoffplan`. For more details about this record and how to work with it in the user interface, see the help topic Creating a Time-Off Plan.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The time-off plan record is scriptable in server SuiteScript only.

Note that triggering of user events is not supported for this record.

Supported Functions

The time-off plan record can be read, created, updated, searched, and deleted using SuiteScript. It cannot be copied or transformed.

Code Samples

The following example shows how to create a time-off plan and modify an existing time-off plan.

```
<code>

//Create the Time-Off Plan record
var record = nlapiCreateRecord('timeoffplan');
record.setFieldValue('name','new Time-Off Plan');
record.setFieldValue('isinactive','F');
record.setFieldValue('includefutureaccruals','F');
record.setFieldValue('subsidiary','1');//Use ID of subsidiary, location, department, or class.
In this example, ID of 1 is used.
record.setFieldValue('location','1');
record.setFieldValue('department','1');
record.setFieldValue('class','1');
record.setFieldValue('startmonth','1');

var id = nlapiSubmitRecord(record);

//Modify existing Time-Off Plan record
var record = nlapiLoadRecord('timeoffplan', id);
record.setFieldValue('isinactive','T');//Inactivate plan.
nlapiSubmitRecord(record);

</code>
```

Time-Off Request

The internal ID for this record is `timeoffrequest`. For more details about this record and how to work with it in the user interface, see the help topics Submitting Time-Off Requests and Approving or Rejecting Time-Off Requests.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

> **Note:** To submit time-off requests, employees must have a start date and time-off plan assigned to them on the employee record.

Supported Script Types

The time-off request record is scriptable in server SuiteScript only.

Note that triggering of user events is not supported for this record.

Supported Functions

The time-off request record is not fully scriptable. Transform and copy are not permitted.

Code Samples

The following example shows how to create and approve a time-off request.

```
<code>

//Create Time-Off Request
var record = nlapiCreateRecord('timeoffrequest');
record.setFieldValue('employee','25');
record.setFieldValue('startdate','12/25/2016');
record.setFieldValue('enddate','12/26/2016');
record.selectNewLineItem('detail');
record.setCurrentLineItemValue('detail', 'timeoffdate', '12/25/2016');
record.setCurrentLineItemValue('detail', 'timeofftype', '1');
record.setCurrentLineItemValue('detail', 'amountoftime', '1.0');
record.commitLineItem('detail');
var id = nlapiSubmitRecord(record);

//Approve Time-Off Request
var record = nlapiLoadRecord('timeoffrequest', id, true);//Where ID is the ID number of the req
uest.
record.setFieldValue('approvalstatus','8');//Approval status is 8.
nlapiSubmitRecord(record);

</code>
```

Time-Off Rule

The internal ID for this record is `timeoffrule`. For more details about this record and how to work with it in the user interface, see the help topic Creating a Time-Off Rule.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The time-off rule record is scriptable in server SuiteScript only.

Note that triggering of user events is not supported for this record.

Supported Functions

The time-off rule record is not fully scriptable. Copy and search are not permitted.

Code Samples

The following code snippets show how to create a time-off rule.

```
<code>

//Create Time-Off Rule to add to exisiting Time-Off Plan record.
var record = nlapiCreateRecord('timeoffrule');
record.setFieldValue('openingbalancefornewemployees', 'T');
record.setFieldValue('entitlement', '1');
record.setFieldValue('maximumtenure', '1');
record.setFieldValue('minimumtenure', '0');
record.setFieldValue('maximumtenureunit', '1');//ID of unit available in drop-down. In this exa
mple, ID of 1 is used.
record.setFieldValue(minimumtenureunit', '1');
record.setFieldValue('entitlementunit', '1');
record.setFieldValue('timeofftype', '1');//Use ID of Time-Off Type. In this example, ID of 1 is
 used.
record.setFieldValue('timeoffplan', '1');//Use ID of Time-Off Plan you want to assign a rule to
. In this example, ID of 1 is used.
var id = nlapiSubmitRecord(record);

</code>
```

Time-Off Type

The internal ID for this record is `timeofftype`. For more details about this record and how to work with it in the user interface, see the help topic Creating a Time-Off Type.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The time-off type record is scriptable in server SuiteScript only.

Note that triggering of user events is not supported for this record.

Supported Functions

The time-off type record is not fully scriptable. Transform and copy are not permitted.

Code Samples

The following example shows how to create a time-off type.

```
<code>

//Create Time-Off Type
var record = nlapiCreateRecord('timeofftype');
record.setFieldValue('name','Time-Off Type Name');
record.setFieldValue('displayname','Time-Off Type Name');
record.setFieldValue('istrackonly','T');
record.setFieldValue('isautogeneratetimeentry', 'T');
record.setFieldValue('minimumincrement', '1');
record.setFieldValue('incrementunit', '1')//ID of unit available in drop-down. In this example,
 ID of 1 is used.
record.setFieldValue ('payrollitem', '1');// Use ID of payroll item in the drop-down. In this e
xample, ID of 1 is used.
var id = nlapiSubmitRecord(record);

</code>
```

Vendor

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

A vendor is a company or person you purchase goods and services from. Vendor records track information about your vendors and enable you to view past transactions and communications with them.

For more details about the vendor record and working with it in the user interface, see the help topic Vendor Records Overview.

The internal ID for this record is `vendor`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The vendor record is scriptable in both client and server SuiteScript.

Supported Functions

The vendor record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Usage Notes

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
Body Fields		
password	Password	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).
password2	Confirm Password	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).

Transform Types

In the NetSuite Help Center, see the help topic nlapiTransformRecord(type, id, transformType, transformValues) for examples on how to transform records.

Target Record Name	Target Record Internal ID	Field Defaults
Purchase Order	purchaseorder	
Vendor Bill	vendorbill	
Vendor Payment	vendorpayment	

Items

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

- Using Item Records in SuiteScript
- Pricing Sublist / Pricing Matrix
- Assembly Item
- Description
- Discount
- Download Item
- Gift Certificate Item
- Inventory Item
- Item Search
- Item Group
- Kit
- Lot Numbered Assembly Item
- Lot Numbered Inventory Item

- Markup
- Non-Inventory Part
- Other Charge Item
- Payment
- Reallocate Items
- Serialized Assembly Item
- Serialized Inventory Item
- Service
- Shipping Item
- Subscription Plan
- Subtotal

Using Item Records in SuiteScript

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

This section includes the following topics:

- Loading Item Types
- Filtering Items by Type
- Advanced Revenue Management Scripting with Items

For information about working with items in the UI, see the help topics Using Item Records and Item Types.

Loading Item Types

When using nlapiLoadRecord(type, id, initializeValues), you can:

- set the *type* parameter to 'inventoryitem' to load the following types of item records: inventoryitem, lotnumberedinventoryitem, serializedinventoryitem
- set the *type* parameter to 'assemblyitem' to load the following types of item records: assemblyitem, lotnumberedassemblyitem, serializedassemblyitem

Filtering Items by Type

The following are valid search filter item type IDs. Note that the item filter IDs are case-sensitive.

Item Type IDs	
Assembly Description Discount DwnLdItem EndGroup GiftCert Group InvtPart	Markup NonInvtPart OthCharge Payment Service ShipItem Subtotal TaxGroup

Item Type IDs	
Kit	TaxItem

To use these IDs:

1. Create a script that will search for items of a specific type or types (for example, search for all non-inventory items).
2. Next, see any of the valid SuiteScript item type IDs.

Sample Code

```
//Create a script that will search for all non-inventory part items
function searchnoninventorypart()
{
var filters = new Array();
filters[0] = new nlobjSearchFilter('type', null, 'anyof', 'NonInvtPart');
var columns = new Array();
columns[0] = new nlobjSearchColumn('internalId');
var items = nlapiSearchRecord('item', null, filters, columns);
}
```

Advanced Revenue Management Scripting with Items

The item record (internal ID `item`) contains several additional accounting fields associated with the Advanced Revenue Management feature. The following table lists these scriptable fields.

Field	Type	Internal ID
Create Revenue Plans On	List/Record	`createrevenueplanson`
Item Revenue Category	List/Record	`itemrevenuecategory`
Revenue Allocation Group	List/Record	`revenueallocationgroup`
Revenue Recognition Rule	List/Record	`revenuerecognitionrule`

Before you begin working with advanced revenue management programmatically, see the help topic Setup for Advanced Revenue Management.

For help working with this record in the UI, see the help topic Item Configuration for Advanced Revenue Management.

Pricing Sublist / Pricing Matrix

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The Pricing sublist is often referred to as a **pricing matrix**, since there can be multiple prices specified for an item, as determined by one or more price levels and one or more quantity levels.

Funtionally, the Pricing sublist shares many of the characteristics of List Sublists. However, scripting to the Pricing sublist is not like scripting to other sublists in NetSuite. For this reason it is recommended that you read all of the following topics to learn about using SuiteScript on this sublist. These topics do not need to be read in order, although it is recommended:

- What is the Pricing Matrix?
- Pricing Sublist Feature Dependencies
- Pricing Sublist Internal IDs
- Pricing Sublist Code Sample
- Matrix Sublist APIs and Standard Sublist APIs

> **Note:** The screenshots in this section depict the NetSuite user interface prior to Version 2010 Release 2.

For general information on item pricing, see these topics in the NetSuite Help Center:

- Item Pricing
- Setting Up Item Pricing
- Using Item Records

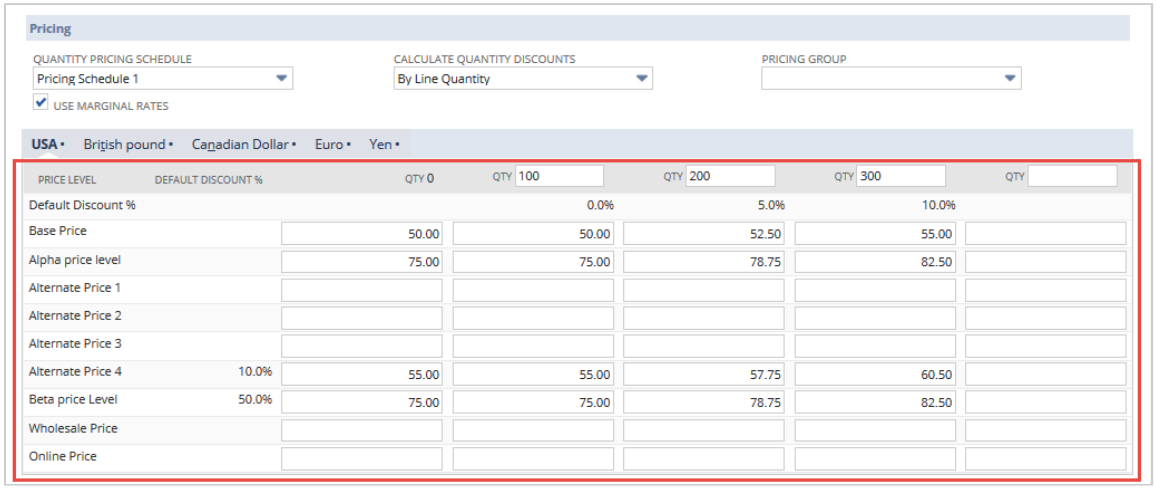
To see which records the Pricing sublist appears on, see Records that Include the Pricing Sublist.

What is the Pricing Matrix?

Depending on the features enabled in your account, the Pricing sublist on many item records can resemble a **matrix** of rows and columns of various prices (see figure). To access the price values on a per-row, per-column basis, you must use Matrix APIs. These APIs are a subset of the non-matrix Sublist APIs that are more commonly used when scripting with other sublists.

> **Note:** Non-matrix sublist APIs can also be used on the Pricing sublist. However, they are not used to get|set values considered to be part of the **pricing matrix**. For information on when to use matrix and non-matrix APIs on the Pricing sublist, see Matrix Sublist APIs and Standard Sublist APIs.

The figure below provides an overview of the rows and columns considered to be the **pricing matrix**. As previously stated, the configuration of the Pricing sublist greatly depends on the features enabled in your account. However, regardless of the features set, all configurations will have some variation of a row / column matrix layout like the one shown below.



This code snippet shows the kind of values you will typically set when working with price values in the pricing matrix. The internal ID of the Pricing sublist, as well as its field IDs, will change depending on the features enabled in your account.

> **Note:** See Pricing Sublist Feature Dependencies and Pricing Sublist Internal IDs for more information.

Example

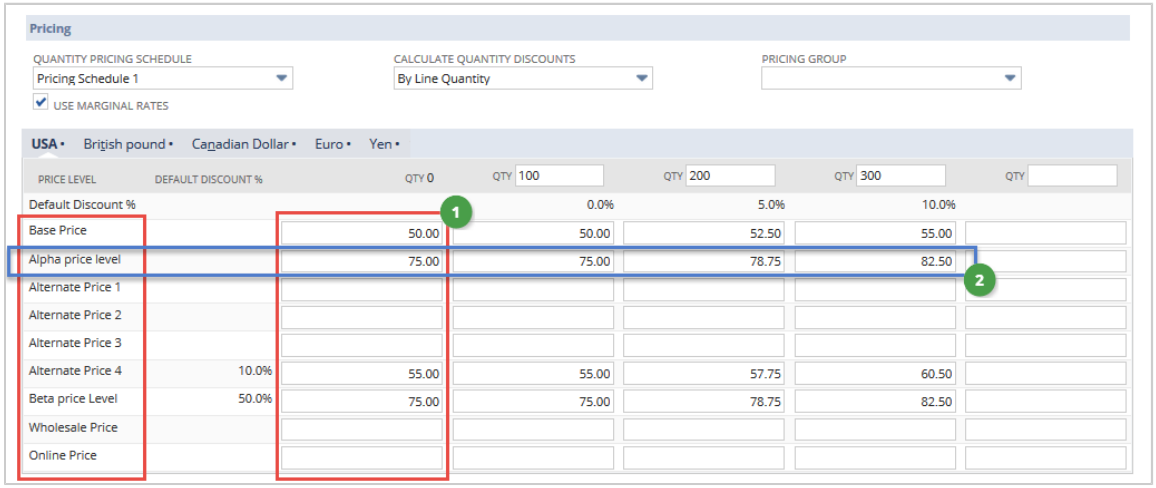
//nlapiGetLineItemMatrixValue(type, fldnam, linenum, column)
nlapiGetLineItemMatrixValue('price', 'price', 2, 1);

In this sample you:

1. Specificy the sublist internal ID (**price**).
2. Specify the internal ID of the pricing field (which will generally be **price**).

   > **Important:** Although the UI labels in this figure show field names such as Alternate Price 1, Alternate Price 2, and Online Price, the internal ID for the **fldnam** parameter is still **price**. The only exception to this is described in Pricing Sublist Field IDs for the **currency** field.

3. Specify the line number (row) of the price you want to get (in this sample, you are getting the value in row 2 - this is the price for **Alternate Price 1**).
4. Specify the column number you want to get the value for (in this sample, you are getting the value in column 1 for **Alternate Price 1**).



1	Column one
2	Row two

Price is typically the internal ID for the fldnam parameter.

Pricing Sublist Feature Dependencies

There are three features that, if enabled or disabled in your account, can affect the overall functionality of the Pricing sublist, its appearance in the UI, and the internal IDs that are referenced in SuiteScript.

You can check which of these features are enabled by looking in the UI, or by calling the nlobjGetContext.getFeature(name) method and specifying the feature internal ID.

The features that affect the Pricing sublist are:

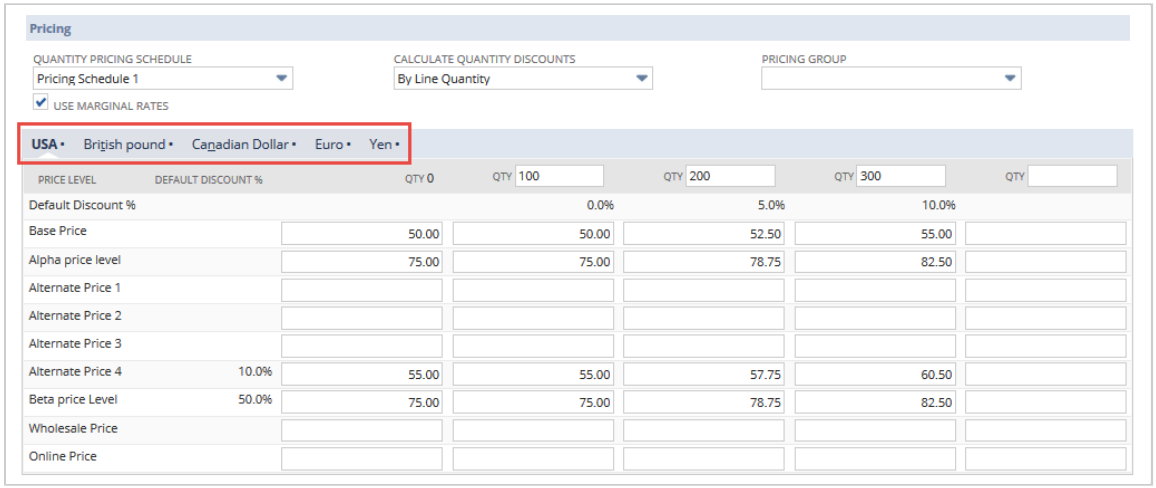
- Multiple Currencies
- Multiple Prices
- Quantity Pricing

If none of these features are enabled in your account, then there is no Pricing sublist on the item record, and the field that holds the item price appears on the Basic subtab as **Sales Price** (see figure). The internal ID for this field is **rate**. You do not use Sublist APIs to set or get values on **rate**. Instead, use Field APIs.

Multiple Currencies

This feature allows for item prices to be set in multiple currencies. Separate pricing is specified for each currency. On the Pricing sublist you will see subtabs with the name of the currencies specified in your account (see figure).

Important: See Pricing Sublist ID to learn how to determine the internal ID of the Pricing sublist based on whether the Multiple Currencies feature is enabled.



In the UI, you can check if this feature is enabled by looking at the Pricing sublist itself or by going to Setup > Company > Enable Features. On the Company tab, the **Multiple Currencies** box will be selected if this feature is enabled. Note: Only a NetSuite administrator can enable this feature.

In SuiteScript, you can get the feature status by writing something similar to:

```
var multiCurrency = nlapiGetContext().getFeature('MULTICURRENCY');
```

Important: See Pricing Sublist Code Sample for more details.

Multiple Prices

This feature allows different prices to be specified for different conditions or types of customers. This requires that Price Levels are set up. There are a set of standard Price Levels provided by NetSuite, and these can be changed or extended by the customer.

This figure shows the Pricing sublist with the Multiple Prices feature enabled. Notice you can specify multiple prices for the same item.

Pricing

QUANTITY PRICING SCHEDULE: Pricing Schedule 1
CALCULATE QUANTITY DISCOUNTS: By Line Quantity
PRICING GROUP:
USE MARGINAL RATES

USA • British pound • Canadian Dollar • Euro • Yen •

PRICE LEVEL	DEFAULT DISCOUNT %	QTY 0	QTY 100	QTY 200	QTY 300	QTY
Default Discount %			0.0%	5.0%	10.0%	
Base Price		50.00	50.00	52.50	55.00	
Alpha price level		75.00	75.00	78.75	82.50	
Alternate Price 1						
Alternate Price 2						
Alternate Price 3						
Alternate Price 4	10.0%	55.00	55.00	57.75	60.50	
Beta price Level	50.0%	75.00	75.00	78.75	82.50	
Wholesale Price						
Online Price						

By comparison, this figure shows the Pricing sublist with the Multiple Prices feature disabled. You can set only one price for the item.

Pricing

QUANTITY PRICING SCHEDULE: Pricing Schedule 1
USE MARGINAL RATES
CALCULATE QUANTITY DISCOUNTS: By Line Quantity

Price Levels •

CURRENCY	QTY 0	QTY 100	QTY 200	QTY 300	QTY
USA	12.00	12.00	12.60	13.20	
British pound					
Canadian Dollar					
Euro					
Yen					

In the UI, you can check if this feature is enabled by looking at the Pricing sublist itself or by going to Setup > Company > Enable Features. On the Transactions subtab, the **Multiple Prices** check box will be selected if this feature is enabled. Note: Only a NetSuite administrator can enable this feature.

In SuiteScript, you can get the feature status by writing something similar to:

```
var multiPrice = nlapiGetContext().getFeature('MULTPRICE');
```

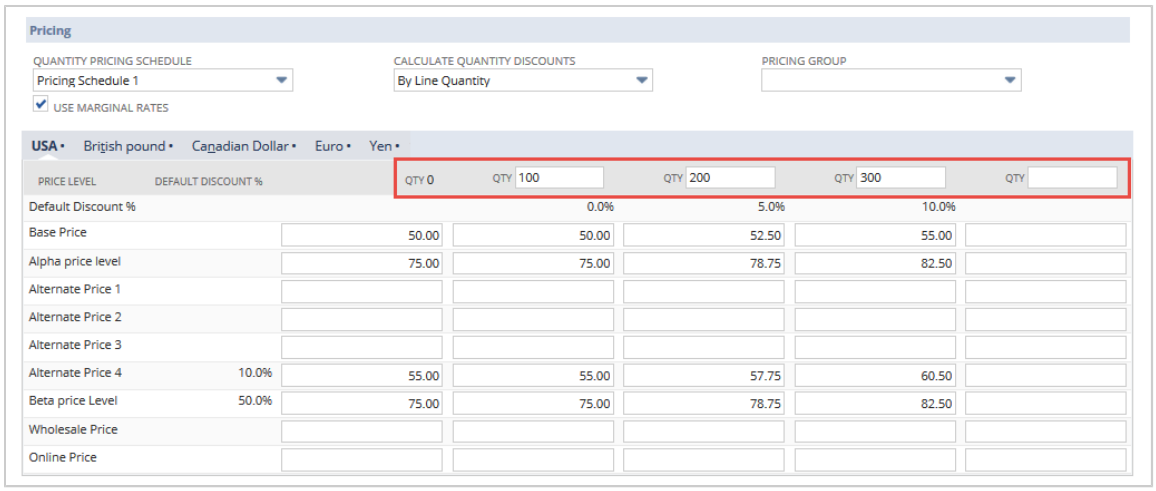
Important: See Pricing Sublist Code Sample for more details.

Quantity Pricing

This feature allows the item price to vary based on the quantity of items sold. Specifically, this feature allows different quantity levels to be specified and allows the price to vary at each quantity level.

This figure shows the Pricing sublist with the Quantity Pricing feature enabled.

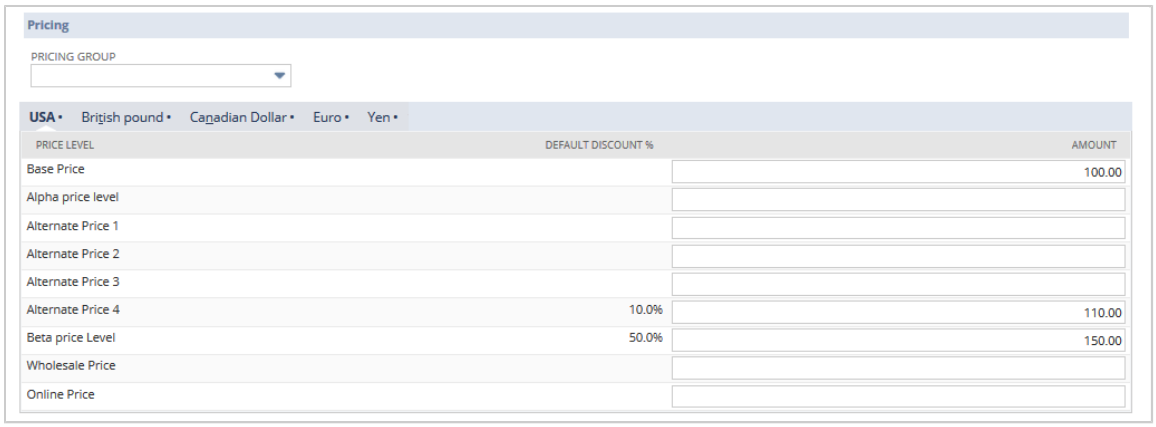
Note: When the Quantity Pricing feature is enabled, an administrator can specify the number of Qty columns that appear on the Pricing sublist. The following figure shows that four Qty columns have been specified. Set the Qty preference by going to Setup > Accounting > Accounting Preferences > Items & Transactions. In the **Maximum # of Quantity-based Price Levels** field, specify the number of columns.



Item pricing is determined by values specified in the Qty fields.

By comparison, this figure shows the Pricing sublist with the Quantity Pricing feature disabled. Item prices are not determined by the quantities specified.

No Qty fields exist in the following image:



In the UI, you can check if this feature is enabled by looking at the Pricing sublist itself or by going to Setup > Company > Enable Features. On the Transactions subtab, the **Quantity Pricing** check box will be selected if this feature is enabled. Note: Only a NetSuite administrator can enable this feature.

In SuiteScript, you can get the feature status by writing something similar to:

```
var quantityPricing = nlapiGetContext().getFeature('QUANTITYPRICING');
```

Important: See Pricing Sublist Code Sample for more details.

Pricing Sublist Internal IDs

As discussed in Pricing Sublist Feature Dependencies, the Pricing sublist looks and functions different depending on the features set in your account.

See Pricing Sublist ID for the internal ID of the Pricing sublist depending on features enabled in your account.

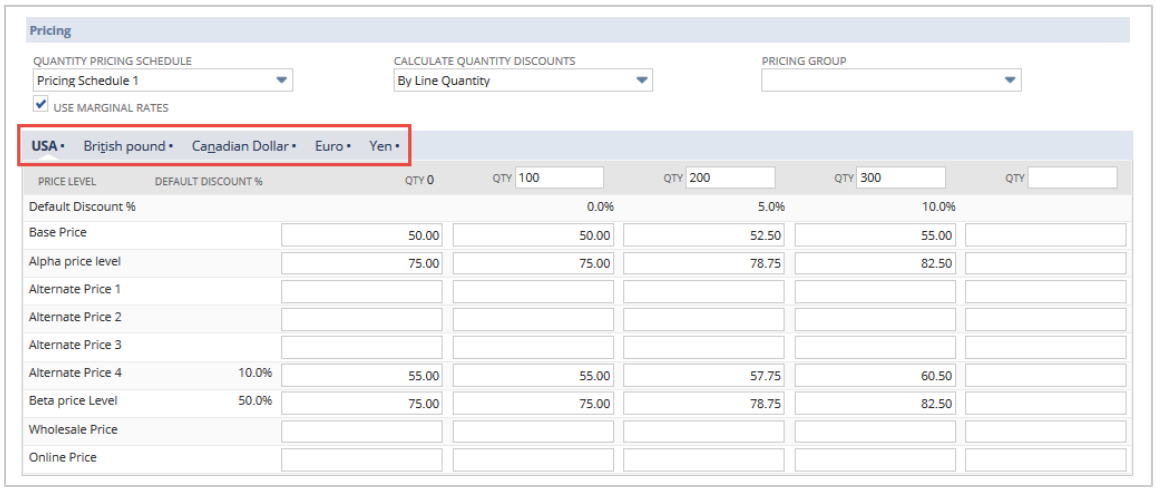
See Pricing Sublist Field IDs for all other field IDs associated with this sublist.

Pricing Sublist ID

In SuiteScript, the internal ID of the Pricing sublist is determined by the features enabled in your NetSuite account.

If the Multiple Currencies feature is **not** enabled in your account, the internal ID for the Pricing sublist is **price**. This means that you will set the **type** parameter in APIs such as nlapiGetMatrixField(**type**, fldnam, column) and nlapiSetLineItemMatrixValue(**type**, fldnam, linenum, column, value) to **price**.

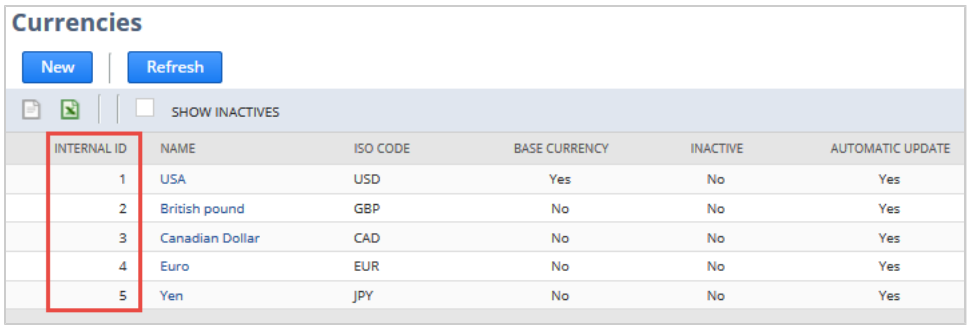
If Multiple Currencies is enabled, then there are separate Pricing sublists per currency (see figure).



Each currency pricing list will have its own internal ID. For example, the internal ID for the currency called USA will be **price1**. This ID reflects the internal ID of the sublist (**price**) and the internal ID of the USA currency (**1**).

The internal ID for the Canadian dollar sublist will be **price3**. This reflects the internal ID of the sublist (**price**) and the internal ID of the Canadian dollar currency (**3**).

This figure shows the currencies that have been set in this account. Notice the Internal ID for each currency is the numberic value appended to **price**. When the Multiple Currencies feature is enabled, you can see the internal ID for each currency by going to Lists > Accounting > Currencies.



INTERNAL ID	NAME	ISO CODE	BASE CURRENCY	INACTIVE	AUTOMATIC UPDATE
1	USA	USD	Yes	No	Yes
2	British pound	GBP	No	No	Yes
3	Canadian Dollar	CAD	No	No	Yes
4	Euro	EUR	No	No	Yes
5	Yen	JPY	No	No	Yes

Based on the internal ID in the figure above, you will set the type parameter in APIs such as nlapiGetMatrixField as follows:

nlapiGetMatrixField(**price1**, fldnam, column) // if scripting on the USA tab
nlapiGetMatrixField(**price2**, fldnam, column) // if scripting on the British pound tab
nlapiGetMatrixField(**price3**, fldnam, column) // if scripting on the Canadian dollar tab
nlapiGetMatrixField(**price4**, fldnam, column) // if scripting on the Euro tab

For topics related to this one, see Pricing Sublist Feature Dependencies.

Pricing Sublist Field IDs

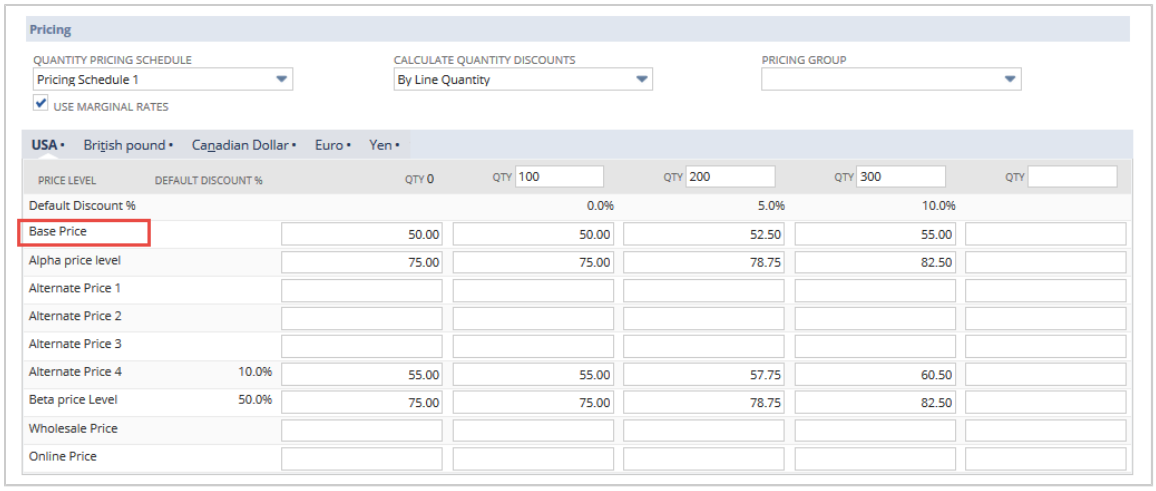
This table provides the internal IDs for all fields associated with the Pricing sublist. Field types are categorized as **matrix** fields, **sublist** fields, and **body** fields.

In SuiteScript, use the IDs that appear in the "Field Internal ID" column for the **fldnam** values in Sublist APIs and Field APIs.

Field UI Label	Field Internal ID	Field Type	Mandatory	Field Notes
Maxtrix Fields				
Base Price Qty	price	string	true	The price for that level **and** quantity. See Figure 1 - Matrix Fields. **Important:** When using matrix APIs (ie., any API that has the word **Matrix** in its name), **price** will generally be the value specified for the **fldnam** parameter. The exception to this is if the Multiple Currencies feature is enabled in your account.
Sublist Fields				
Default Discount %	discount			See Figure 2 - Sublist Fields. See also Standard Sublist APIs for a code sample that references this internal ID.
	currency			The currency field is a hidden field. It is not visible in the UI. See Figure 2 - Sublist Fields. This field is scriptable only when the Multiple Currencies feature is enabled. The internal IDs for the **fldname** parameter in matrix APIs will be price1currency, price2currency,

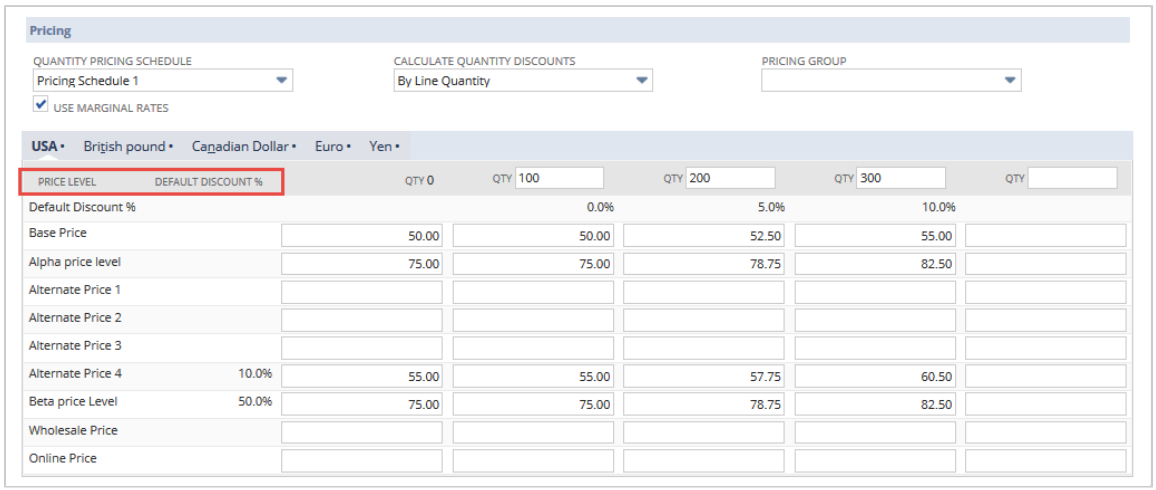
Field UI Label	Field Internal ID	Field Type	Mandatory	Field Notes
				pricecurrency3, etc., to reflect the currency internal ID.
Price Level	pricelevel			The pricelevel for this price. See Figure 2 - Sublist Fields. See also Standard Sublist APIs for a code sample that references this internal ID.
Body Fields				
Quantity Pricing Schedule	quantitypricingschedule	select	false	If a quantity pricing schedule has been specified in the UI drop-down, item prices will be calculated for the pricing matrix according to the specified schedule. See Figure 3 - Body Fields. Also note that the value of the quanity pricing schedule sets the value of the Calculate Quantity Discounts (overallquantitypricingtype) field. Use Field APIs to access this field.
Calculate Quantity Discounts	overallquantitypricingtype	select	false	Used to determine the quantity amount at the time the item is priced on the order. (This field does not change price settings in the matrix). See Figure 3 - Body Fields. Use Field APIs to access this field.
Use Marginal Rates	usemarginalrates	checkbox	false	Used to determine how the quantity discounts are applied at the time the item is priced on the order with a specified quantity. (This field does not change the price settings in the matrix.) See Figure 3 - Body Fields. Use Field APIs to access this field.
Pricing Group	pricinggroup	select	false	Used to provide customer- specific pricing. Could affect the pricing at the time the item is placed on the order and associated with a specific customer. See Figure 3 - Body Fields. Use Field APIs to access this field.

Figure 1. Figure 1 - Matrix Fields



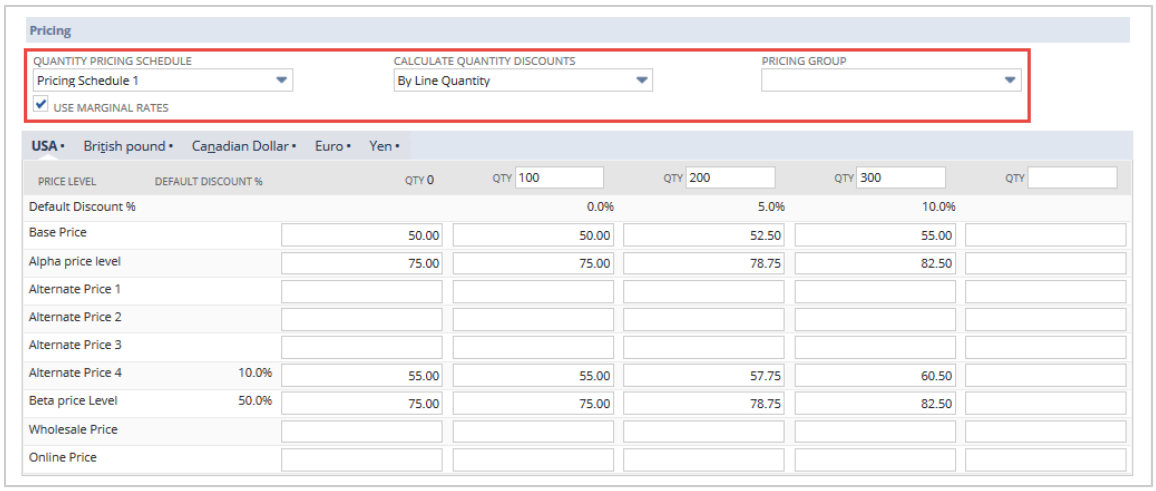
Pricing sublist matrix field.

Figure 2. Figure 2 - Sublist Fields



Use non-matrix Sublist APIs to get or set values for these pricing sublist fields.

Figure 3. Figure 3 - Body Fields



Use standard Field APIs to get or set values for Pricing sublist body fields.

Pricing Sublist Code Sample

This sample shows how to determine which pricing-related features are enabled in your account. It then shows how to programmatically determine the internal ID for the Pricing sublist itself, and then check to see if a Quantity Schedule has been applied to the items in this list. The script also shows how to set item prices and quanity levels depending on various conditions set within the pricing matrix.

> **Note:** If your browser is inserting scroll bars in this code sample, maximize your browser window, or expand the main frame that this sample appears in.

```
// Check the features enabled in the account. See
Pricing Sublist Feature Dependencies for
// details on why this is important.
var multiCurrency = nlapiGetContext().getFeature('MULTICURRENCY');
var multiPrice = nlapiGetContext().getFeature('MULTPRICE');
```

```
var quantityPricing = nlapiGetContext().getFeature('QUANTITYPRICING');

// Set the name of the Price sublist based on features enabled and currency type.
// See
Pricing Sublist Internal IDs for details on why this is important.
var priceID;
var currencyID = "EUR";

// Set the ID for the sublist and the price field. Note that if all pricing-related features
// are disabled, you will set the price in the rate field. See
Pricing Sublist Feature Dependencies
// for details.
if (multiCurrency == 'F' && multiPrice == 'F' && quantityPricing == 'F' )
      priceID = "rate";
   else
   {
      priceID = "price";
      if ( multiCurrency == "T" )
      {
         var internalId = nlapiSearchRecord('currency', null, new nlobjSearchFilter('symbol',
         null,'contains', currencyID))[0].getId();

           // Append the currency ID to the sublist name
                    priceID = priceID + internalId;
      }
   }

// Check to see if the item is using a Quantity Schedule
// If a Quantity Schedule is used, only the base price needs to be set.
// All other prices will be set according to the schedule.
var itemRecord = nlapiLoadRecord('inventoryitem', itemID);
var qtyPriceSchedule = itemRecord.getFieldValue('quantitypricingschedule');

// Set the base price
var basePrice = 100;

// You must select, set, and then commit the sublist line you want to change.
itemRecord.selectLineItem(priceID, 1);
itemRecord.setCurrentLineItemMatrixValue(priceID, 'price', 1, basePrice);
itemRecord.commitLineItem(priceID);

// Get the number of columns in the price matrix
// Each column represents a different quantity level
columnCount = itemRecord.getMatrixCount(priceID, 'price');

// Set the base price in each quantity of the price matrix for a specific sublist, e.g. currenc
y

   // Set the base price in each quantity
   for (var j=1; j<=columnCount; j++)
        {

      // Set the price for this cell of the matrix
      itemRecord.selectLineItem(priceID, 1);
      itemRecord.setCurrentLineItemMatrixValue(priceID, 'price', j, currencyBasePrice);
```

```
      itemRecord.commitLineItem( priceID );
   }

// Display the full price matrix for a specific currency as an HTML table

   // get the size of the matrix
   var quantityLevels = itemRecord.getMatrixCount(priceID, 'price');
   var priceLevels = itemRecord.getLineItemCount(priceID);
   var priceName = "";
   var priceNameField = "pricelevel";
   var itemPrice = 0;
   var fieldObj = null;

   // create a table to present the results
   var strName = "<table>";

   if ( quantityLevels > 1 )
   {
      strName += "<tr>";

      // write out the quantity levels as the first row
           for ( var j=1; j<=quantityLevels; j++)
      {
         strName += "<td>";

         // this Matrix API obtains an nlobjField object
         // the nlobjField object can be used to obtain the UI label
         fieldObj = itemRecord.getMatrixField( priceID, 'price', j);
         if (fieldObj != null )
         strName += fieldObj.getLabel();
         strName += j;
         strName += "</td>";

         strName += "<td>";

         // this Matrix API obtains the value of the Quantity level
         strName += itemRecord.getMatrixValue( priceID, 'price', j);
         strName += "</td>";
      }

           strName += "</tr>";

   }

   // loop through the matrix one row at a time
   for ( var i=1; i<=priceLevels; i++)
   {
           strName += "<tr>";

      // loop through each column of the matrix
           for ( j=1; j<=quantityLevels; j++)
           {
                   // get the price for this cell of the matrix
                   itemPrice = itemRecord.getLineItemMatrixValue( priceID, 'price', i, j);
```

```
        // Get the name of the price level. Note: you will use a standard
        // sublist API and not a matrix API for this.
        priceName = itemRecord.getLineItemText( priceID, priceNameField, i);

        strName += "<td>";
        strName += priceName;
        strName += "</td>";
        strName += "<td>";
        strName += itemPrice;
        strName += "</td>";
          }
          strName += "</tr>";
    }
    strName += "</table>";
```

Matrix Sublist APIs and Standard Sublist APIs

When writing SuiteScript against the Pricing sublist, you may end up using different types of Sublist APIs. If you want to get|set values in the pricing matrix, you will use Matrix APIs.

If you want to get|set non-matrix fields, you will use all non-matrix Sublist APIs or Field APIs, depending on which fields you are trying to access.

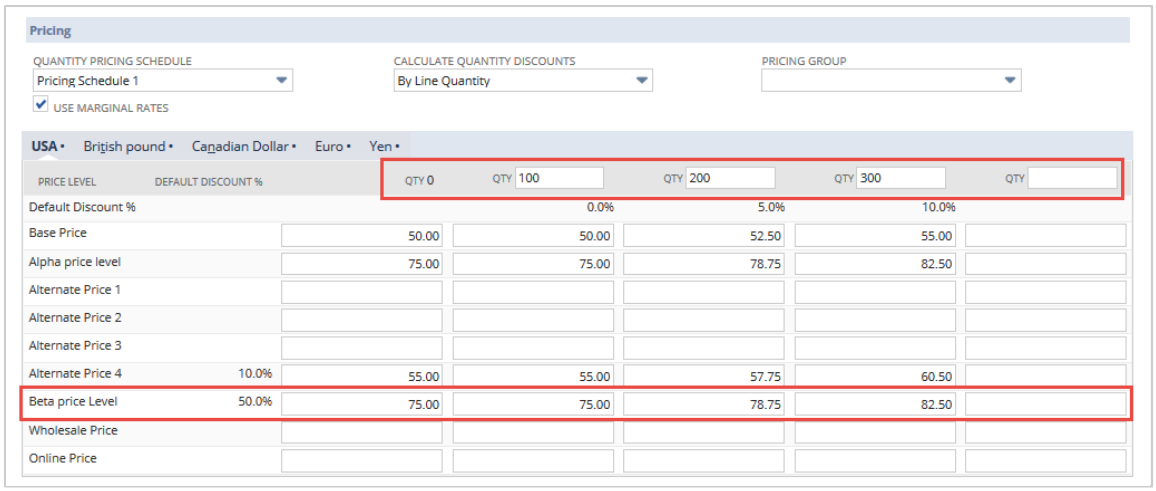
Note: See What is the Pricing Matrix? for information on the pricing matrix. See Pricing Sublist Field IDs for information on the differences among matrix, non-matrix, and body sublist fields.

Matrix APIs

The following are considered to be **matrix** APIs for use on the Pricing sublist. Use these APIs to get|set matrix fields. See Pricing Sublist Field IDs to learn which fields are considered matrix fields.

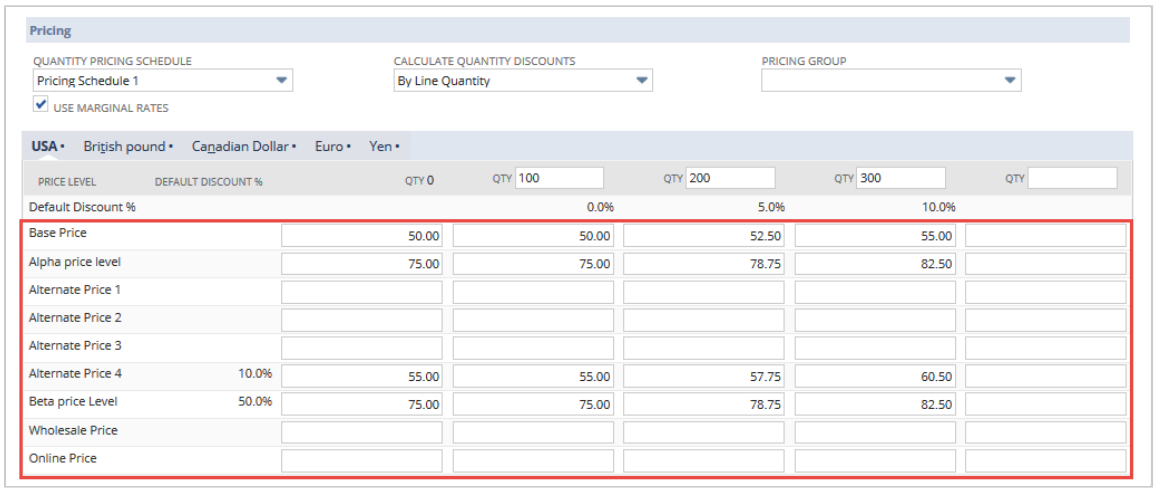
Click these links to see the API documentation for each matrix API. Also see the figures below for a visual representation for where on the pricing matrix each matrix API executes.

- nlapiGetMatrixField(type, fldnam, column)
- nlapiGetMatrixValue(type, fldnam, column)
- nlapiSetMatrixValue(type, fldnam, column, value, firefieldchanged, synchronous)
- nlapiGetMatrixCount(type, fldnam)
- nlapiGetCurrentLineItemMatrixValue(type, fldnam, column)
- nlapiSetCurrentLineItemMatrixValue(type, fldnam, column, value, firefieldchanged, synchronous)
- nlapiGetLineItemMatrixValue(type, fldnam, linenum, column)
- nlapiGetLineItemMatrixField(type, fldnam, linenum, column)
- nlapiFindLineItemMatrixValue(type, fldnam, val, column)



On matrix header fields, use nlapiGetMatrixField, nlapiGetMatrixValue, and nlapiGetMatrixCount.

When on an existing line in the matrix, use nlapiGetCurrentLineItemMatrixValue and nlapiSetCurrentLineItemMatrixValue to get and set the price on that line in the specific column, respectively.



For all other lines in the matrix, use nlapiGetLineItemMatrixValue, nlapiGetItemMatrixField, and nlapiFindLineItemValue.

Standard Sublist APIs

If you want to reference the other fields in the Pricing sublist, such as **currency, name**, or **discount**, use the existing nlapiGetLineItemValue(...) or nlobjRecord.getLineItemValue(...) APIs and pass in the existing fldnam (example: price1currency). Also see Pricing Sublist Field IDs, which specifies which fields on the Pricing sublist can be set using standard Sublist APIs.

Example:

```
// load an item record
var record = nlapiLoadRecord('inventoryitem', 536);

// get the value of the currency field on line 2
```

```
var currency = record.getLineItemValue('price1', 'currency', '2');

// get the value of the pricelevelname field on line 2
var pricelevelname2 = record.getLineItemValue('price1', 'pricelevel', '2');
var pricelevelname3 = record.getLineItemValue('price1', 'pricelevel', '3');
var pricelevelname4 = record.getLineItemValue('price1', 'pricelevel', '4');

// returns the discount from line item 2
var discount2 = record.getLineItemValue('price1', 'discount', '2');
var discount3 = record.getLineItemValue('price1', 'discount', '3');
var discount4 = record.getLineItemValue('price1', 'discount', '4');
```

Records that Include the Pricing Sublist

The Pricing sublist appears on the following records that are scriptable using SuiteScript: Assembly Item, Lot Numbered Assembly Item, Serialized Assembly Item, Lot Numbered Inventory Item, Serialized Inventory Item, Gift Certificate Item, Kit Item, Inventory Item.

Assembly Item

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

An assembly item is an inventory item made of several components, but identified as a single item. Assemblies are manufactured by combining raw materials you stock.

You create assembly item records to define the members of an assembly, then NetSuite enables you to track both the raw materials and the assembled items separately.

For example, an Wolfe Electronics sells a computer called Creativo 2400 that they assemble in-house. The Creativo 2400 computer is assembled from these inventory components: one Superion 2 GHz processor, one Creativo 2400 motherboard, 1 GB RAM, 80 GB Hard Drive, one Superion sound card, and one power supply.

NetSuite tracks the stock of the Creativo 2400 and each component item separately. Then, Wolfe can track the stock level of Creativo 2400 in inventory and available to ship to customers, and the quantity of materials available to assemble more.

For details about this type of item, see the help topic Assembly Items.

The internal ID for this record is `assemblyitem`. This record is also sometimes referred to as Build/Assembly.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The assembly item record is scriptable in server SuiteScript only.

Supported Functions

The assembly item record is fully scriptable — it can be created, updated, copied, deleted, and searched using server SuiteScript. It can also be transformed.

Usage Notes

The taxschedule field of the Members sublist is visible only in the UI when the Advanced Taxes feature is enabled.

The assembly item record includes the Pricing Sublist / Pricing Matrix.

Code Samples

The following sample changes the item name of an assembly.

```
var assembly = nlapiLoadRecord('assemblyitem', 123);
assembly.setFieldValue('itemid', 'new name');
nlapiSubmitRecord(assembly);
```

Description

Note: The content in this help topic pertains to all versions of SuiteScript.

Description line items let you put sentence- or paragraph-long descriptions on items you are not actually selling. For example, you may want to enter special shipping instructions or a disclaimer.

Description items have no amount field. They are only used to add text to transactions. They can be used on both purchase and sales transactions.

The item name of a description item does not appear on printed forms, only the descriptive text. Nothing appears in the amount column for description items.

As you create a transaction, select a description item from the items list.

For information about using description items in the UI, see the help topic Description Items.

The internal ID for this record is `descriptionitem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The description record is scriptable in both client and server SuiteScript.

Supported Functions

The description record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Discount

Note: The content in this help topic pertains to all versions of SuiteScript.

Discount item records are used to create discounts you can apply to your transactions.

For information about using discount items in the UI, see the help topic Discount Items.

The internal ID for this record is `discountitem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The discount record is scriptable in both client and server SuiteScript.

Supported Functions

The description record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Download Item

Note: The content in this help topic pertains to all versions of SuiteScript.

You create download item records for files that you want customers to be able to purchase and download in your Web store.

Customers are charged per download item as opposed to per item. For example, if you want to charge customers for music downloads per song, you would create an item for each song. If you want to charge customers per album, you would create one item and attach each song for the album.

If a download requires a license code, such as a software download, you can add the license code on the Downloads subtab of the customer's record. When a license code is added, the code is available

to the customer in the Customer Center or the My Account tab of your website and included in invoice email notification.

For information about using download items in the UI, see the help topic Download Items.

The internal ID for this record is `downloaditem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The download item record is scriptable in both client and server SuiteScript.

Supported Functions

The download item record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Gift Certificate Item

> **Note:** The content in this help topic pertains to all versions of SuiteScript.

You can create gift certificate items that allow customers to purchase store credit they can send to someone as a gift. The recipient uses the gift certificate code when placing an order through your Web store or entering a transaction with a sales representative. You can set a preference in NetSuite for how you want to generate the gift certificate codes: you can create them yourself, or use a random hash code automatically generated by the system.

Gift certificate codes are not active until the order used to purchase the gift certificate is billed.

For information about using gift certificate items in the UI, see the help topic Gift Certificates.

The internal ID for this record is `giftcertificateitem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The gift certificate record is scriptable in both client and server SuiteScript.

Supported Functions

The gift certificate item record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

The gift certificate item record includes the Pricing Sublist / Pricing Matrix.

Inventory Item

Note: The content in this help topic pertains to all versions of SuiteScript.

NetSuite inventory item records enable you to track the quantity and value of your inventory. Your balance sheet will automatically reflect the value of your inventory on hand, and your income statement will automatically reflect the markup you charge for these items.

This record is also referred to as Inventory Part in the UI.

For information about using inventory items in the UI, see the help topic Inventory Items.

The internal ID for this record is `inventoryitem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The inventory item record is scriptable in both client and server SuiteScript.

Supported Functions

The inventory item record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

The inventory item record includes the Pricing Sublist / Pricing Matrix.

Item Group

Note: The content in this help topic pertains to all versions of SuiteScript.

An item group is stocked and sold as a single unit, but are made up of several individual items.

For information about using item groups in the UI, see the help topic Item Groups.

The internal ID for this record is `itemgroup`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The item group record is scriptable in both client and server SuiteScript.

Supported Functions

The item group record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Item Search

Note: The content in this help topic pertains to all versions of SuiteScript.

The internal ID for this record is `item`.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The item search record is a search record only.

Supported Functions

The item search record is a **search record only**. You cannot create or copy this record.

Kit

Note: The content in this help topic pertains to all versions of SuiteScript.

Kits or packages let you create items that are collected from other items. You can assign multiple price levels to your kits and even make them available in your website. Whenever you sell a kit, inventory items are deducted from inventory.

For information about using kit items in the UI, see the help topic Kit/Package Items.

The internal ID for this record is `kititem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The kit record is scriptable in server SuiteScript.

Supported Functions

The kit record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

The taxschedule field of the Members sublist is visible only in the UI when the Advanced Taxes feature is enabled.

The kit item record includes the Pricing Sublist / Pricing Matrix.

Lot Numbered Assembly Item

Note: The content in this help topic pertains to all versions of SuiteScript.

Lot numbered assembly items enable you to build items from raw materials and track the inventory of both the finished items and the raw materials separately. The completed assembly is assigned a lot number to track it as it enters and leaves your inventory.

For information about using lot numbered items in the UI, see the help topic Lot Numbered Items. For information about using assembly items in the UI, see the help topic Assembly Items.

The internal ID for this record is `lotnumberedassemblyitem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The lot numbered assembly item record is scriptable in both client and server SuiteScript.

Supported Functions

The lot numbered assembly item record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

The taxschedule field of the Members sublist is visible only in the UI when the Advanced Taxes feature is enabled.

The lot numbered assembly item record includes the Pricing Sublist / Pricing Matrix.

Lot Numbered Inventory Item

Note: The content in this help topic pertains to all versions of SuiteScript.

Lot numbered inventory items track the purchase, stock, and sale of groups of items by assigning lot numbers. Lot numbered item records track the quantity of items and the specific cost for each lot as products are purchased and sold.

For information about using lot numbered items in the UI, see the help topic Lot Numbered Items. For information about using inventory items in the UI, see the help topic Inventory Items.

The internal ID for this record is `lotnumberedinventoryitem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The lot numbered inventory item record is scriptable in both client and server SuiteScript.

Supported Functions

The lot numbered inventory item record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

The lot numbered inventory item record includes the Pricing Sublist / Pricing Matrix.

Markup

Note: The content in this help topic pertains to all versions of SuiteScript.

You can use markup items to apply an additional charge to an order. Using markup items enables you to track markup amounts without affecting inventory valuation.

For example, you may want to charge a rush fee for completing a service or delivering an item quicker than is usually guaranteed. You can choose to markup the amount for this charge by a flat additional fee.

For information about using markup items in the UI, see the help topic Markup Items.

The internal ID for this record is `markupitem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The markup record is scriptable in both client and server SuiteScript.

Supported Functions

The markup record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Non-Inventory Part

Note: The content in this help topic pertains to all versions of SuiteScript.

Items that you always drop ship or other items that you sell or purchase but do not stock can be recorded and tracked as non-inventory items.

For information about using non-inventory items in the UI, see the help topic Non-Inventory Items.

The internal ID for this record is `noninventoryitem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The non-inventory record is scriptable in both client and server SuiteScript.

Supported Functions

The non-inventory part record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Other Charge Item

> **Note:** The content in this help topic pertains to all versions of SuiteScript.

Other charge items can be used to designate items or services you purchase or sell that do not fall into another type of item.

For information about using other charge items in the UI, see the help topic Other Charge Items.

The internal ID for this record is `otherchargeitem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Functions

The other charge item record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Payment

Note: The content in this help topic pertains to all versions of SuiteScript.

You can create payment items for types of payments that are made to invoices and should show separately.

For example, you may want to create a payment item to specify a down payment amount.

For information about using payment items in the UI, see the help topic Payment Items.

The internal ID for this record is `paymentitem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The payment record is scriptable in both client and server SuiteScript.

Supported Functions

The payment record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Reallocate Items

Note: The content in this help topic pertains to all versions of SuiteScript.

NetSuite sometimes automatically allocates items from inventory to commit them to fill orders. Inventory may be automatically committed in the following cases:

- As each sales order is created or approved, your account automatically allocates inventory from the item's quantity available.
- Inventory is automatically committed to fill backorders when goods are received from your vendors.
- The inventory generated from assembly work orders is automatically allocated.

For information about working with this type of item in the UI, see the help topic Reallocating Items.

The internal ID for this record is `reallocateitem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The reallocate items record is only exposed in user event scripts. You can execute beforeLoad, beforeSubmit, and afterSubmit user event scripts on this record.

Supported Functions

The reallocate items record cannot be created, loaded, or deleted using scripts.

Serialized Assembly Item

> **Note:** The content in this help topic pertains to all versions of SuiteScript.

Serialized assembly items enable you to build items from raw materials and track the inventory of both finished items and the raw materials separately. The completed assembly is assigned a serial number to track it as it enters and leaves your inventory. Serialized assembly items are available on sales transactions and inventory adjustment transactions. They are not available on purchase transactions.

For information about using serialized items in the UI, see the help topic Serial Numbered Items. For information about using assembly items in the UI, see the help topic Assembly Items.

The internal ID for this record is `serializedassemblyitem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The serialized assembly item record is scriptable in both client and server SuiteScript.

Supported Functions

The serialized assembly record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

The serialized assembly item record includes the Pricing Sublist / Pricing Matrix.

Serialized Inventory Item

Note: The content in this help topic pertains to all versions of SuiteScript.

Serialized inventory item records are used to track information about items which you maintain a stock of. Note that you must first enable serialized inventory items in your NetSuite account before you can access this record type. To enable serialized inventory items, go to Setup > Company > Enable Features. On the Items & Inventory tab, under Inventory, select the Serialized Inventory check box.

For information about using serialized items in the UI, see the help topic Serial Numbered Items. For information about using inventory items in the UI, see the help topic Inventory Items.

The internal ID for this record is `serializedinventoryitem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The serialized inventory item record is scriptable in both client and server SuiteScript.

Supported Functions

The serialized inventory record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

The serialized inventory item record includes the Pricing Sublist / Pricing Matrix.

Service

Note: The content in this help topic pertains to all versions of SuiteScript.

A service item is an item you create to track time and record billable hours.

For information about using service items in the UI, see the help topic Service Items.

The internal ID for this record is `serviceitem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The service record is scriptable in both client and server SuiteScript.

Supported Functions

The service record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Shipping Item

> **Note:** The content in this help topic pertains to all versions of SuiteScript.

A shipping item is a delivery method for a shipping carrier. It describes how to ship an item and can include shipping rate information, handling rates, and rules for shipping and handling. It can also specify when shipping is free. See the help topic Shipping Items for additional information.

The internal ID for this record is `shipitem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The shipping item record is supported in all server-side scripts. Client-side scripting is not supported.

Supported Functions

The following SuiteScript functionality is supported:

- Read
- Edit
- Delete
- Search

Note: Create and copy are not supported. You can create this record only in the UI.

Usage Notes

The following shipping item record components are not scriptable:

- Handling Table
- Shipping Table
- Free Shipping Items Tab

Subscription Plan

Note: The content in this help topic pertains to all versions of SuiteScript.

The subscription plan record is used to build information for subscriptions.

The subscription plan record is available only when the SuiteBilling feature is enabled at Setup > Company > Enable Features (Administrator), on the Transactions subtab. When the feature is enabled, you can access a list of subscription plan records in the UI by going to Lists > Subscriptions > Subscription Plans.

The internal ID for this record is `subscriptionplan`.

See the SuiteScript Records Browser for all internal IDs associated with this record. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The subscription plan record is scriptable in server and client SuiteScript.

Supported Functions

The subscription plan record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Subtotal

Note: The content in this help topic pertains to all versions of SuiteScript.

Subtotal item records are used to separate groups or individual line items if you want to offer a discount or tax on certain line items but not others.

For information about using subtotal items in the UI, see the help topic Subtotal Items.

The internal ID for this record is `subtotalitem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The subtotal record is scriptable in both client and server SuiteScript.

Supported Functions

The subtotal record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Communications

Note: The content in this help topic pertains to all versions of SuiteScript.

The following communication records are scriptable in SuiteScript:

- Message
- Note

Message

Note: The content in this help topic pertains to all versions of SuiteScript.

A message is used to record correspondence you have with a specific business. Use the message record to add an email message to an existing customer, contact, or opportunity record. After an email

message has been added to a record, any related emails are automatically attached to the same record as well as to any recipients of the original email.

For information about adding an email message to a record in the NetSuite user interface, see Sending Email from Records.

The internal ID for this record is `message`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The message record is scriptable in server SuiteScript only.

Supported Functions

For new messages, full scripting is supported. This support enables you to do the equivalent of clicking the **Attach** button in the UI for email messages. Note that there are also other types of messages, but those are not supported. In the message that you are creating, you can set all supported fields for email messages before saving. After you save the new message, there are restrictions applied to it. You cannot edit the existing message. It is possible to delete an existing message, if the permissions for the user are set to allow this type of operation. However, this operation is not recommended, because it can cause email communication threads to be displayed incorrectly.

Usage Notes

- Message records can be edited only during the create operation. After they are created and submitted, existing message records cannot be edited.
- Existing message records can be copied and deleted.
- Only beforeLoad and afterSubmit user event scripts will execute on the Message record type when a message is created by an inbound email case capture. Scripts set to execute on a beforeSubmit event will not execute.

  For example, if you have a test script like the following deployed to the Message record type:

```
 function beforeLoad(type, name)
{
   nlapiLogExecution('DEBUG', 'Before Load');
}
function beforeSubmit(type, name)
{
   nlapiLogExecution('DEBUG', 'Before Submit');
}
```

```
function afterSubmit(type, name)
{
   nlapiLogExecution('DEBUG', 'After Submit');
}
```

only the beforeLoad(...) and afterSubmit(...) functions will execute if the message was created to respond to an emailed case.

- When creating a new message, you can use the ccbcclist sublist to add an email address to the cc or bcc fields. For an example, see Code Sample.
- NetSuite 2016.2 introduced a new requirement for creating a message. An entity that is defined as the author or recipient for a newly created message must have an associated email in NetSuite. If the NetSuite record for an entity defined as the author of a message does not include an email, the `authoremail` field must be defined on the message record. If the NetSuite record for an entity defined as the recipient of a message does not include an email, the `recipientemail` field must be defined on the message record. Prior to 2016.2, the `authoremail` and `recipientemail` fields were not mandatory for the message record.

Code Sample

The following SuiteScript 2.0 snippet shows how to add an email address to the cc line of a new message, when working in standard (deferredDynamic) mode.

```
...
rec.insertLine({
    sublistId: 'ccbcclist',
    line: 0
});

rec.setSublistValue({
    sublistId: 'ccbcclist',
    fieldId: 'email',
    value: 'john@smith.com',
    line: 0
 });

rec.setSublistValue({
    sublistId: 'ccbcclist',
    fieldId: 'cc',
    value: 'true',
    line: 0
 });
...
```

Note

Note: The content in this help topic pertains to all versions of SuiteScript.

Notes are used to attach information to another record. Use the notes record to create new notes and attach them to a specific record.

The internal ID for this record is `note`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The note record is scriptable in server SuiteScript only.

Supported Functions

The note record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Transactions

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

> **Warning:** Memorized transactions do not support SuiteScript.

The following transaction records are scriptable in SuiteScript.

- Advanced Intercompany Journal Entry
- Assembly Build
- Assembly Unbuild
- Bin Putaway Worksheet
- Bin Transfer
- Blanket Purchase Order
- Cash Refund
- Cash Sale
- Change Order
- Charge
- Check
- Credit Card Charge
- Credit Card Refund
- Credit Memo
- Custom Transaction

- Customer Deposit
- Customer Payment
- Customer Refund
- Deposit
- Deposit Application
- Estimate / Quote
- Expense Report
- Fulfillment Request
- Intercompany Journal Entry
- Intercompany Transfer Order
- Inventory Adjustment
- Inventory Cost Revaluation
- Inventory Count
- Inventory Detail
- Inventory Status Change
- Inventory Transfer
- Invoice
- Item Demand Plan
- Item Fulfillment
- Item Receipt
- Item Supply Plan
- Journal Entry
- Landed Cost
- Manufacturing Operation Task
- Manufacturing Planned Time
- Multi-Book Accounting Transaction
- Opportunity
- Order Schedule
- Paycheck
- Paycheck Journal
- Purchase Contract
- Purchase Order
- Requisition
- Return Authorization
- Revenue Arrangement
- Revenue Commitment
- Revenue Commitment Reversal
- Sales Order
- Statistical Journal Entry
- Store Pickup Fulfillment

- Subscription
- Subscription Line
- Time
- Transaction Search
- Transfer Order
- Usage
- Vendor Bill
- Vendor Credit
- Vendor Payment
- Vendor Return Authorization
- Work Order
- Work Order Close
- Work Order Completion
- Work Order Issue

Advanced Intercompany Journal Entry

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

An advanced intercompany journal entry includes all the operations of the original journal entries, in addition to new time-saving functions. For SuiteScript details on the original intercompany journals, see Intercompany Journal Entry.

In the user interface, you can access this record at Transactions > Financial > Make Advanced Intercompany Journal Entries.

If your account has the Multi-Book Accounting feature enabled, you can also work with *book specific* intercompany journal entry records, which in the user interface are available at Transactions > Financial > Make Book Specific Advanced Intercompany Journal Entries. Although they have different entry forms, both book specific and regular intercompany journal entries are the same record type. Within SuiteScript, they are differentiated by the accountingbook field. In other words, a record that has a value set for accountingbook is book specific. Otherwise, the record is not a book-specific intercompany journal entry.

For help working with this record in the user interface, see the help topics Advanced Intercompany Journal Entries and Book-Specific Advanced Intercompany Journal Entries.

The internal ID for this record is `advintercompanyjournalentry`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The intercompany journal entry record is scriptable in both client and server SuiteScript.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

The advanced intercompany journal entry record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Code Sample

The following example shows how to create an advanced intercompany journal entry.

```
var record = nlapiCreateRecord('advintercompanyjournalentry');
record.setFieldValue('subsidiary', 1);

record.selectNewLineItem('line');
record.setCurrentLineItemValue('line', 'subsidiary', 1);
record.setCurrentLineItemValue('line', 'account', 100);
record.setCurrentLineItemValue('line, 'debit', 10);
record.commitLineItem();

record.selectNewLineItem('line');
record.setCurrentLineItemValue('line', 'subsidiary', 1);
record.setCurrentLineItemValue('line', 'account', 101);
record.setCurrentLineItemValue('line, 'credit', 10);
record.commitLineItem();

record.selectNewLineItem('line');
record.setCurrentLineItemValue('line', 'subsidiary', 2);
record.setCurrentLineItemValue('line', 'account', 200);
record.setCurrentLineItemValue('line, 'debit', 10);
record.commitLineItem();

record.selectNewLineItem('line');
record.setCurrentLineItemValue('line', 'subsidiary', 2);
record.setCurrentLineItemValue('line', 'account', 201);
record.setCurrentLineItemValue('line, 'credit', 10);
record.commitLineItem();

nlapiSubmitRecord(record);
```

Assembly Build

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

An assembly item is an inventory item made of several components, but identified as a single item. This type of item lets you define the members of an assembly and to separately track both the component items and the assembled items in inventory. For each assembly build, the assembly item stock level increases and the member items' individual stock levels decrease.

An assembly build transaction records the physical manufacture of an assembly item from component items and the related inventory level changes.

For details about this type of transaction, see Building Assembly Items. For details about this type of item, see the help topic Assembly Items.

The internal ID for this record is `assemblybuild`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Also note that this record has available transforms. See the SuiteScript Records Browser for available transforms. In the NetSuite Help Center, see the help topic nlapiTransformRecord(type, id, transformType, transformValues) for examples on how to transform records.

Supported Script Types

The assembly build record is scriptable in server SuiteScript only.

Supported Functions

The assembly build entry record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Assembly Unbuild

> **Note:** The content in this help topic pertains to all versions of SuiteScript.

An assembly item is an inventory item made of several components, but identified as a single item. This type of item lets you define the members of an assembly and to separately track both the component items and the assembled items in inventory.

An assembly unbuild transaction records the physical taking apart of an assembly item back into its component items and the related inventory level changes. For each assembly unbuild, the assembly item stock level decreases and the member items' individual stock levels increase.

For details about this type of transaction, see Building Assembly Items. For details about this type of item, see the help topic Assembly Items.

The internal ID for this record is `assemblyunbuild`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The assembly unbuild record is scriptable in server SuiteScript only.

Supported Functions

The assembly unbuild entry record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Bin Putaway Worksheet

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Use the Bin Putaway Worksheet to print a list of bin numbers for items that need to be re-stocked in your warehouse or stock room.

Items are added to this list any time an item that uses bins has a greater total on-hand count than the combined on-hand count of its associated bins.

How the worksheet looks and functions depends on whether you use basic Bin Management or use Advanced Bin / Numbered Inventory Management. For details, see the help topics Basic Bin Putaway Worksheet and Advanced Bin Putaway Worksheet.

The internal ID for this record is `binworksheet`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The bin putaway worksheet record can only be scripted in dynamic mode. For details about dynamic scripting, see the following help topics:

- Working with Records in Dynamic Mode
- How do I enable dynamic mode?
- Is dynamic mode better than standard mode?
- Standard vs. Dynamic Mode Code Samples

Note that client (remote object) scripting does not support dynamic scripting.

Supported Functions

The bin putaway worksheet record is partially scriptable — it can be created, deleted, and searched using SuiteScript. It cannot be copied or updated.

Bin Transfer

Note: The content in this help topic pertains to all versions of SuiteScript.

You can record a bin transfer to move items between bins within a warehouse.

For example, at your East Coast location, you stock item #AB123 in two bins: bin #3003 and #4004. There are currently 100 of the item in each bin. You can enter a bin transfer to record the transfer of 50 items out of bin #3003 and into bin #4004.

Recording a bin transfer does not post to your chart of accounts and has no financial impact. The transfer only updates the quantity on hand in each bin for the items transferred.

On the bin transfer record, identify the item, the bin the item will come from, the bin the item will move into and the quantity to be moved. For details, see the help topic Bin Transfers.

The internal ID for this record is `bintransfer`.

The bin transfer record contains a subrecord: Inventory Detail. For details about subrecords in SuiteScript, see the help topic SuiteScript 2.0 Scripting Subrecords.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The bin transfer is scriptable in server SuiteScript only.

Supported Functions

The bin transfer record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Blanket Purchase Order

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

With this record, you can take advantage of fixed pricing for a preset number of items that you will buy during a specific time period. This approach lets you avoid sporadic pricing negotiations with vendors.

This record is available only when the Blanket Purchase Order feature is enabled at Setup > Company > Enable Features, on the Transactions subtab.

In the user interface, you access this record at Transactions > Purchases > Enter Blanket Purchase Order. For help working with this record in the user interface, see the help topic Creating a Blanket Purchase Order.

The internal ID of this record is `blanketpurchaseorder`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The blanket purchase order record is scriptable in both client and server SuiteScript.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

The blanket purchase order record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Cash Refund

Note: The content in this help topic pertains to all versions of SuiteScript.

A cash refund transaction records the return of money to a customer who immediately paid for goods or services using cash, a check or a credit card.

For details about this type of transaction, see Refunding a Cash Sale.

The internal ID for this record is `cashrefund`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The cash refund record is scriptable in both client and server SuiteScript.

Supported Functions

The cash refund record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

See the following sections for more details on working with this record:

- Fields
- Email Operations

The Multi-Partner Management feature must be enabled in your account for the Partners sublist to appear.

Fields

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
Body Fields		
ccnumber	Credit Card #	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in

Field Internal ID	Field UI Label	Note
		beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).
estgrossprofit	Est. Gross Profit	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
estgrossprofitpercent	Est. Gross Profit Percent	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
totalcostestimate	Est. Extended Cost	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search.

Email Operations

NetSuite enables a user to email a copy of a transaction to a customer or another recipient.

If you deploy a beforeLoad user event script on a transaction, in general the script executes when the transaction is emailed. In these cases, the sending of the email is considered an operation of type email on the transaction. However, if the companywide preference **Use Popup for Main Transaction Email Button** is enabled, a beforeLoad script will not execute in certain cases, depending on how the email is sent. That is, the script will not execute if the user views the transaction and selects Actions > Email, or edits the transaction and sends the email by using the Communications tab. In these cases, the system displays a popup window that the user can work with to manually edit and send the email. Actions taken with this window are not considered operations on the transaction. If you have your system configured this way, and you want to deploy a beforeLoad script on emails sent using the popup, deploy the script on the Message record type.

An additional way of emailing a transaction is by editing the transaction and using the Save & Email button. Note that, even if the Use Popup for Main Transaction Email Button preference is enabled, the popup window is not available for emails sent this way. For that reason, a beforeLoad user event script deployed on the transaction will always be triggered by use of the Save & Email button.

For more information on the Use Popup for Main Transaction Email Button preference, see the help topic Setting Email Preferences. For details about emailing transactions, see the help topic Emailing Transactions.

Cash Sale

Note: The content in this help topic pertains to all versions of SuiteScript.

A cash sale is a transaction that records the sale of goods or services for which you receive immediate payment. Enter a cash sale when payment for goods or services has been received at the time of delivery.

Record cash sales by adding each item sold from your Items list. Then, cash sale line-items specify the goods and services sold and their sales amounts. The sum of all sales amounts plus any applicable tax equals the total amount paid for this sale.

For details about this type of transaction, see the help topic Cash Sales.

The internal ID for this record is `cashsale`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The cash sale record is scriptable in both client and server SuiteScript.

Supported Functions

The cash sale record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

See the following sections for more details on working with this record:

- Fields
- Email Operations

The following cash sale record sublists are **list** sublists:

- Billable Expenses
- Billable Items
- Billable Time

In the NetSuite Help Center, see the help topic List Sublists for information on this sublist type.

The Multi-Partner Management feature must be enabled in your account for the Partners sublist to appear.

The Shipping sublist's Shipping Tax Code field, internal ID shippingtaxcode, appears only if per-line taxes have been set on the Item sublist.

Fields

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
Body Fields		
ccnumber	Credit Card #	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).
estgrossprofit	Est. Gross Profit	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.

Field Internal ID	Field UI Label	Note
estgrossprofitpercent	Est. Gross Profit Percent	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
isrecurringpayment	Recurring Payment	A value for this field is stored only if the value for paymentmethod is a credit card.
totalcostestimate	Est. Extended Cost	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search.

Also note that this record has available transforms. See the SuiteScript Records Browser for available transforms. In the NetSuite Help Center, see the help topic nlapiTransformRecord(type, id, transformType, transformValues) for examples on how to transform records.

Email Operations

NetSuite enables a user to email a copy of a transaction to a customer or another recipient.

If you deploy a beforeLoad user event script on a transaction, in general the script executes when the transaction is emailed. In these cases, the sending of the email is considered an operation of type email on the transaction. However, if the companywide preference **Use Popup for Main Transaction Email Button** is enabled, a beforeLoad script will not execute in certain cases, depending on how the email is sent. That is, the script will not execute if the user views the transaction and selects Actions > Email, or edits the transaction and sends the email by using the Communications tab. In these cases, the system displays a popup window that the user can work with to manually edit and send the email. Actions taken with this window are not considered operations on the transaction. If you have your system configured this way, and you want to deploy a beforeLoad script on emails sent using the popup, deploy the script on the Message record type.

An additional way of emailing a transaction is by editing the transaction and using the Save & Email button. Note that, even if the Use Popup for Main Transaction Email Button preference is enabled, the popup window is not available for emails sent this way. For that reason, a beforeLoad user event script deployed on the transaction will always be triggered by use of the Save & Email button.

For more information on the Use Popup for Main Transaction Email Button preference, see the help topic Setting Email Preferences. For details about emailing transactions, see the help topic Emailing Transactions.

Change Order

Note: The content in this help topic pertains to all versions of SuiteScript.

The change order record is used to transact changes to subscriptions. For example, a change order is required to activate a drafted subscription.

The change order record is available only when the SuiteBilling feature is enabled at Setup > Company > Enable Features (Administrator), on the Transactions subtab. When the feature is enabled, you can access a list of change order records in the UI by going to Transactions > Subscriptions > View Subscription Change Orders.

The internal ID for this record is `subscriptionchangeorder`.

See the SuiteScript Records Browser for all internal IDs associated with this record. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The change order record is scriptable in server and client SuiteScript.

Supported Functions

The change order record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Charge

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The charge record is used to represent a single billable amount that a client must pay.

The charge record is available only when the Charge-Based Billing feature is enabled at Setup > Company > Enable Features, on the Transactions subtab. When the feature is enabled, you can access the charge record in the UI by choosing Transactions > Customers > Create Charges > List.

The internal ID for this record is `charge`.

See the SuiteScript Records Browser for all internal IDs associated with this record. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The charge record is scriptable in server SuiteScript only.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

The charge record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Code Sample

The following sample shows how to create a charge record.

```
var charge = nlapiCreateRecord('charge');
      charge.setFieldText('stage', 'Ready for billing');
      charge.setFieldValue('chargetype', 'Time');
      charge.setFieldValue('billto', '43');
      charge.setFieldValue('chargedate', '3/19/2013');
```

```
charge.setFieldValue('salesorder', '122');
charge.setFieldValue('salesorderline', '1');
charge.setFieldValue('currency', '1');
charge.setFieldValue('billingitem', '21');
charge.setFieldValue('timerecord', '2');
charge.setFieldValue('description', 'Charge description');
charge.setFieldValue('rate', '2');
charge.setFieldValue('quantity', '3');
charge.setFieldValue('amount', '5');
recId = nlapiSubmitRecord(charge);
```

Check

Note: The content in this help topic pertains to all versions of SuiteScript.

A check transaction creates and records a check used to pay an expense, records an expense paid in cash and not entered as a bill, or records a non-check debit transaction, such as a debit card transaction, ATM (automated teller machine) transaction, or EFT (electronic funds transfer) payment. A check transaction records an expense directly to your books by debiting the expense account specified in the transaction detail and crediting the bank account the check selected for the check.

For more details about this type of transaction, see the help topic Checking.

The internal ID for this record is `check`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The check record is scriptable in both client and server SuiteScript.

Supported Functions

The check record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
estgrossprofit	Est. Gross Profit	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
estgrossprofitpercent	Est. Gross Profit Percent	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.

Field Internal ID	Field UI Label	Note
totalcostestimate	Est. Extended Cost	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search.
usertotal	Amount	This field is not available via search or lookup for any transactions.

Using Landed Cost Fields

When you create a landed cost category, the associated field IDs for the first category are landedcostamount1 and landedcostsource1. If you create a second category, the IDs will be landedcostamount2 and landedcostsource2.

This pattern increments by one with each additional category. For example, the IDs for the next landed cost category will be landedcostamount3 and landedcostsource3, and so on.

Credit Card Charge

Note: The content in this help topic pertains to all versions of SuiteScript.

The credit card charge record is available only when a credit card account exists in the system. For information about creating a credit card account, see the help topic Creating Accounts.

The internal ID for this record is `creditcardcharge`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The credit card charge record is scriptable in server SuiteScript only.

Supported Functions

The credit card charge record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Usage Notes

The script user must have the Credit Card permission with Full permission level.

Code Samples

The following code snippets show how to create a $30 credit card charge.

```
var recordType = "creditcardcharge";

var record = nlapiCreateRecord(recordType, false)

record.setFieldValue("entity", "4");
```

```
record.setFieldValue("account", "128"); // Account must be created before (Type = Credit Card)
record.setFieldValue("exchangerate", "1.00");
record.setFieldValue("postingperiod", "45");
record.setFieldValue("memo", "MEMO");
record.setFieldValue("class", "1");
record.setFieldValue("department", "1");
record.setFieldValue("location", "1");

record.setLineItemValue("expense", "account", 1, "58");
record.setLineItemValue("expense", "amount", 1, "30.00");
record.setLineItemValue("expense", "memo", 1, "MEMO2");
record.setLineItemValue("expense", "class_", 1, "1");
record.setLineItemValue("expense", "department", 1, "1");
record.setLineItemValue("expense", "location", 1, "1");
record.setLineItemValue("expense", "customer", 1, "3");
record.setLineItemValue("expense", "isbillable", 1, "F");

nlapiSubmitRecord(record);
```

Credit Card Refund

Note: The content in this help topic pertains to all versions of SuiteScript.

The credit card refund record is available only when a credit card account exists in the system. For information about creating a credit card account, see the help topic Creating Accounts.

The internal ID for this record is `creditcardrefund`.

Supported Script Types

The credit card refund record is scriptable in server SuiteScript only.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Functions

The credit card refund record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Usage Notes

The script user must have the Credit Card Refund permission with Full permission level.

Code Samples

The following code snippets show how to create a $30 credit card refund.

```
var recordType = "creditcardrefund";
```

```
var record = nlapiCreateRecord(recordType, false)

record.setFieldValue("entity", "4");
record.setFieldValue("account", "128"); // Account must be created before (Type = Credit Card)
record.setFieldValue("exchangerate", "1.00");
record.setFieldValue("postingperiod", "45");
record.setFieldValue("memo", "MEMO");
record.setFieldValue("class", "1");
record.setFieldValue("department", "1");
record.setFieldValue("location", "1");

record.setLineItemValue("expense", "account", 1, "58");
record.setLineItemValue("expense", "amount", 1, "30.00");
record.setLineItemValue("expense", "memo", 1, "MEMO2");
record.setLineItemValue("expense", "class_", 1, "1");
record.setLineItemValue("expense", "department", 1, "1");
record.setLineItemValue("expense", "location", 1, "1");
record.setLineItemValue("expense", "customer", 1, "3");
record.setLineItemValue("expense", "isbillable", 1, "F");

nlapiSubmitRecord(record);
```

Credit Memo

Note: The content in this help topic pertains to all versions of SuiteScript.

A credit memo transaction decreases the amount a customer owes you. This type of transaction can be used to reverse a charge billed to a customer. If a customer receives a credit memo after having paid an invoice, this memo can be applied to any of the customer's open or future invoices.

For more details about this type of transaction, see the help topic Customer Credit Memos.

The internal ID for this record is `creditmemo`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The credit memo record is scriptable in both client and server SuiteScript.

Supported Functions

The credit card refund record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

NetSuite enables a user to email a copy of a transaction to a customer or another recipient. There are several ways to email a transaction. One way is to open the transaction for viewing and select Actions > Email. Additional approaches for emailing transactions are described in Emailing Transactions.

If you deploy a beforeLoad user event script on a transaction, in general the script executes when the transaction is emailed. In these cases, the sending of the email is considered an operation of type email on the transaction record. However, if the companywide preference Use Popup for Main Transaction Email Button is selected, a beforeLoad script will not execute when a user selects Activity > Email. In this case, the system displays a popup window that the user can work with to manually edit and send the email. Actions taken with this form are not considered operations on the transaction. If you have your system configured this way, and you want to deploy a beforeLoad script on emails sent using the popup, deploy the script on the Message record type.

Note that the Use Popup for Main Transaction Email Button preference does not affect other methods of emailing a transaction. For example, a user may email a transaction by using its Communications subtab or by using the Save and Email button. In these cases, a beforeLoad user event script deployed on the transaction still executes, regardless of how the preference is configured. For more information on the Use Popup for Main Transaction Email Button preference, see the help topic Setting Email Preferences.

The Multi-Partner Management feature must be enabled in your account for the Partners sublist to appear.

Fields

estgrossprofit	Est. Gross Profit	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
estgrossprofitpercent	Est. Gross Profit Percent	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
totalcostestimate	Est. Extended Cost	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search.

Customer Deposit

Note: The content in this help topic pertains to all versions of SuiteScript.

A customer deposit transaction records the funds received when a customer makes an advance payment for an order. This payment is recorded in the general ledger as a liability until the goods or services are actually delivered, and does not affect the customer's accounts receivable balance. After the order is filled, the deposit is applied against the invoice.

For more details about this type of transaction, see Customer Deposits.

The internal ID for this record is `customerdeposit`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The customer deposit record is scriptable in both client and server SuiteScript.

Supported Functions

The customer deposit record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
Body Fields		
ccnumber	Credit Card #	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).
isrecurringpayment	Recurring Payment	A value for this field is stored only if the value for paymentmethod is a credit card.

Customer Payment

Note: The content in this help topic pertains to all versions of SuiteScript.

When a customer makes a payment, record the payment and apply it to the appropriate invoice or cash sale. Applying a payment decreases the amount due and tracks income.

For details about this type of transaction, see the help topic Customer Payments.

The internal ID for this record is `customerpayment`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Important: When the apply sublist appears on the customer payment record, the Apply subtab is visible, however, all apply sublist data appears on a subtab called Invoice. Also, note that the UI label for the **Amt. Due** (due) field can also appear as **Amount Remaining**.

Supported Script Types

The customer payment record is scriptable in server SuiteScript only.

Supported Functions

The customer payment record is partially scriptable — it can be updated, deleted, and searched using SuiteScript. It cannot be created or copied.

Usage Notes

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
Body Fields		
ccnumber	Credit Card #	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).
isrecurringpayment	Recurring Payment	A value for this field is stored only if the value for paymentmethod is a credit card.

The Deposits sublist is a **list** sublist. In the NetSuite Help Center, see the help topic List Sublists for information on this sublist type.

Customer Refund

Note: The content in this help topic pertains to all versions of SuiteScript.

The internal ID for this record is `customerrefund`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The customer refund record is scriptable in server SuiteScript only.

Supported Functions

The customer deposit record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
Body Fields		
ccnumber	Credit Card #	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).
estgrossprofit	Est. Gross Profit	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.

Field Internal ID	Field UI Label	Note
estgrossprofitpercent	Est. Gross Profit Percent	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
totalcostestimate	Est. Extended Cost	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search.

The Deposits sublist is a **list** sublist. In the NetSuite Help Center, see the help topic List Sublists for information on this sublist type.

Deposit

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

You use the deposit record to adjust the balance of an account.

In the UI, you access this record at Transactions > Bank > Make Deposits. For details on working with deposits, see the help topic Deposits.

The internal ID of this record is `deposit`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The deposit record is scriptable in both client and server SuiteScript.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

The deposit record cannot be copied, but otherwise it is fully scriptable. It can be created, updated, deleted, and searched in SuiteScript.

Usage Notes

Be aware of the following:

- To successfully add a new deposit record, you must include at least one line in one of the following three sublists: Payments, Other Deposits, or Cash Back.
- If you use the Multi-Book Accounting feature, be aware that the Accounting Book Detail sublist is read-only and not scriptable.

Code Samples in SuiteScript 1.0

The following samples show how to create, load, and delete deposit records in SuiteScript 1.0.

```
function createBankDeposit(cashSaleId){
    var rec = nlapiCreateRecord("deposit");
    rec.setFieldValue("account","1");
    rec.setFieldValue("subsidiary","1");
    rec.setLineItemValue("payment","id",1,cashSaleId);
    rec.setLineItemValue("payment","deposit",1,"T");
    var recId = nlapiSubmitRecord(rec);
    return recId;
}

function testDeposit()
{
    var cashSaleId = null;
    var depositId = null;
    try{
        cashSaleId = createCashSale(); // or just type internalId of valid Cash Sale
        depositId = createBankDeposit(cashSaleId);
        var deposit = nlapiLoadRecord("deposit",depositId);
    }
    finally{
        if(depositId!=null)
            nlapiDeleteRecord("deposit",depositId);
        if(cashSaleId!= null)
            nlapiDeleteRecord("cashSale",cashSaleId);
    }
}
```

Disabling or Enabling User Filters

When you create or load a deposit record, you can control whether the script respects the filter preferences set by the user on the record's Deposits > Payments subtab. You control this behavior by using the `disablepaymentfilters` initialization flag.

Set the parameter to `true` to ignore the preferences and disable filters, so that all the payments are loaded.

Set the parameter to `false` to indicate that the system should respect the user's preferences and enable the filters. In this case, filters are applied and fewer payments are loaded. If you have a large number of payments, you can set `disablepaymentfilters` to `false` to improve performance. If you do not set a value for the `disablepaymentfilters` parameter, it defaults to false.

The exact way you work with `disablepaymentfilters` varies slightly depending on which version of SuiteScript you are using, as described in the following sections.

Disabling Filters in SuiteScript 2.0

In SuiteScript 2.0, you can interact with `disablepaymentfilters` when you use the defaultValues parameter, which is available for the record.create(options) and record.load(options).

For example, the following SuiteScript 2.0 snippet would ignore the user's preferences:

```
var newRecord = record.create({
    type: record.Type.DEPOSIT,
    defaultValues: {
       disablepaymentfilters: true
```

```
        deposits: true
      }
})
```

Disabling Filters in SuiteScript 1.0

In SuiteScript 1.0, you use the `disablepaymentfilters` parameter, which is available for nlapiCreateRecord(type, initializeValues) and nlapiLoadRecord(type, id, initializeValues). For example, the following SuiteScript 1.0 snippet would ignore the user's preferences:

```
var newRecord = nlapiCreateRecord('deposit', {account:'2', disablepaymentfilters: 'true'} );
```

Specifying Payment IDs

When you create a deposit record, you can choose to specify payment IDs. You control this behavior by using the `deposits` initialization parameter.

Specifying Payment IDs in SuiteScript 2.0

To specify payment IDs in SuiteScript 2.0, use the `deposits` initialization parameter, which is available for record.create(options). The `deposits` parameter is not available for record.load(options).

For example, the following SuiteScript 2.0 snippet would select the payment IDs of 818 and 819:

```
var myDeposit = record.create({
    type: record.Type.DEPOSIT,
    defaultValues: {
          'deposits': '818,819'
     }
})
```

Specifying Payment IDs in SuiteScript 1.0

To specify payment IDs in SuiteScript 1.0, use the `deposits` initialization parameter, which is available for nlapiCreateRecord(type, initializeValues). The `deposits` parameter is not available for nlapiLoadRecord(type, id, initializeValues).

For example, the following SuiteScript 1.0 snippet would select the payment IDs of 818 and 819:

```
var newRecord = nlapiCreateRecord('deposit', {'recordmode': 'dynamic', 'account': '1', 'deposit
s': '818,819'} );
```

Deposit Application

Note: The content in this help topic pertains to all versions of SuiteScript.

A deposit application transaction applies a customer deposit against an invoice after the order is complete.

For details about this type of transaction, see the help topic Applying a Customer Deposit.

The internal ID for this record is `depositapplication`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The deposit application record is scriptable in both client and server SuiteScript.

Supported Functions

The deposit application record cannot be created, but otherwise it is fully scriptable. It can be copied, updated, deleted, and searched in SuiteScript.

Usage Notes

In SuiteScript 1.0, you do not use the nlapiCreateRecord(...) to create a deposit application record. Deposit applications are always created when a customer ceposit is applied to an invoice. The application can only be created by applying an open customer Deposit from the deposit sublist of the customer payment. On submit, the backend creates a deposit application in the amount applied.

You can use the **doc** field on the apply sublist of the customer payment to get the internal ID of the deposit or invoice.

Estimate / Quote

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The internal ID for this record is `estimate`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The estimate / quote record is scriptable in both client and server SuiteScript.

Supported Functions

The estimate / quote record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Usage Notes

See the following sections for more details on working with this record:

- Fields
- Email Operations

Fields

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
Body Fields		
estgrossprofit	Est. Gross Profit	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
estgrossprofitpercent	Est. Gross Profit Percent	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
totalcostestimate	Est. Extended Cost	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search.

Also note that this record has available transforms. See the SuiteScript Records Browser for available transforms. In the NetSuite Help Center, see the help topic nlapiTransformRecord(type, id, transformType, transformValues) for examples on how to transform records.

The Multi-Partner Management feature must be enabled in your account for the Partners sublist to appear.

The Shipping sublist's Shipping Tax Code field, internal ID shippingtaxcode, appears only if per-line taxes have been set on the Item sublist.

Email Operations

NetSuite enables a user to email a copy of a transaction to a customer or another recipient.

If you deploy a beforeLoad user event script on a transaction, in general the script executes when the transaction is emailed. In these cases, the sending of the email is considered an operation of type email on the transaction. However, if the companywide preference **Use Popup for Main Transaction Email Button** is enabled, a beforeLoad script will not execute in certain cases, depending on how the email is sent. That is, the script will not execute if the user views the transaction and selects Actions > Email, or edits the transaction and sends the email by using the Communications tab. In these cases, the system displays a popup window that the user can work with to manually edit and send the email. Actions taken with this window are not considered operations on the transaction. If you have your system configured this way, and you want to deploy a beforeLoad script on emails sent using the popup, deploy the script on the Message record type.

An additional way of emailing a transaction is by editing the transaction and using the Save & Email button. Note that, even if the Use Popup for Main Transaction Email Button preference is enabled, the popup window is not available for emails sent this way. For that reason, a beforeLoad user event script deployed on the transaction will always be triggered by use of the Save & Email button.

For more information on the Use Popup for Main Transaction Email Button preference, see the help topic Setting Email Preferences. For details about emailing transactions, see the help topic Emailing Transactions.

Expense Report

Note: The content in this help topic pertains to all versions of SuiteScript.

An expense report transaction records an employee's expenses for approval and conversion into a bill. The expense total remains in an unapproved expense account and has no accounting impact until the expense is approved by someone with accounting authority. After an expense report is approved, a bill is created and the expense amount is reflected on the books.

This transaction is available when the Estimates feature is enabled at Setup > Company > Enable Features, on the Employees subtab.

For more details about this type of transaction, see the help topic Expense Reporting.

The internal ID for this record is `expensereport`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The expense report record is scriptable in both client and server SuiteScript.

Supported Functions

The expense report record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Fulfillment Request

Note: The content in this help topic pertains to all versions of SuiteScript.

The fulfillment request record represents a request or demand to fulfill a sales order at a fulfillment location. You create a fulfillment request record from a sales order record.

To use this record, you must have the Fulfillment Request feature enabled. See the help topic Fulfillment Requests for more information about enabling and configuring the feature in NetSuite.

The internal ID for this record is `fulfillmentrequest`.

Supported Script Types

The fulfillment request record is scriptable in both client and server SuiteScript.

Supported Functions

This record supports the following functions: read, edit, transform, delete, and search.

Note: The create and copy functions are not supported.

Usage Notes

To create a fulfillment request record, you must have a sales order ID. The Fulfillment Choice and Location must also be set on each line in the sales order. Depending on the selected fulfillment choice (Ship or Store Pickup), NetSuite sets the fulfillment request type to Ship or Store Pickup respectively. If there are multiple lines on the sales order with different fulfillment choices or locations, you must create separate fulfillment requests for each location/fulfillment choice combination. If all lines on the sales order have the same fulfillment choice and location, you create a single fulfillment request. Trying to create a fulfillment request with different fulfillment choices or different locations results in an exception.

For example, if a sales order has two lines, with the fulfillment choice on the first line set to Ship, and the fulfillment choice on the second line set to Store Pickup, you need to create two fulfillment request records.

Note: When you automate the creation of fulfillment requests in your account, NetSuite creates the required number of fulfillment requests automatically.

Fulfillment requests have one of several statuses. Only some statuses can be set with SuiteScript. Other statuses are set by NetSuite and are based on the fulfillment state of the line items in the fulfillment request. The following table lists the possible statuses of the fulfillment request record. See the help topic Fulfillment Request Statuses for more information about fulfillment request statuses.

Status Code	Status Name	Settable
A	New	Yes
B	In Progress	Yes
C	Cancelled	Yes
D	Picked	No
I	Packed	No
E	Partially Fulfilled	No
G	Fulfilled with Exceptions	No
F	Fulfilled	No
R	Rejected	No

You can add fulfillment request exceptions at the line item level to a fulfillment request. To create a fulfillment request exception, you need to supply a quantity, an exception type, and an exception reason. The exception reasons are different for each exception type. The following table lists the exception types and reasons.

Exception Type ID	Exception Type	Exception Reason ID	Exception Reason
1	Picking	-1	Inventory mismatch
		-2	Physical damage
		-3	Other
2	Packing	-4	Packing material shortage

Exception Type ID	Exception Type	Exception Reason ID	Exception Reason
		-5	Other
3	Picking up	-6	Too heavy
		-7	Too big
		-8	Wrong color
		-9	Other
4	Shipping	-10	Shipper can't pick up
		-11	Shipper did not come
		-12	Shipping address issues
		-13	Other

Code Samples

Example 1: Create a fulfillment request from a sales order. The following example shows how to create a fulfillment request from a sales order. The sales order ID is 321654. You use the nlapiTransformRecord API to transform the sales order record to a fulfillment request.

```
var fulfillmentRequestRecord = nlapiTransformRecord("salesorder", 321654, "fulfillmentrequest",
 true);

var fulfillmentRequestRecordId = nlapiSubmitRecord(fulfillmentRequestRecord);
```

Example 2: Add a fulfillment exception on a fulfillment request. This example shows how to add one fulfillment exception reason to an existing fulfillment request. The fulfillment request ID is 221100. The fulfillment exception is for the second line item in the fulfillment request.

```
var fulfillmentRequestRecord = nlapiLoadRecord("fulfillmentrequest", "221100", true);

fulfillmentRequestRecord.selectNewLineItem('fulfillmentexception')

fulfillmentRequestRecord.setCurrentLineItemValue("fulfillmentexception", "itemline", 2);
fulfillmentRequestRecord.setCurrentLineItemValue("fulfillmentexception", "exceptionquantity", 1
);
fulfillmentRequestRecord.setCurrentLineItemValue("fulfillmentexception", "exceptiontype", 1);
fulfillmentRequestRecord.setCurrentLineItemValue("fulfillmentexception", "exceptionreason", -1)
;
fulfillmentRequestRecord.setCurrentLineItemValue("fulfillmentexception", "exceptioncomments", f
ulfillmentRequestExceptionLine.getExceptionComments());
fulfillmentRequestRecord.commitLineItem("fulfillmentexception");

var fulfillmentRequestRecordId = nlapiSubmitRecord(fulfillmentRequestRecord);
```

Example 3: Create fulfillment requests on a sales order with different fulfillment choices. If a sales order has different fulfillment choices or locations at the line level, you first need to get the values of the fulfillment choice and location fields on each line. As you loop through lines in the fulfillment request, you check whether the values on the line are the same as the values on the first line; if they are different, you discard the line.

```
var salesOrderID = 321654;
```

```
var fulfillmentRequestRecord = nlapiTransformRecord("salesorder", salesOrderID, "fulfillmentreq
uest", {recordmode : 'dynamic'});
var fulfillmentRequestLinesCount = fulfillmentRequestRecord.getLineItemCount("item");

while (fulfillmentRequestLinesCount > 0)
{
	var initialLocation = null;
	var initialItemFulfillmentChoice = null;

	for (line = 1; line <= fulfillmentRequestLinesCount; line++)
	{
		var currentLocation = fulfillmentRequestRecord.getLineItemValue("item", "location
", line);
		var currentItemFulfillmentChoice = fulfillmentRequestRecord.getLineItemValue("ite
m", "itemfulfillmentchoice", line);

		if (initialLocation == null)
		{
			initialLocation = currentLocation;
			initialItemFulfillmentChoice = currentItemFulfillmentChoice;
		}

		if ((initialLocation == currentLocation) && (initialItemFulfillmentChoice == curr
entItemFulfillmentChoice))
		{
			fulfillmentRequestRecord.setLineItemValue("item", "itemreceive", line, "T"
);
		}
		else
		{
			// Discard the line because its location or fulfillment choice is differen
t
			fulfillmentRequestRecord.setLineItemValue("item", "itemreceive", line, "F"
);
		}
	}

	nlapiSubmitRecord(fulfillmentRequestRecord);
	fulfillmentRequestRecord = nlapiTransformRecord("salesorder", salesOrderID, "fulfillment
request", {recordmode : 'dynamic'});
	fulfillmentRequestLinesCount = fulfillmentRequestRecord.getLineItemCount("item");
}
```

Intercompany Journal Entry

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Intercompany journal entries are a specialized type of journal available specifically for OneWorld. An intercompany journal entry records debits and credits to be posted to ledger accounts for transactions between two subsidiaries. These records adjust the value of any set of accounts without the need for transactions such as invoices and bills.

In the user interface, you can access this record at Transactions > Financial > Make Intercompany Journal Entries.

If your account has the Multi-Book Accounting feature enabled, you can also work with *book specific* intercompany journal entry records, which in the user interface are available at Transactions > Financial > Make Book Specific Intercompany Journal Entries. Although they have different entry forms, both book specific and regular intercompany journal entries are the same record type. Within SuiteScript, they are differentiated by the accountingbook field. In other words, a record that has a value set for accountingbook is book specific. Otherwise, the record is a regular intercompany journal entry.

For help working with this record in the user interface, see the help topics Intercompany Journal Entries and Book-Specific Intercompany Journal Entries.

The internal ID for this record is `intercompanyjournalentry`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The intercompany journal entry record is scriptable in both client and server SuiteScript.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

The intercompany journal entry record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Code Sample

The following example shows how to create a book specific intercompany journal entry. The record is book specific because a value has been set for the accountingbook field.

```
var rec = nlapiCreateRecord('intercompanyjournalentry');
rec.setFieldValue('accountingbook', 2); // Setting a value for this field makes the record book
 specific.
rec.setFieldValue('subsidiary', 1);
rec.setFieldValue('tosubsidiary', 3);

rec.selectNewLineItem('line');
rec.setCurrentLineItemValue('line', 'linesubsidiary', 1);
rec.setCurrentLineItemValue('line', 'account', 1);
rec.setCurrentLineItemValue('line', 'credit', '2.00');
rec.commitLineItem('line');
rec.selectNewLineItem('line');
rec.setCurrentLineItemValue('line', 'linesubsidiary', 1);
rec.setCurrentLineItemValue('line', 'account', 2);
rec.setCurrentLineItemValue('line', 'debit', '2.00');
rec.commitLineItem('line');

rec.selectNewLineItem('line');
```

```
rec.setCurrentLineItemValue('line', 'linesubsidiary', 3);
rec.setCurrentLineItemValue('line', 'account', 6);
rec.setCurrentLineItemValue('line', 'credit', '2.00');
rec.commitLineItem('line');
rec.selectNewLineItem('line');
rec.setCurrentLineItemValue('line', 'linesubsidiary', 3);
rec.setCurrentLineItemValue('line', 'account', 149);
rec.setCurrentLineItemValue('line', 'debit', '2.00');
rec.commitLineItem('line');

var id = nlapiSubmitRecord(rec);
;
```

Intercompany Transfer Order

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

In accounts using NetSuite OneWorld and the Multi-Location Inventory(MLI) feature, you can use the Intercompany Transfer Order transaction to move inventory from a location for one subsidiary to a location for another subsidiary.

In the user interface, you can access this record at Transactions > Inventory > Enter Intercompany Transfer Orders..

For help working with this record in the user interface, see the help topic Intercompany Inventory Transfers - Non-Arm's Length.

The internal ID for this record is `intercompanytransferorder`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The intercompany transfer order record is scriptable in both client and server SuiteScript.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

The intercompany transfer order record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Usage Notes

Intercompany transfer orders are only available when transfer pricing is used, meaning the Use Item Cost as Transfer Cost, at Setup > Accounting > Preferences > Accounting Preferences (Administrator), on the Order Management subtab, must be disabled.

This record has available transforms. See the SuiteScript Records Browser for available transforms. In the NetSuite Help Center, see the help topic nlapiTransformRecord(type, id, transformType, transformValues) for examples on how to transform records.

Code Sample

The following example shows how to create an intercompany transfer order.

```
...
NLRecordObject incoTransfer = nlapiCreateRecord('intercompanytransferorder', true);
incoTransfer.setFieldValue('orderstatus', expectedValues.get(orderstatus));
incoTransfer.setFieldValue('subsidiary', expectedValues.get(subsidiary));
incoTransfer.setFieldValue('tosubsidiary', expectedValues.get(tosubsidiary));
incoTransfer.setFieldText('location', sourceLocationName1);
incoTransfer.setFieldText('transferlocation', destinationLocationName1);
incoTransfer.selectNewLineItem('item');
incoTransfer.setCurrentLineItemValue('item', 'Item.item', itemKey);
incoTransfer.setCurrentLineItemValue('item', 'Item.rate', rate);
incoTransfer.setCurrentLineItemValue('item','quantity', quantity);
incoTransfer.commitLineItem('item');
String key = nlapiSubmitRecord(incoTransfer);
...
```

Inventory Adjustment

Note: The content in this help topic pertains to all versions of SuiteScript.

The inventory adjustment transaction changes the quantity and value of an inventory item without entering a purchase order. For example, this transaction can be used to account for clerical errors, changes in cost, thefts, or miscounts. If you use the LIFO or FIFO costing methods, this transaction preserves the quantity and value of an inventory item at the same time preserving the costing history of the item.

For more details about this type of transaction, see the help topic Adjusting Inventory.

The internal ID for this record is `inventoryadjustment`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The inventory adjustment record is scriptable in both client and server SuiteScript.

Supported Functions

The inventory adjustment record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Inventory Cost Revaluation

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The inventory cost revaluation record is used to recalculate the value of items configured to use standard costing.

This record is available when the Standard Costing feature is enabled at Setup > Company > Enable Features, on the Items & Inventory subtab. When the feature is enabled, you can access the inventory cost revaluation record in the UI by choosing Transactions > Inventory > Revalue Inventory Cost.

The internal ID for this record is `inventorycostrevaluation`.

See the SuiteScript Records Browser for all internal IDs associated with this record. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The inventory cost revaluation record is scriptable in both client SuiteScript and Server SuiteScript.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

The inventory cost revaluation record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Code Sample

The following sample shows how to create and update inventory cost revaluation records.

```
function inventoryCostRevaluation(){
   var recId = null;
   var newId = null;
   var recType = 'INVENTORYCOSTREVALUATION';
   try{

      // Add
      var revaluationRec = nlapiCreateRecord(recType);
      revaluationRec.setFieldValue('subsidiary','1');
      revaluationRec.setFieldValue('item',itemId1);    // Some assembly item
      revaluationRec.setFieldValue('account','1');
      revaluationRec.setFieldValue('location','1');
      revaluationRec.setLineItemValue('costcomponent', 'cost', 1, '2');
      revaluationRec.setLineItemValue('costcomponent', 'componentitem', 1, componentItemId1);
 // Some inv. item
      revaluationRec.setLineItemValue('costcomponent', 'quantity', 1, '3');

      recId = nlapiSubmitRecord(revaluationRec);
```

```
      var revaluationAddedRec = nlapiLoadRecord(recType, recId);


      // Update
      // Note you cannot change subsidiary and item
      revaluationAddedRec.setFieldValue('account','2');
      revaluationAddedRec.setLineItemValue('costcomponent', 'quantity', 1, '5');
      nlapiSubmitRecord(revaluationAddedRec);

      var revaluationUpdatedRec = nlapiLoadRecord(recType, recId2);

      // Copy
      var copiedRecord = nlapiCopyRecord(recType, recId);
      newId = nlapiSubmitRecord(copiedRecord);
      copiedRecord = nlapiLoadRecord(recType,newId);

      //Search columns
      var arrSearchColumns = new Array();
      arrSearchColumns[0] = new nlobjSearchColumn('memo');
      arrSearchColumns[1] = new nlobjSearchColumn('subsidiary');

      // Filters
      var arrSearchFilters = new Array();
      arrSearchFilters[0] = new nlobjSearchFilter('memo', null,'contains', 'em');

      //Search
      var arrSearchResults = nlapiSearchRecord(recType,null,arrSearchFilters,arrSearchColumns);



      nlapiLogExecution('DEBUG', ''+arguments.callee.name +' passed.');
   }
   catch (e) {
      nlapiLogExecution('ERROR', ''+arguments.callee.name +' failed! ',e);

   }
   finally{
      if(recId!=null){
         nlapiDeleteRecord(recType, recId);
      }
   if(newId!=null){
         nlapiDeleteRecord(recType, newId);
      }
   }
}
```

Inventory Count

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The inventory count record enables you to maintain better inventory accuracy and tighter control of assets.

The inventory count record is available only when the Inventory Count feature is enabled at Setup > Enable Features, on the Items & Inventory subtab. For an overview of the feature, see the help topic Inventory Count.

In the user interface, you access the inventory count record at Transactions > Inventory > Enter Inventory Count.

The internal ID for this record is `inventorycount`.

See the SuiteScript Records Browser for all internal IDs associated with this record. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The inventory count record is scriptable in both client and server SuiteScript.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

The inventory count record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

You cannot use SuiteScript to update the Status field on inventory count records. You must set this field by clicking the Start Count button manually on the inventory count record in the user interface.

Code Samples

Creating an Inventory Count Record Using SuiteScript 1.0

The following example shows how to create an inventory count record using SuiteScript 1.0.

```
var REC_TYPE = 'inventorycount';
var inventoryCount = nlapiCreateRecord(REC_TYPE);

inventoryCount.setFieldValue('subsidiary', 1);
inventoryCount.setFieldValue('location', 1);
inventoryCount.setFieldValue('tranid', '5');
inventoryCount.setFieldValue('trandate', '11/20/2013');
inventoryCount.setFieldValue('account', 'Checking');
inventoryCount.setFieldValue('department', 'Department US');
inventoryCount.setFieldValue('class', 'Class US');
inventoryCount.setFieldValue('memo', 'SS memo');

inventoryCount.selectNewLineItem('item');
inventoryCount.setCurrentLineItemValue('item', 'item', 247);
inventoryCount.setCurrentLineItemValue('item', 'rate', 69.5);
inventoryCount.setCurrentLineItemValue('item', 'memo', 'Line memo');
inventoryCount.commitLineItem('item');
```

```
var inventoryCountId = nlapiSubmitRecord(inventoryCount);
```

Inventory Detail

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The inventory detail subrecord is available when the Advanced Bin / Numbered Inventory Management feature is enabled. This type of record stores values relating to bin numbers and/or serial/lot numbers for items, including line items on transactions. This data includes the quantity on hand and quantity available per bin number, or per serial/lot number, or when both are in use, per bin number/serial lot number combination.

In the UI, the inventory detail subrecord displays as a popup when you click the Inventory Detail body field or sublist field.

For more details, see the help topic Advanced Bin / Numbered Inventory Management.

The internal ID for this subrecord is `inventorydetail`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The inventory detail record is scriptable in client SuiteScript. Server-side scripts must access through the parent record.

Supported Functions

The inventory detail record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

Inventory Detail is scriptable from both the body field and the line item.

Inventory Detail is considered a subrecord, represented by the nlobjSubrecord object in SuiteScript. For details on working with this subrecord type, see the help topic Scripting the Inventory Detail Subrecord. For general details on working with subrecords, see the help topic Working with Subrecords in SuiteScript.

For information on adding an inventory detail subrecord to an inventory count record, see *Creating an Inventory Count Record Using SuiteScript 2.0*.

Inventory Status Change

Note: The content in this help topic pertains to all versions of SuiteScript.

The inventory status change record is available in NetSuite and NetSuite OneWorld accounts when the Inventory Status feature is enabled. An administrator can enable this feature at Setup > Company >

EnableFeatures, on the Items & Inventory subtab. For more information, see the help topic Inventory Status.

The internal ID for the inventory status change record is `inventorystatuschange`. The internal ID for its Items sublist is `inventory`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The inventory status change record is scriptable in both client and server SuiteScript.

Supported Functions

The inventory status change record can be created, read, edited, copied, deleted, searched, and transformed using SuiteScript.

Usage Notes

In the UI, you cannot select the same value in the Previous State and Revised State fields. Using SuiteScript, there is no validation if you set the same value for `previousstatus` and `revisedstatus`.

Code Sample

The following code snippet illustrates how to create an inventory status change record to move 10 good widgets at your US location to the inspection status.

```
var record = nlapiCreateRecord('inventorystatuschange');
record.setFieldText('location', 'US Only Location');
record.setFieldText('previousstatus', 'Good');
record.setFieldText('revisedstatus', 'Inspection');
record.selectNewLineItem('inventory');
record.setCurrentLineItemText('inventory', 'item', 'Widget');
record.setCurrentLineItemValue('inventory', 'quantity', 10);
record.commit('inventory');
var recordId = nlapiSubmitRecord(record);
```

Inventory Transfer

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The Inventory Transfer transaction posts details about per-location item inventory level changes when items are transferred between two locations. This basic inventory transfer decreases items in the source location and increases them in the receiving location, all in one step.

This transaction is available when the Locations feature and the Multi-Location Inventory (MLI) feature are enabled. For more details, see the help topic Transferring Inventory.

The internal ID for this record is `inventorytransfer`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The inventory transfer record is scriptable in server SuiteScript only.

Supported Functions

The inventory detail record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Usage Notes

This record has available transforms. See the SuiteScript Records Browser for available transforms. In the NetSuite Help Center, see the help topic nlapiTransformRecord(type, id, transformType, transformValues) for examples on how to transform records in SuiteScript 1.0.

Invoice

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The internal ID for this record is `invoice`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The invoice record is scriptable in both client and server SuiteScript.

Supported Functions

The invoice record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Usage Notes

See the following sections for more details on working with this record:

- Fields
- Email Operations

Fields

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
Body Fields		
estgrossprofit	Est. Gross Profit	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
estgrossprofitpercent	Est. Gross Profit Percent	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
totalcostestimate	Est. Extended Cost	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search.

This record has available transforms. See the SuiteScript Records Browser for available transforms. In the NetSuite Help Center, see the help topic nlapiTransformRecord(type, id, transformType, transformValues) for examples on how to transform records in SuiteScript 1.0.

The following invoice record sublists are **list** sublists:

- Billable Expenses
- Billable Items
- Billable Time

In the NetSuite Help Center, see the help topic List Sublists for information on this sublist type.

The Multi-Partner Management feature must be enabled in your account for the Partners sublist to appear.

The Shipping sublist's Shipping Tax Code field, internal ID shippingtaxcode, appears only if per-line taxes have been set on the Item sublist.

Email Operations

NetSuite enables a user to email a copy of a transaction to a customer or another recipient.

If you deploy a beforeLoad user event script on a transaction, in general the script executes when the transaction is emailed. In these cases, the sending of the email is considered an operation of type email on the transaction. However, if the companywide preference **Use Popup for Main Transaction Email Button** is enabled, a beforeLoad script will not execute in certain cases, depending on how the email is sent. That is, the script will not execute if the user views the transaction and selects Actions > Email, or edits the transaction and sends the email by using the Communications tab. In these cases, the system displays a popup window that the user can work with to manually edit and send the email. Actions taken with this window are not considered operations on the transaction. If you have your system configured this way, and you want to deploy a beforeLoad script on emails sent using the popup, deploy the script on the Message record type.

An additional way of emailing a transaction is by editing the transaction and using the Save & Email button. Note that, even if the Use Popup for Main Transaction Email Button preference is enabled, the popup window is not available for emails sent this way. For that reason, a beforeLoad user event script deployed on the transaction will always be triggered by use of the Save & Email button.

For more information on the Use Popup for Main Transaction Email Button preference, see the help topic Setting Email Preferences. For details about emailing transactions, see the help topic Emailing Transactions.

Item Demand Plan

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The item demand plan record is available for inventory items when the Demand Planning feature is enabled, and for assembly/BOM items when both the Demand Planning and Work Orders features are enabled. Item demand plans can be added for items where the supplyReplenishMethod field is set to Time Phased.

An item demand plan transaction stores the quantity expected to be needed, during specified time periods, for an item. NetSuite supports three types of demand plans: monthly, weekly, and daily.

For more information about using item demand plans, see the help topic Demand Planning.

The internal ID for this record is `itemdemandplan`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The item demand plan record is scriptable in both client and server SuiteScript.

Supported Functions

The invoice record is partially scriptable — it can be created, updated, deleted, and searched using SuiteScript. It cannot be copied or transformed.

Usage Notes

A demand plan records the expected future demand for an item based on previous or projected demand. When the Demand Planning feature is enabled, demand plans can be created for inventory items. When the Work Orders feature is also enabled, demand plans also can be created for assembly/BOM items. Demand plans can only be created for items that have a value of “Time Phased” for the **supplyreplenishmethod** field.

Each demand plan record includes:

- A set of body fields used to uniquely identify the demand plan, define the time period it covers, and indicate the time period it uses (monthly, weekly, or daily).

  - Body fields must be defined before matrix field values can be edited.
- A matrix of projected quantities per time period, similar to the matrix used for item pricing.
  - In a monthly demand plan, this matrix includes a row for each month in the time period, and one column with the projected quantity demand for each month.
  - In a weekly demand plan, this matrix includes a row for each week in the time period, and one column with the projected quantity demand for each week.
  - In a daily demand plan, this matrix includes a row for each week in the time period and seven columns with the projected quantity demand for each day of each week.
- Review the following table for details about Item Demand Plan body and matrix sublist fields. For more details and code samples, see Demand Plan Detail Sublist.

Field Internal ID	Field UI Label	Note
Body Fields		
subsidiary	Subsidiary	Required in OneWorld accounts.
location	Location	Required when the Multi-Location Inventory feature enabled.
item	Item	Required. Can only use items with supplyreplenishment method set to Time Phased.
units	Unit of Measure	Optional. Available when the Multiple Units of Measure feature is enabled.
memo	Memo	Optional.
startdate	Start Date	Optional. Defaults to the first day of the current year, for example for 2011, defaults to 1/1/2011.
enddate	End Date	Optional. Defaults to the last day of the current year, for example for 2011, defaults to 12/31/2011.
demandplancalendartype	View	Required. Valid values are MONTHLY, WEEKLY, or DAILY. (Must use all capital letters.)
Matrix Fields		
quantity	Quantity	▪ For monthly and weekly demand plans, each row has one quantity column. ▪ For daily demand plans, each row has seven quantity columns.
startdate	Start Date	System-calculated, read-only values. ▪ For monthly plans, the date of the first day of the month that the row represents. ▪ For weekly and daily plans, the date of the first day of the week that the row represents, based on the preference set for First Day of Week at Setup > Company > General Preferences.
enddate	End Date	System-calculated, read-only values. ▪ For monthly plans, the date of the last day of the month that the row represents.

Field Internal ID	Field UI Label	Note
		▪ For weekly and daily plans, the date of the last(seventh) day of the week that the row represents, based on the preference set for First Day of Week at Setup > Company > General Preferences.

Note: It is recommended that you work with the Item Demand Plan record in dynamic mode. See the help topic Working with Records in Dynamic Mode.

Demand Plan Detail Sublist

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The internal ID for this sublist is **demandplandetail**.

The Demand Plan Detail sublist appears on the Item Demand Plan record type.

The Demand Plan Detail sublist is a matrix that is similar to the Pricing sublist. This matrix stores projected quantities demanded by date. Each row in the matrix represents a specific month, week, or day, and each column in the matrix represents an expected quantity demand.

Functionally, this sublist shares many of the characteristics of List Sublists. However, scripting with the Demand Plan Detail sublist is not like scripting with most other sublists in NetSuite. You must use Matrix APIs for the Demand Plan Detail Sublist to access quantity values on a per-row, per-column basis, similar to the way that item pricing values are accessed. These APIs are a subset of the Sublist APIs more commonly used for scripting with other sublists.

The format of the Demand Plan Detail sublist depends on the values set in body fields for the start date of the plan, the end date of the plan, and the time period to be used (monthly, weekly, or daily). Because of this dependence, it is recommended that you work with the Item Demand Plan record and the Demand Plan Detail sublist in dynamic mode. See the help topic Working with Records in Dynamic Mode.

Be aware of the following requirements:

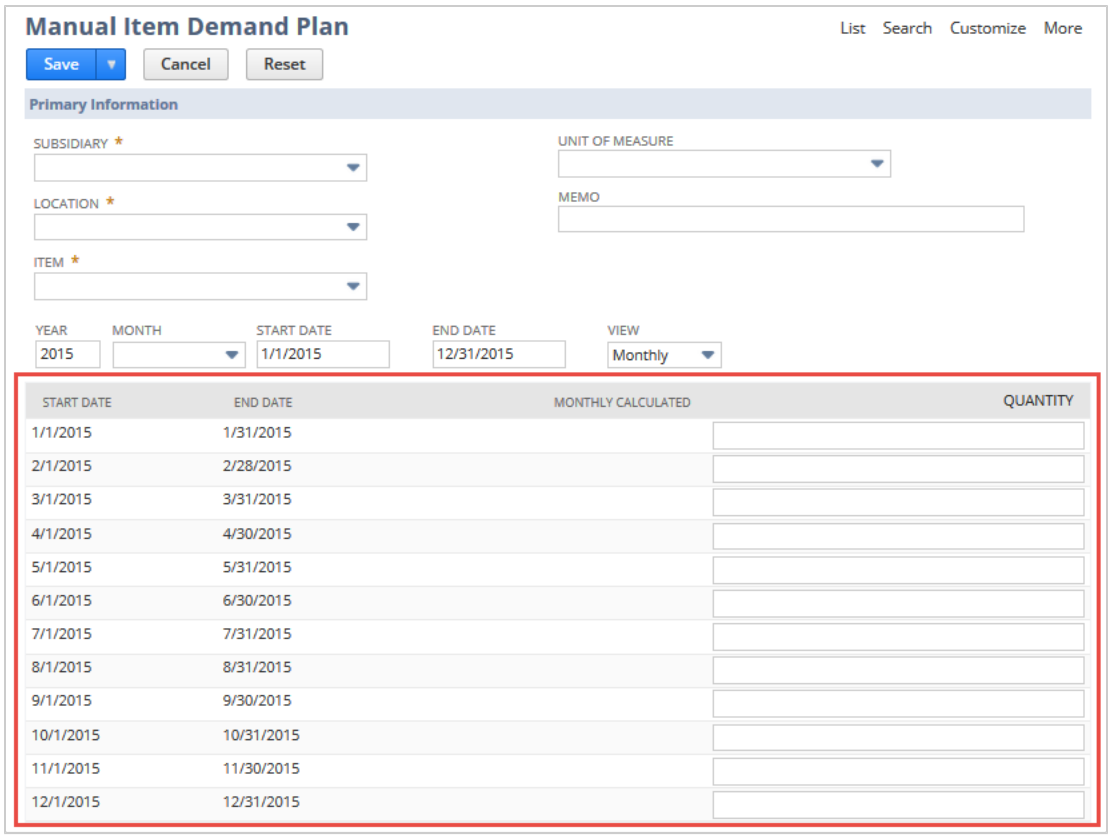
- To script with the Item Demand Plan record and the Demand Plan Detail sublist for inventory items, the Demand Planning feature must be enabled. For assembly/BOM items, the Work Orders feature also must be enabled.
- Demand plans are supported only for item(s) that have the **supplyreplenishmentmethod** field set to Time Phased.
- Required body field values must be defined before matrix field values can be edited. In dynamic mode, current values may be retrieved. Start date and end date body fields default to the first day and last day of the currenty year.

For more details and code samples for each type of demand plan, see the following:

- Monthly Demand Plan
- Weekly Demand Plan
- Daily Demand Plan

Monthly Demand Plan

A monthly demand plan includes a row for each month within the body field start date and end date, and one quantity column for each month.



- The sublist startdate and enddate fields are system-calculated and read-only.
  - The startdate is the date of the first day of the month represented by each row.
  - The enddate is the date of the last day of the month represented by each row.
  - The month for row 1 is the month set in the body field start date, the month for row 2 is the next month, and so on, until the month set in the body field end date is reached.
- The values for the quantity field can be set in SuiteScript. For monthly demand plans, the column parameter for this field is always 1.

Monthly Demand Plan Code Sample

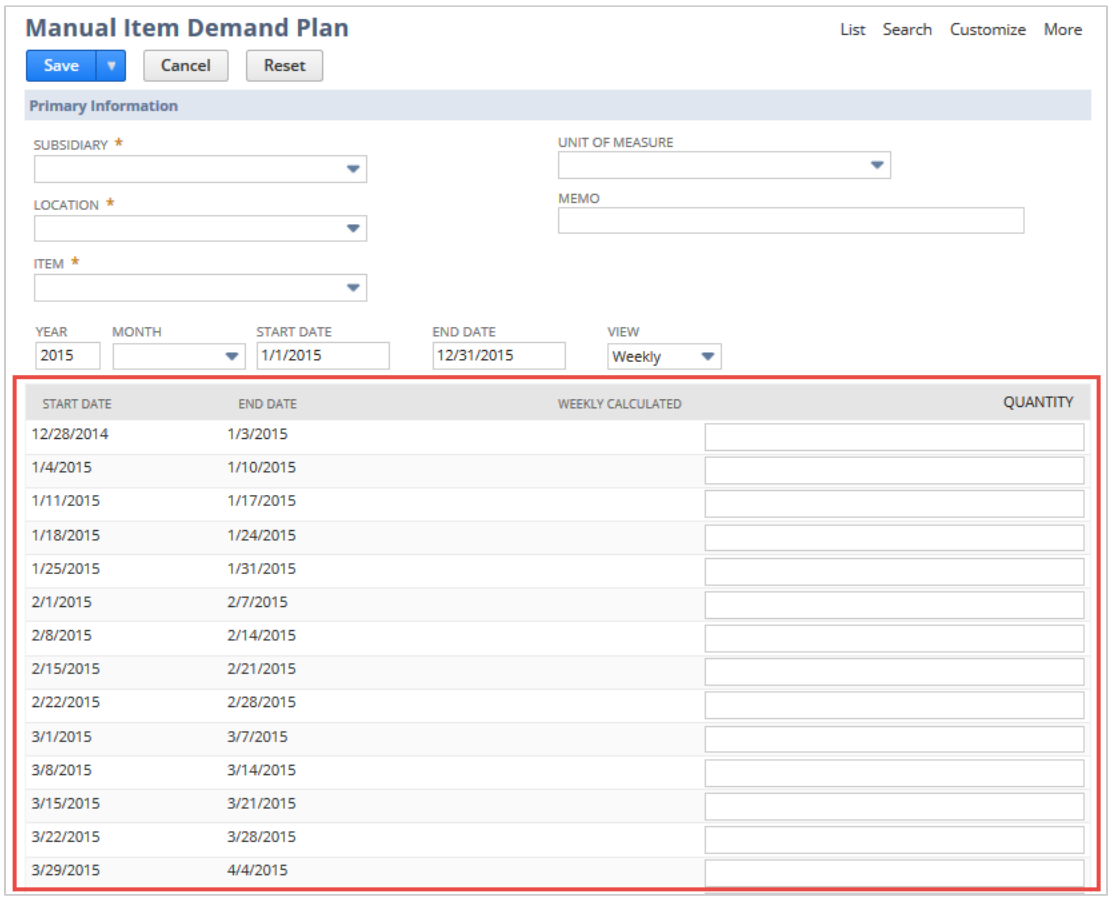
The following code sets quantities for the months of January and February, 2011:

```
var record = nlapiCreateRecord('itemdemandplan', {recordmode: 'dynamic'});
record.setFieldValue('demandplancalendartype','MONTHLY');

record.setFieldValue('subsidiary', 1);
record.setFieldValue('location', 1);
record.setFieldValue('item', 165);
record.setFieldValue('startdate','1/1/2015');
record.setFieldValue('enddate','12/31/2015');

record.selectLineItem('demandplandetail', '1');
record.setCurrentLineItemMatrixValue('demandplandetail', 'quantity', '1', 100);

record.selectLineItem('demandplandetail', '2');
record.setCurrentLineItemMatrixValue('demandplandetail', 'quantity', '1', 200);
```

```
var id = nlapiSubmitRecord(record,true);
```

Weekly Demand Plan

A weekly demand plan includes a row for each week contained in the time period set by the body field start date and end date, and one quantity column for each week.



- The sublist startdate and enddate fields are system-calculated and read-only.
  - The startdate is the date of the first day of the week represented by each row.
  - The enddate is the date of the last day of the week represented by each row.

    > **Note:** The first day of the week by default is Sunday, but may vary according tothe company preference set for First Day of the Week at Setup > Company > General Preferences.

  - The week for row 1 is the week of the date set in the body field start date. Note that unless the body field start date happens to be the first day of the week, the startdate for this first row may precede the body field start date.
  - The week for the final sublist row is the week of the date set in the body field end date. Note that unless the body field end date happens to be the last day of the week, the enddate for this last row may be after the body field enddate.
- The values for the quantity field can be set in SuiteScript. For weekly demand plans, the column parameter for this field is always 1.

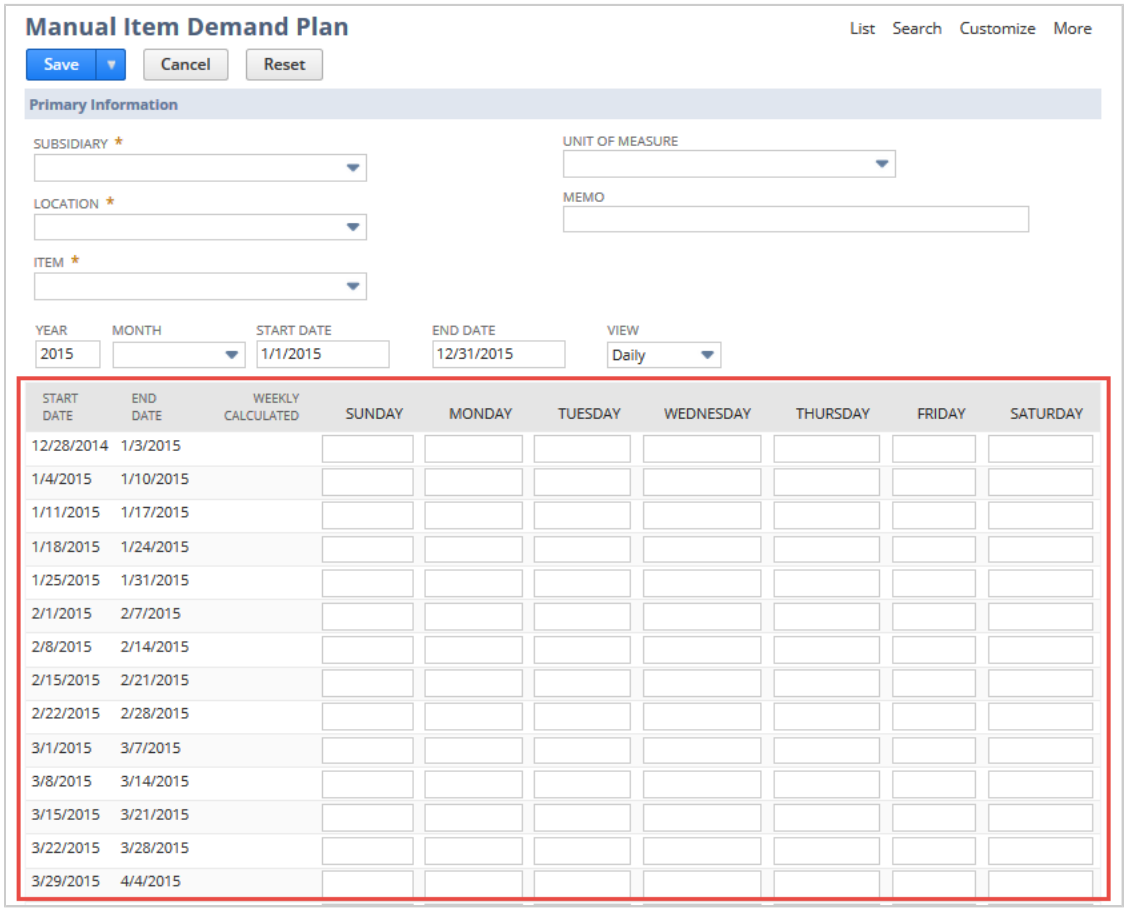
Weekly Demand Plan Code Sample

The following code sets quantities for the first two weeks of 2011:

```
var record = nlapiCreateRecord( 'itemdemandplan',{recordmode: 'dynamic'});
record.setFieldValue('demandplancalendartype',WEEKLY);

record.setFieldValue('subsidiary', 1);
record.setFieldValue('location', 1);
record.setFieldValue('item', 165);
record.setFieldValue('startdate','1/1/2015');
record.setFieldValue('enddate','12/31/2015');

record.selectLineItem('demandplandetail', '1');
record.setCurrentLineItemMatrixValue('demandplandetail', 'quantity', '1', 100);

record.selectLineItem('demandplandetail', '2');
record.setCurrentLineItemMatrixValue('demandplandetail', 'quantity', '1', 200);

var id = nlapiSubmitRecord(record,true);
```

Daily Demand Plan

A daily demand plan includes a row for each week contained in the time period set by the body field start date and end date, and seven quantity columns for each week, one for each day of the week.



- The sublist startdate and enddate fields are system-calculated and read-only.
  - The startdate is the date of the first day of the week represented by each row.
  - The enddate is the date of the last day of the week represented by each row.

    Note: The first day of the week by default is Sunday, but may vary according tothe company preference set for First Day of the Week at Setup > Company > General Preferences.

  - The week for row 1 is the week of the date set in the body field start date. Note that unless the body field start date happens to be the first day of the week, the startdate for this first row may precede the body field start date.
  - The week for the final sublist row is the week of the date set in the body field end date. Note that unless the body field end date happens to be the last day of the week, the enddate for this last row may be after the body field enddate.
- The values for the quantity fields can be set in SuiteScript.
  - The column parameter for a quantity field is 1,2,3,4,5,6, or 7, depending upon the day of the week.
  - In the screenshot above, the week starts with Sunday, which is the default first day of the week, and in this case, maps to a column parameter of 1. However, 1 does not always map to Sunday; it maps to the first day of the week as set in the company preferences.

Daily Demand Plan Code Sample

```
var record = nlapiCreateRecord( 'itemdemandplan',{recordmode: 'dynamic'});
record.setFieldValue('demandplancalendartype','DAILY');

record.setFieldValue('subsidiary', 1);
record.setFieldValue('location', 1);
record.setFieldValue('item', 165);
record.setFieldValue('startdate','1/1/2011');
record.setFieldValue('enddate','12/31/2011');

record.selectLineItem('demandplandetail', '1');   // week of 12/26/2010 to 1/1/2011
record.setCurrentLineItemMatrixValue('demandplandetail', 'quantity', '1', 100);  //sunday
record.setCurrentLineItemMatrixValue('demandplandetail', 'quantity', '2', 101);  //monday
record.setCurrentLineItemMatrixValue('demandplandetail', 'quantity', '3', 102);  //tuesday

record.selectLineItem('demandplandetail', '2');   //week of 1/2/2011 to 1/8/2011
record.setCurrentLineItemMatrixValue('demandplandetail', 'quantity', '1', 200); //sunday
record.setCurrentLineItemMatrixValue('demandplandetail', 'quantity', '5', 200); //thursday

var id = nlapiSubmitRecord(record,true);
```

Matrix APIs for the Demand Plan Detail Sublist

Use the following matrix APIs with the Demand Plan Detail sublist:

- nlapiGetCurrentLineItemMatrixValue(type, fldnam, column)
- nlapiSetCurrentLineItemMatrixValue(type, fldnam, column, value, firefieldchanged, synchronous)

> **Note:** With the two APIs above, use this API first to select an existing line: nlapiSelectLineItem(type, linenum).

- nlapiGetLineItemMatrixField(type, fldnam, linenum, column)
- nlapiGetLineItemMatrixValue(type, fldnam, linenum, column)
- nlapiFindLineItemMatrixValue(type, fldnam, val, column)

For more information about APIs, see the help topic Sublist APIs.

Item Fulfillment

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

An item fulfillment transaction records the shipment of some or all items on an order to the customer. The processes for item fulfillment transactions depend on whether the Advanced Shipping feature is enabled.

- If Advanced Shipping is not enabled, the fulfillment and invoicing processes are combined. When an item fulfillment is created, a related invoice is created at the same time.
- If Advanced Shipping is enabled, fulfillment and invoicing are two independent processes, and shipments can be recorded separately from billing.

For more details about this type of transaction, see the help topic Order Fulfillment.

The internal ID for this record is `itemfulfillment`. `Copy` and `create` are not allowed for this record.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The item fulfillment record is scriptable in both client and server SuiteScript.

Supported Functions

The item fulfillment record is partially scriptable — it can be updated, deleted, and searched using SuiteScript. It cannot be created or copied.

Usage Notes

When working with this record, you can set *pick, pack*, or *ship* as event trigger types that will execute your user event script. In the NetSuite Help Center, see the help topic User Event Script Execution Types for more information.

The Item Fulfillment/Item Receipt sublist is a **list** sublist. In the NetSuite Help Center, see the help topic List Sublists for information on this sublist type.

The Shipping sublist is included on the item fulfillment record. This sublist is not currently showing on this record in the SuiteScript Records Browser. To get the internal IDs for the Shipping sublist, open the Records Browser and navigate to one of the other record types that support this sublist. These records are sales order, cash sale, invoice, and estimate / quote.

The Shipping sublist's Shipping Tax Code field, internal ID shippingtaxcode, appears only if per-line taxes have been set on the Item sublist.

Item Receipt

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

An item receipt transaction records the receipt of returned items from customers. This transaction updates the following information:

- Items on return authorizations are recorded as received.
- Inventory records are updated for the new stock levels.
- Inventory asset accounts are updated with the values of returned items.
- Status of the return is updated.

The item receipt transaction is available when the Advanced Receiving feature is enabled.

For more details about this type of transaction, see the help topics Receiving a Customer Return and Handling Returned Items.

The internal ID of this record is `itemreceipt`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The item receipt record is scriptable in both client and server SuiteScript.

Supported Functions

The item receipt record is partially scriptable — it can be updated, deleted, and searched using SuiteScript. It cannot be created or copied.

Usage Notes

The Item Fulfillment/Item Receipt sublist is a **list** sublist. In the NetSuite Help Center, see the help topic List Sublists for information on this sublist type.

Using Landed Cost Fields

When you create a landed cost category, the associated field IDs for the first category are landedcostamount1 and landedcostsource1. If you create a second category, the IDs will be landedcostamount2 and landedcostsource2.

This pattern increments by one with each additional category. For example, the IDs for the next landed cost category will be landedcostamount3 and landedcostsource3, and so on.

Creating Item Receipt Records

You cannot create standalone item receipts using SuiteScript. For example, the following will throw an error:

```
var ir = nlapiCreateRecord("itemreceipt");
```

To create an item receipt, you must use the nlapiTransformRecord(...) API, which transforms the data from one record type, **purchase order**, for example, into an item receipt. To create an item receipt, your code would be similar to the following:

```
function trans()
{
   var fromrecord;
   var fromid;
   var torecord;
   var trecord;
   var qty;

   fromrecord = 'purchaseorder';
   fromid = 26 ; // Transform PO with ID = 26 ;
   torecord = 'itemreceipt';

   // Transform a record with a specific id to a different record type.
   // For example - from PO to Item Receipt
   // Get the object of the transformed record.
   trecord = nlapiTransformRecord(fromrecord, fromid, torecord);
   qty = trecord.getLineItemValue('item', 'quantity', 1 );
   trecord.setLineItemValue('item', 'quantity', 1, '2' );
   var idl = nlapiSubmitRecord(trecord, true);

   nlapiSendEmail(-5, -5, 'Transform Email' + 'Original Qty =
   ' + qty + ' ' + 'Record Created = ' + idl , null);
}
```

Item Supply Plan

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The internal ID for this record is `itemsupplyplan`.

This record includes the Orders sublist.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The item supply plan is scriptable in both client and server SuiteScript.

Supported Functions

The item supply plan record is fully scriptable — it can be created, copied, updated, deleted, and searched using SuiteScript.

Usage Notes

The item, location, and units body fields cannot be changed in update operations.

An item supply plan's receiptdate cannot be earlier than the orderdate.

The ordercreated field is read-only. It is set to True when an order is generated from an item supply plan.

Code Samples

The following code creates an item supply plan.

```
function createItemSupplyPlanMinimal()
{
    var isp = nlapiCreateRecord('itemsupplyplan');
    isp.setFieldValue('subsidiary', 1);
    isp.setFieldValue('location', 1);
    isp.setFieldValue('item', 165);

    isp.setFieldValue('memo', 'memotest');
    isp.setFieldValue('unit', 3);

   isp.selectNewLineItem('order');
    isp.setCurrentLineItemValue('order','orderdate', '05/05/2012');
    isp.setCurrentLineItemValue('order', 'receiptdate', '5/8/2012');
    isp.setCurrentLineItemValue('order', 'quantity', 1);
    isp.setCurrentLineItemValue('order', 'ordertype', 'PurchOrd');

    isp.commitLineItem('order');

    var id = nlapiSubmitRecord(isp);
}
```

The following code updates an existing item supply plan.

```
function updateItemSupplyPlan()
```

```
{
    var isp = nlapiLoadRecord('itemsupplyplan', 3);
   isp.setFieldValue('memo','memotest2');

   isp.setLineItemValue('order', 'receiptdate', 4, '11/3/2012');
   var id = nlapiSubmitRecord(isp);
}
```

Journal Entry

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

You use the journal entry record to adjust balances in your ledger accounts without entering posting transactions.

In the user interface, you access this record in the user interface at Transactions > Financial > Make Journal Entries.

If your account has the Multi-Book Accounting feature enabled, you can also work with book specific journal entry records, which are available in the user interface at Transactions > Financial > Make Book Specific Journal Entries. Although they have different entry forms, both book specific and regular intercompany journal entries are the same record type. Within SuiteScript, they are differentiated by the accountingbook field. In other words, a record that has a value set for accountingbook is book specific. Otherwise, the record is a regular intercompany journal entry.

For help working with this record in the user interface, see the help topics Intercompany Journal Entries and Book-Specific Journal Entries.

The internal ID for this record is `journalentry`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The journal entry record is scriptable in both client SuiteScript and server SuiteScript.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

The journal entry record is fully scriptable — it can be created, copied, updated, deleted, and searched using SuiteScript.

Usage Notes

With the Advanced Revenue Management feature, you can directly attach a revenue recognition plan to a book specific journal entry or an intercompany journal entry. Before you begin working with

advanced revenue management programmatically, see the help topic Setup for Advanced Revenue Management.

The following table lists the scriptable field associated with this record and advanced revenue management.

Field	Type	Internal ID
End Date	Date	`enddate`
Revenue Recognition Rule	List/Record	`revenuerecognitionrule`
Start Date	Date	`startdate`

For help working with this record in the user interface, see the help topic Creating Revenue Elements from Journal Entries

Code Sample

The following example shows how to create a book specific journal entry. The record is book specific because a value has been set for the accountingbook field.

```
var initvalues = new Array();
initvalues.bookje = 'T';
var rec = nlapiCreateRecord('journalentry', initvalues);
rec.setFieldValue('accountingbook', '2'); // Setting a value for this field makes the record bo
ok specific.
rec.setFieldValue('subsidiary', '4');
rec.setFieldValue('trandate', '5/16/2013');
rec.selectNewLineItem('line');
rec.setCurrentLineItemValue('line', 'account', '6');
rec.setCurrentLineItemValue('line', 'credit', '2.00');
rec.commitLineItem('line');
rec.selectNewLineItem('line');
rec.setCurrentLineItemValue('line', 'account', '149');
rec.setCurrentLineItemValue('line', 'debit', '2.00');
rec.commitLineItem('line');
var id = nlapiSubmitRecord(rec);
var x = 0;
```

Landed Cost

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The landed cost subrecord supports the Landed Cost Allocation per Line functionality, which is part of the Landed Cost feature. Landed costs typically include location-specific expenses such as customs duties and freight fees. The landed cost subrecord can be used in conjunction with several transactions: check, credit card charge, item receipt, and vendor bill. The purpose of the subrecord is to show the landing costs associated with a particular line in the parent transaction's Items sublist.

The subrecord is available only when the Landed Cost feature is enabled at Setup > Company > Setup Tasks > Enable Features, on the Items & Inventory subtab. For details on working with the subrecord in the user interface, see the help topic Using Landed Cost Allocation Per Line on Transactions.

The internal ID for this subrecord is `landedcost`.

Important: Landed cost is considered a subrecord, not a record. Subrecords are represented by the nlobjSubrecord object in SuiteScript. For general details on working with subrecords, see the help topic Working with Subrecords in SuiteScript. For details about this finding information for subrecords in the Record Browser, see the help topic Finding Subrecord Details in the Records Browser.

Supported Script Types

The landed cost subrecord is scriptable in server SuiteScript only. The user events are not supported.

Supported Functions

The landed cost subrecord is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

To script to the landed cost subrecord, both of the following must be true:

- The parent transaction's Landed Cost per Line option is checked.
- The sublist item being referenced has been configured to use the Track Landed Cost option. For details on configuring the item, see the help topic Set Up Item Records for Landed Cost.

Code Sample

The following snippets show how to create a landed cost subrecord and perform other basic tasks.

```
// Creating a landed cost subrecord
var purchaseOrder = nlapiCreateRecord('purchaseorder');
purchaseOrder.setFieldText('entity', 'Acme Medical Supply');
purchaseOrder.setLineItemValue('item', 'item', 1, inventoryItemId);

var purchaseOrderId = nlapiSubmitRecord(purchaseOrder);

var itemReceipt = nlapiTransformRecord('purchaseorder', purchaseOrderId, 'itemreceipt');
itemReceipt.selectLineItem('item', 1);
itemReceipt.setCurrentLineItemValue('item', 'location', 1);
itemReceipt.setFieldValue('landedcostperline', 'T');

var landedCost = itemReceipt.createCurrentLineItemSubrecord('item', 'landedcost');
landedCost.selectNewLineItem('landedcostdata');
landedCost.setCurrentLineItemValue('landedcostdata', 'costcategory', 1);
landedCost.setCurrentLineItemValue('landedcostdata', 'amount', 456);
landedCost.commitLineItem('landedcostdata');
landedCost.selectNewLineItem('landedcostdata');
landedCost.setCurrentLineItemValue('landedcostdata', 'costcategory', 3);
landedCost.setCurrentLineItemValue('landedcostdata', 'amount', 78.96);
landedCost.commitLineItem('landedcostdata');
landedCost.commit();
```

```
itemReceipt.commitLineItem('item');

var itemReceiptId = nlapiSubmitRecord(itemReceipt);


// Viewing the subrecord
itemReceipt = nlapiLoadRecord('itemreceipt', itemReceiptId);
itemReceipt.selectLineItem('item', 1);
landedCost = itemReceipt.viewLineItemSubrecord('item', 'landedcost', 1);
landedCost.getLineItemValue('landedcostdata', 'amount', i);


// Updating the subrecord
landedCost = itemReceipt.editCurrentLineItemSubrecord('item', 'landedcost');

landedCost.removeLineItem('landedcostdata', 2);

landedCost.setLineItemValue('landedcostdata', 'costcategory', 1, 2);
landedCost.setLineItemValue('landedcostdata', 'amount', 1, 3.98);
landedCost.selectNewLineItem('landedcostdata');
landedCost.setCurrentLineItemValue('landedcostdata', 'costcategory', 3);
landedCost.setCurrentLineItemValue('landedcostdata', 'amount', 103);
landedCost.commitLineItem('landedcostdata');
landedCost.commit();

itemReceipt.commitLineItem('item');

nlapiSubmitRecord(itemReceipt);


// Deleting the subrecord
itemReceipt.selectLineItem('item', 1);
itemReceipt.removeCurrentLineItemSubrecord('item', 'landedcost');
itemReceipt.commitLineItem('item');
nlapiSubmitRecord(itemReceipt);
```

Manufacturing Operation Task

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The Manufacturing Routing and Work Center feature lets you specify a sequence of tasks required for the completion of a Work In Process (WIP) work order. This record represents a job that must be completed by a specific employee group. In the UI, generally these records are created automatically when you save a WIP work order that references a specific routing record — each step described in the routing record becomes an operation task record, viewable on the work order's Operations subtab. You can also manually create an operation task record by clicking the New Operation Task button on the work order's Operations subtab. You can view all existing manufacturing operation task records by going to Transactions > Manufacturing > Manufacturing Operation Tasks.

For more details about this record, see the help topic Manufacturing Operation Tasks. For more information about the Manufacturing Routing and Work Center feature, see Manufacturing Operation Task.

The internal ID for this record is `manufacturingoperationtask`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The manufacturing operation task record is scriptable in both client and server SuiteScript.

Supported Functions

The manufacturing operation task record is partially scriptable — it can be created, updated, deleted, and searched using SuiteScript. It cannot be copied.

Usage Notes

To work with the manufacturing operation task record, the Manufacturing Routing and Work Center feature must be enabled at Setup > Company > Enable Features, on the Items & Inventory tab.

In the UI, the manufacturing operation task record is accessed by going to Transactions > Manufacturing > Manufacturing Operation Tasks. Alternatively, you can navigate to the Operations subtab of a WIP work order that uses the routing feature. The Operations subtab lists existing operation task records for that work order and allows you to create new operation task records.

Note these additional details:

- This record supports client and server-side scripting.
- All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.
- Refer to the table below for details on supported API functions.
- To create a new manufacturing operation task record, you must reference a valid existing WIP work order, as shown in the Code Sample below.

Code Sample

When creating a manufacturing operation task record, you must use initValues to reference a valid existing WIP work order. For example:

```
var initValues = new Array();
initValues.workorder = '65';
var task = nlapiCreateRecord('manufacturingoperationtask', initValues);
task.setFieldValue('title', 'Some title');
task.setFieldValue('operationsequence', 6);
task.setFieldValue('setuptime', 30);
task.setFieldValue('runrate', 20);
task.setFieldValue('manufacturingcosttemplate', '1');
task.setFieldValue('manufacturingworkcenter', '113');
var recId = nlapiSubmitRecord(task);
```

Manufacturing Planned Time

Note: The content in this help topic pertains to all versions of SuiteScript.

The Manufacturing Planned Time search type enables you to search for data displayed on the Planned Time subtab of work orders. This subtab is available only when the Show Planned Capacity on Work Orders preference is enabled.

The Planned Time subtab is used to display data about the amount of time being allocated to each work center per day for the work order. This data is derived from the associated operation task records.

You can set the Show Planned Capacity on Work Orders preference at Setup > Accounting > Accounting Preferences. The preference is available only when the Manufacturing Routing and Work Center feature is enabled.

The internal ID for this record is `mfgplannedtime`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

This search is supported in both client and server SuiteScript.

Supported Functions

Only search is supported for this record.

Usage Notes

Be aware of the following:

- This record is a search record only. You cannot create or copy this record.
- In the user interface, you can view this search by navigating to Reports > New Search and clicking Manufacturing Planned Time.

For help creating search scripts, see the help topic Search Samples.

Multi-Book Accounting Transaction

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

If your account has the Multi-Book Accounting feature enabled, you can use SuiteScript to search for transactions using accounting book as a search filter or a search column. To execute this type of search, you use the multi-book accounting transaction search record.

In the user interface, you can view the multi-book accounting transaction search by navigating to Reports > New Search and clicking Multi-Book Accounting Transaction.

The internal ID for this record is `accountingtransaction`. Note that this record is a search record only. You cannot create or copy this record.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The multi-book accounting transaction record is supported in both client and server SuiteScript.

Supported Functions

Only search is supported for this record.

Sample Code

The following example shows how to search for a specific transaction and include in the results details about the transaction's accounting book and foreign exchange rate, among other data.

```
var filters = new Array();
filters[0] = new nlobjSearchFilter( 'internalid', null, 'is', '18' );
// last parameter represents the actual internal ID of the transaction

var columns = new Array();
columns[0] = new nlobjSearchColumn( 'accountingbook' ).setSort(false);
columns[1] = new nlobjSearchColumn( 'line', 'transaction', null ).setSort(false);
columns[2] = new nlobjSearchColumn( 'account' );
columns[3] = new nlobjSearchColumn( 'amount' );
columns[4] = new nlobjSearchColumn( 'exchangerate' );

var searchresults = nlapiSearchRecord( 'accountingtransaction', null, filters, columns );

for ( var i = 0; searchresults != null && i < searchresults.length; i++ )
{
        var searchresult = searchresults[ i ];
        var record = searchresult.getId( );
        var rectype = searchresult.getRecordType( );
        var book = searchresult.getValue( 'accountingbook' );
        var line = searchresult.getValue( 'line', 'transaction' );
        var account= searchresult.getValue( 'account' );
        var amount= searchresult.getValue( 'amount' );
        var fxrate= searchresult.getValue( 'exchangerate' );

alert(i + ": " + book + " | " + line + " | " + record + " | " + rectype + " | " + account + " |
 " + amount + " | " + fxrate);
}
```

Opportunity

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Opportunities represent negotiations with prospects. You must first enable opportunities in your NetSuite account before you can access this record type. To enable opportunities, go to Setup > Company > Enable Features. On the CRM tab, under Sales, select the Opportunities check box.

For details about working with opportunities in NetSuite, see Opportunity Records.

The internal ID for this record is `opportunity`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The opportunity record is scriptable in both client and server SuiteScript.

Supported Functions

The manufacturing operation task record is fully scriptable — it can be created, copied, updated, deleted, and searched using SuiteScript. It can also be transformed.

Usage Notes

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
Body Fields		
estgrossprofit	Est. Gross Profit	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
estgrossprofitpercent	Est. Gross Profit Percent	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
totalcostestimate	Est. Extended Cost	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search.

Also note that this record has available transforms. See the SuiteScript Records Browser for available transforms. In the NetSuite Help Center, see the help topic nlapiTransformRecord(type, id, transformType, transformValues) for examples on how to transform records.

The Multi-Partner Management feature must be enabled in your account for the Partners sublist to appear.

Order Schedule

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The internal ID for this subrecord is `orderschedule`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

Server-side scripts for this subrecord must access through the parent record.

Supported Functions

For details, see the help topic Working with Subrecords in SuiteScript.

Usage Notes

Order Schedule is scriptable from both the body field and the line item.

Order Schedule is considered a subrecord, represented by the nlobjSubrecord object in SuiteScript.

Paycheck

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

This record is available only when the Payroll feature is enabled at Setup > Company > Enable Features on the Employees subtab.

In the user interface, you can access this record at Transactions > Employees > Create Payroll. For help working with this record in the user interface, see the help topics Creating a Payroll Batch and Editing an Individual Paycheck from a Payroll Batch.

The internal ID for this record is `paycheck`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

This record is scriptable in both client and server SuiteScript.

Supported Functions

The paycheck record is partially scriptable — it can be read and updated only.

Usage Notes

Be aware of the following:

- The paycheck record is created through the payroll batch record.

Code Samples

The following code snippets show how to search for paycheck records.

```
var filters = new Array();
filters[0] = new nlobjSearchFilter('batchnumber', null, 'equalTo', 100);

// Define search columns
var columns = new Array();
columns[0] = new nlobjSearchColumn('batchnumber');
columns[1] = new nlobjSearchColumn('employee');

var searchResults = nlapiSearchRecord('paycheck', null, filters, columns);
```

Paycheck Journal

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The Paycheck Journal record is intended to enable global payroll solutions. You can use it along with the Payroll Item record to create custom payroll solutions and to support integrations with external payroll systems.

The internal ID for this record is `paycheckjournal`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Usage Notes

The Paycheck Journal feature must be enabled to work with the Paycheck Journal record.

Code Samples

The following sample creates a paycheck journal transaction.

```
//create Paycheck Journal with earnings and deduction sublist
// add 2 Earning and 1 deduction sublists

function createPaycheckJournal()
{
    var pj = nlapiCreateRecord('paycheckjournal');
```

```
   pj.setFieldValue('trandate', "6/10/2012");
   pj.setFieldValue('employee', 4 ); //internal Id of employee
   pj.setFieldValue('account', 28 ); //internal Id of account

   pj.selectNewLineItem('earning');
       pj.setCurrentLineItemValue('earning','payrollitem', '102');
      pj.setCurrentLineItemValue('earning', 'amount', 20.35);
      pj.commitLineItem('earning');

   pj.selectNewLineItem('earning');
      pj.setCurrentLineItemValue('earning','payrollitem', '102');
       pj.setCurrentLineItemValue('earning', 'amount', 33.35);
      pj.commitLineItem('earning');

   pj.selectNewLineItem('deduction');
       pj.setCurrentLineItemValue('deduction','payrollitem', '103');
       pj.setCurrentLineItemValue('deduction', 'amount', 444.44);
       pj.commitLineItem('deduction');

   nlapiSubmitRecord(pj);
}
```

The following sample updates a paycheck journal transaction.

```
//update Paycheck journal
//set new amount of line 2 Earning list
// and clear deduction list

function updatePaycheckJournal()
{
    var pj = nlapiLoadRecord('paycheckjournal', 305 ); // internalID of

   pj.setLineItemValue('earning', 'amount', 2, 444.44);  // 2 is the line no we intend to updat
e

   for (var lineNo=1; lineNo <= pj.getLineItemCount('deduction'); lineNo++)
           pj.removeLineItem('deduction', lineNo );
   nlapiSubmitRecord(pj);
}
```

Payroll Batch

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

This record is available only when the Payroll feature is enabled at Setup > Company > Enable Features on the Employees subtab.

In the user interface, you can access this record at Transactions > Employees > Create Payroll. For help working with this record in the user interface, see the help topic Creating a Payroll Batch.

The internal ID of this record is `payrollbatch`.

Supported Script Types

The payroll batch record is scriptable in both client, and server SuiteScript.

Supported Functions

The payroll batch record is fully scriptable, which means that the record can be created, updated, copied, deleted and searched using SuiteScript.

Usage Notes

Be aware of the following:

- After a payroll batch is created it is submitted twice. One time to calculate and again to commit.
- To calculate a payroll batch, it needs to be reloaded after it is submitted until one of the following status' is generated: calculated or error, committed or error.
- If you are calculating and committing a large amount of data a loop should be written to recheck the status of the payroll batch. Refer to the following table for more details.

Payroll Batch Status	Value	Description
`Created`	A	A new payroll batch has been created.
`Calculated`	B	A payroll batch has been calculated.
`Edited`	C	A payroll batch may have been calculated previously, or it is a newly created batch with some paychecks added or calculated.
`About to Commit`	D	Payroll is about to be committed.
`Committed_At_Service`	E	Payroll is committed for processing by payroll service, but some records need to be created in NetSuite.
`Committed`	F	Payroll is committed.
`Completed`	P	Payroll batch is completed.
`Reversed`	R	Payroll batch is reversed.
`Error`	X	Error occurred in payroll batch.

Code Samples

The following code snippets show how to create a payroll batch and perform other basic tasks. It also includes how to add an employee to a payroll batch. For more information, see Payroll Batch Employee.

```
<code>

//Create the payroll batch record
var record = nlapiCreateRecord('payrollbatch', 'true');
record.setFieldValue('offcycle', 'F');
record.setFieldValue('payfrequency', '52');
record.setFieldValue('periodending', '6/15/2016');

var id = nlapiSubmitRecord(record);
```

```
//Add employees to the payroll batch
//In most cases, you would iterate through the addmorepayeesmachine to find the employees you w
ould add to the batch
var addemp = nlapiLoadRecord('payrollbatchaddemployees', id, 'true');
addemp.setLineItemValue('addmorepayeesmachine','payemp', 1,'T');
nlapiSubmitRecord(addemp);

//Submit the record to be calculated
record=nlapiLoadRecord('payrollbatch', id, 'false');
nlapiSubmitRecord(record);

//Check the status of the record to ensure that calculating is complete before proceeding
do}
 record = nlapiLoadRecord('payrollbatch', id, 'false');
 status = record.getFieldValue('status');
}while ( !(status =='B' || status == 'E' || status =='X') );

//Set the flag to commit the payroll batch
if (statusa =='B') {
 record.setFieldValue('commit', 'T');

//Commit
 nlapiSubmitRecord(record);

do {
 record = nlapiLoadRecord('payrollbatch', id, 'false');
 status = record.getFieldValue('status');
}while ( !(status == 'F' || status == 'E' || status == 'X') );

</code>
```

Payroll Batch Employee

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

This record enables you to add employees to the payroll batch.

This record is available only when the Payroll feature is enabled at Setup > Company > Enable Features on the Employees subtab.

In the user interface, you can access this record at Transactions > Employees > Create Payroll. For help working with this record in the user interface, see the help topic Adding Employees to a Payroll Batch.

The internal ID of this record is `payrollbatchaddemployees`.

Supported Script Types

This record is scriptable in both client and server SuiteScript.

Supported Functions

The payroll batch employee record is read and edit only.

Usage Notes

Be aware of the following:

- You cannot deselect an employee through SuiteScript after they have been added to the payroll batch and the record is submitted. If you need to remove an employee from the payroll batch you will need to do this through the UI.

Price Plan

Note: The content in this help topic pertains to all versions of SuiteScript.

The price plan record, along with the price book record, is used to build pricing information for subscriptions.

The price plan record is available only when the SuiteBilling feature is enabled at Setup > Company > Enable Features (Administrator), on the Transactions subtab.

The internal ID for this record is `priceplan`.

See the SuiteScript Records Browser for all internal IDs associated with this record. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The price plan record is scriptable in server and client SuiteScript.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

The price plan record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Purchase Contract

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

This record enables you to take advantage of contracted quantity-based terms and discounts when creating purchase orders.

This record is available only when the Purchase Contracts feature is enabled at Setup > Company > Enable Features, on the Transactions subtab.

In the user interface, you access this record at Transactions > Purchases > Enter Purchase Contracts. For help working with this record in the user interface, see the help topic Creating Purchase Contracts.

The internal ID of this record is `purchasecontract`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

This record is scriptable in both client and server SuiteScript.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

This record is fully scriptable, which means that the record can be created, updated, copied, deleted, and searched using SuiteScript.

Purchase Order

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

A purchase order transaction authorizes the purchase of goods and/or services. This transaction tracks items expected to be received, items actually received, and items yet to be received. A purchase order has no accounting impact until the included items are received. This transaction is enabled when the Purchase Orders feature is enabled.

For more details about this type of transaction, see the help topic Purchasing.

The internal ID for this record is `purchaseorder`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The purchase order record is scriptable in both client and server SuiteScript.

Supported Functions

This record is fully scriptable, which means that the record can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Usage Notes

See the following sections for more details on this record:

- Fields
- Transforms

- Email Operations

Fields

estgrossprofit	Est. Gross Profit	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
estgrossprofitpercent	Est. Gross Profit Percent	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
totalcostestimate	Est. Extended Cost	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search.

Transforms

This record has available transforms. See the SuiteScript Records Browser for available transforms. In the NetSuite Help Center, see the help topic nlapiTransformRecord(type, id, transformType, transformValues) for examples on how to transform records.

Email Operations

NetSuite enables a user to email a copy of a transaction to a customer or another recipient.

If you deploy a beforeLoad user event script on a transaction, in general the script executes when the transaction is emailed. In these cases, the sending of the email is considered an operation of type email on the transaction. However, if the companywide preference **Use Popup for Main Transaction Email Button** is enabled, a beforeLoad script will not execute in certain cases, depending on how the email is sent. That is, the script will not execute if the user views the transaction and selects Actions > Email, or edits the transaction and sends the email by using the Communications tab. In these cases, the system displays a popup window that the user can work with to manually edit and send the email. Actions taken with this window are not considered operations on the transaction. If you have your system configured this way, and you want to deploy a beforeLoad script on emails sent using the popup, deploy the script on the Message record type.

An additional way of emailing a transaction is by editing the transaction and using the Save & Email button. Note that, even if the Use Popup for Main Transaction Email Button preference is enabled, the popup window is not available for emails sent this way. For that reason, a beforeLoad user event script deployed on the transaction will always be triggered by use of the Save & Email button.

For more information on the Use Popup for Main Transaction Email Button preference, see the help topic Setting Email Preferences. For details about emailing transactions, see the help topic Emailing Transactions.

Requisition

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

You use the requisition record to initiate the purchase process for goods and services needed within your company.

This record is available only when the Requisitions feature is enabled, at Setup > Enable Features, on the Transactions subtab. In the user interface, you access this record at Transactions > Purchases/ Vendors > Enter Requisitions. For help working with this record in the user interface, see the help topic Entering a Requisition.

The internal ID for this record is `purchaserequisition`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The requisition record is scriptable in both client and server SuiteScript.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

The requisition record is fully scriptable, which means that the record can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Code Samples

The following code snippets show how to add a requisition record and perform other basic tasks.

```
// Requisition SuiteScript name
var recType = 'PurchaseRequisition';

// Create and Add Record
var rec = nlapiCreateRecord(recType);
rec.setFieldValue('location',1);
rec.setLineItemValue('expense','account',1,59);
rec.setLineItemValue('expense','amount',1,2.3);
var recId = nlapiSubmitRecord(rec);

// Load and Update
rec = nlapiLoadRecord(recType,recId);
rec.setFieldValue('memo','memo');
nlapiSubmitRecord(rec);

// Search
var searchFilters = new Array();
searchFilters[0] = new nlobjSearchFilter('internalId', null, 'anyOf', recId);
var searchColumns = new Array();
searchColumns[0] = new nlobjSearchColumn('memo');
var result = nlapiSearchRecord('transaction', null, searchFilters, searchColumns);

// Copy
var rec2 = nlapiCopyRecord(recType, recId);
var rec2Id = nlapiSubmitRecord(rec2);

// Transformation (Initialization)
```

```
var po = nlapiTransformRecord(recType,rec2Id,"PurchaseOrder");
po.setFieldValue('entity','105');
var poId = nlapiSubmitRecord(po);

// Delete
nlapiDeleteRecord('PurchaseOrder',poId);
nlapiDeleteRecord(recType,recId);
nlapiDeleteRecord(recType,rec2Id);
```

Return Authorization

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

A return authorization transaction, also known as a return materials authorization (RMA), records information about an expected return of items from a customer, including the item IDs, vendors, quantities, and prices (which determine the amounts to be credited or refunded to the customer. This transaction is non-posting. It is available when the Return Authorizations feature is enabled.

For more details about this type of transaction, see the help topic Customer Return Management.

The internal ID for this record is `returnauthorization`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The return authorization record is scriptable in both client and server SuiteScript.

Supported Functions

The return authorization record is fully scriptable, which means that the record can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Usage Notes

See the following sections for more details on working with this record:

- Transforms
- Email Operations

Fields

estgrossprofit	Est. Gross Profit	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search.

		When this field appears on the sublist line level, this field is not scriptable.
estgrossprofitpercent	Est. Gross Profit Percent	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
totalcostestimate	Est. Extended Cost	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search.

The Multi-Partner Management feature must be enabled in your account for the Partners sublist to appear.

Transforms

This record has available transforms. See the SuiteScript Records Browser for available transforms. In the NetSuite Help Center, see the help topic nlapiTransformRecord(type, id, transformType, transformValues) for examples on how to transform records.

Email Operations

NetSuite enables a user to email a copy of a transaction to a customer or another recipient.

If you deploy a beforeLoad user event script on a transaction, in general the script executes when the transaction is emailed. In these cases, the sending of the email is considered an operation of type email on the transaction. However, if the companywide preference **Use Popup for Main Transaction Email Button** is enabled, a beforeLoad script will not execute in certain cases, depending on how the email is sent. That is, the script will not execute if the user views the transaction and selects Actions > Email, or edits the transaction and sends the email by using the Communications tab. In these cases, the system displays a popup window that the user can work with to manually edit and send the email. Actions taken with this window are not considered operations on the transaction. If you have your system configured this way, and you want to deploy a beforeLoad script on emails sent using the popup, deploy the script on the Message record type.

An additional way of emailing a transaction is by editing the transaction and using the Save & Email button. Note that, even if the Use Popup for Main Transaction Email Button preference is enabled, the popup window is not available for emails sent this way. For that reason, a beforeLoad user event script deployed on the transaction will always be triggered by use of the Save & Email button.

For more information on the Use Popup for Main Transaction Email Button preference, see the help topic Setting Email Preferences. For details about emailing transactions, see the help topic Emailing Transactions.

Revenue Arrangement

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Revenue Arrangement is a transaction record that contains the details of customer performance obligations for purposes of revenue allocation and recognition.

This record is part of advanced revenue management. To use advanced revenue management, the Accounting Periods feature and Advanced Revenue Management feature must be enabled. Before you

begin working with advanced revenue management programmatically, see the help topic Setup for Advanced Revenue Management.

The internal ID for this record is `revenuearrangement`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The revenue arrangement record is scriptable in server SuiteScript only.

Supported Functions

The return authorization record is fully scriptable, which means that the record can be created, updated, copied, deleted, and searched using SuiteScript.

Revenue Element Sublist

The internal id for this sublist is `revenueelement`. The sublist type is inline editor.

The following table lists the scriptable line item fields on the Revenue Element sublist. Revenue elements correspond to individual lines in a source transaction.

Field	Type	Internal ID
Allocation Type	List/Record	`allocationtype`
Amortization End Date	Date	`amortizationenddate`
Amortization Start Date	Date	`amortizationstartdate`
Base Fair Value	Decimal Number	`fairvalue`
Calculated Fair Value Amount	Decimal Number	`calculatedamount`
Contract Expense Account	List/Record	`contractexpenseacct`
Contract Expense Offset Account	List/Record	`contractexpenseoffsetacct`
Cost Amortization Amount	Decimal Number	`costamortizationamount`
Deferral Account	List/Record	`deferralaccount`
End Date	Date	`revrecenddate`
Fair Value Override	Check Box	`fairvalueoverride`
VSOE	Check Box	`isvsoeprice`
Permit Discount	Check Box	`permitdiscount`
Reference ID	Long Text	`referenceid`
Recognition Account	List/Record	`recognitionaccount`
Return of Element	List/Record	`returnofelement`
Revenue Allocation Group	List/Record	`revenueallocationgroup`

Revenue Amount	Decimal Number	`allocationamount`
Revenue Recognition Rule	List/Record	`revenuerecognitionrule`
Start Date	Date	`revrecstartdate`

For help working with this record in the user interface, see the help topic Revenue Arrangement Management.

Revenue Commitment

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The internal ID for this record is `revenuecommitment`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The revenue commitment record is scriptable in client and server SuiteScript.

Supported Functions

The revenue commitment record is partially scriptable. It can be updated, copied, deleted, and searched using SuiteScript. It cannot be created.

Usage Notes

You cannot create this record using the standard nlapiCreateRecord(...) function. To create a Revenue Commitment record, you must execute a Sales Order to Revenue Commitment transformation. In the NetSuite Help Center, see the help topic nlapiTransformRecord(type, id, transformType, transformValues) for examples on how to transform records.

The Multi-Partner Management feature must be enabled in your account for the Partners sublist to appear.

Revenue Commitment Reversal

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

If a revenue commitment cannot be invoiced, or the revenue and unbilled receivable must be reversed for some reason, a user with sufficient permission must reverse the revenue commitment. For details, see the help topic Creating Revenue Commitment Reversals.

The internal ID for this record is `revenuecommitmentreversal`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The revenue commitment reversal record is scriptable in client and server SuiteScript.

Supported Functions

The revenue commitment reversal record is partially scriptable. It can be updated, copied, deleted, and searched using SuiteScript. It cannot be created.

Usage Notes

You cannot create this record using the standard nlapiCreateRecord(...) function. To create a Revenue Commitment Reversal record, you must execute a Return Authorization to Revenue Commitment Reversal transformation. Note that the Return Authorization must be approved and received for the transform to work.

In the NetSuite Help Center, see the help topic nlapiTransformRecord(type, id, transformType, transformValues) for examples on how to transform records.

Sales Order

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

A sales order transaction records a commitment to sell goods or services to a customer. Sales orders have no accounting impact until items are shipped or services are completed.

For details about this type of transaction, see the help topic Sales Orders.

The internal ID for this record is `salesorder`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The sales order record is scriptable in both client and server SuiteScript.

Supported Functions

The return authorization record is fully scriptable, which means that the record can be created, updated, copied, deleted, and searched using SuiteScript. It can also be transformed.

Usage Notes

See the following sections for more details about working with this record:

- Fields
- Email Operations

Fields

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
Body Fields		
estgrossprofit	Est. Gross Profit	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
estgrossprofitpercent	Est. Gross Profit Percent	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
isrecurringpayment	Recurring Payment	A value for this field is stored only if the value for paymentmethod is a credit card.
totalcostestimate	Est. Extended Cost	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search.

Also note that this record has available transforms. See the SuiteScript Records Browser for available transforms. In the NetSuite Help Center, see the help topic nlapiTransformRecord(type, id, transformType, transformValues) for examples on how to transform records.

The Shipping sublist's Shipping Tax Code field, internal ID shippingtaxcode, appears only if per-line taxes have been set on the Item sublist.

Email Operations

NetSuite enables a user to email a copy of a transaction to a customer or another recipient.

If you deploy a beforeLoad user event script on a transaction, in general the script executes when the transaction is emailed. In these cases, the sending of the email is considered an operation of type email on the transaction. However, if the companywide preference **Use Popup for Main Transaction Email Button** is enabled, a beforeLoad script will not execute in certain cases, depending on how the email is sent. That is, the script will not execute if the user views the transaction and selects Actions > Email, or edits the transaction and sends the email by using the Communications tab. In these cases, the system displays a popup window that the user can work with to manually edit and send the email. Actions taken with this window are not considered operations on the transaction. If you have your system configured this way, and you want to deploy a beforeLoad script on emails sent using the popup, deploy the script on the Message record type.

An additional way of emailing a transaction is by editing the transaction and using the Save & Email button. Note that, even if the Use Popup for Main Transaction Email Button preference is enabled, the popup window is not available for emails sent this way. For that reason, a beforeLoad user event script deployed on the transaction will always be triggered by use of the Save & Email button.

For more information on the Use Popup for Main Transaction Email Button preference, see the help topic Setting Email Preferences. For details about emailing transactions, see the help topic Emailing Transactions.

Statistical Journal Entry

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The statistical journal entry record lets you increase or reduce the balance of a statistical account.

To use this record, the Statistical Accounts feature must be enabled at Setup > Enable Features, on the Accounting subtab. Also, you must have already created at least one statistical account. In the UI, you access this record at Transactions > Financial > Make Statistical Journal Entries. For help working with this record in the user interface, see the help topic Making Statistical Journal Entries.

This internal ID for this record is `statisticaljournalentry`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The statistical journal entry record is scriptable in both client and server SuiteScript.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

The statistical journal entry record is fully scriptable, which means that the record can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

Be aware of the following:

- The unitlabel field is read-only. It is populated automatically when you set the unitstype field. (The unitlabel value is the unitstype's base unit value. This value is defined in the corresponding unit of measure record.)
- The unitstype body field cannot be updated after the record is created.
- The field labeled Amount in the user interface is called debit in SuiteScript.
- Every statistical journal entry record must have at least one line.
- All sublist lines must use the same unit of measure, which is defined by the unitstype body field.

Code Samples

The following samples show you can script to the statistical journal entry record.

Adding and Deleting

The following code snippets show how you can create the record, add lines, load the record, and delete it.

```
// Note: These samples use constants. They are defined like: var subsidiaryId = '1';
```

```
// Create Record
var statisticalJournal = nlapiCreateRecord('STATISTICALJOURNALENTRY');
statisticalJournal.setFieldValue('subsidiary', subsidiaryId);
statisticalJournal.setFieldValue('externalid', externalId);
statisticalJournal.setFieldValue('unitstype', unitsTypeId);
statisticalJournal.setFieldValue('unit', unitId);

// Add line to the record
statisticalJournal.setLineItemValue('line', 'account', 1, statisticalAccountId);
statisticalJournal.setLineItemValue('line', 'debit', 1, amount); // field "debit" has label "Am
ount" in UI
statisticalJournal.setLineItemValue('line', 'lineunit', 1, unitId);
statisticalJournal.setLineItemValue('line', 'memo', 1, memo);
statisticalJournal.setLineItemValue('line', 'class', 1, classId);
statisticalJournal.setLineItemValue('line', 'department', 1, departmentId);
statisticalJournal.setLineItemValue('line', 'location', 1, locationId);

// Add record
var recId = nlapiSubmitRecord(statisticalJournal);

// Load record
var statisticalJournalAddedRec = nlapiLoadRecord('STATISTICALJOURNALENTRY', recId);

// Delete record
nlapiDeleteRecord('STATISTICALJOURNALENTRY', recId);
```

Updating the Subsidiary Field

The subsidiary field can be updated only in dynamic mode, as shown in the following sample.

```
var recType = 'StatisticalJournalEntry';

// 1st Load record in Dynamic Mode
var rec = nlapiLoadRecord(recType,'168', {recordmode: 'dynamic'});

// 2nd Remove all lines
for (i=1; i <= rec.getLineItemCount('line'); i++)
{
   rec.removeLineItem('line','1');
}

// 3rd Change subsidiary
rec.setFieldValue('subsidiary','6');

// 4th Add new line for changed subsidiary
// the new account must be available in the new subsidiary
rec.selectNewLineItem('line');
rec.setCurrentLineItemValue('line','account','286');
rec.setCurrentLineItemValue('line','debit','5');
rec.commitLineItem('line');

nlapiSubmitRecord(rec);
```

Store Pickup Fulfillment

Note: The content in this help topic pertains to all versions of SuiteScript.

The store pickup fulfillment record is used when you fulfill an order from a location that allows store fulfillment and the fulfillment choice of line items in the order is Store Pickup.

Note: If there are other line items in the sales order with the fulfillment choice set to Ship, you create an item fulfillment record for those lines. See Item Fulfillment for more information.

The internal ID for this record is `storepickupfulfillment`.

Supported Script Types

The store pickup fulfillment record is scriptable in both client and server SuiteScript.

Supported Functions

The store pickup fulfillment record supports the following functions: read, edit, transform, delete, and search.

Note: The create and copy functions are not supported.

Usage Notes

To use this record, you must have the Store Pickup feature enabled. See the help topic Store Pickup for more information about enabling and configuring the feature in NetSuite.

You should use the Store Pickup feature with the Fulfillment Request feature when possible. A fulfillment request is an intermediate record between the sales order and the store pickup fulfillment. With this intermediate record, you can better manage workload at a store location, as well as add fulfillment exceptions if there are problems fulfilling an item. With the Fulfillment Request feature enabled, you first create a sales order, then transform the sales order record to a fulfillment request record, and finally you transform the fulfillment request record to a store pickup fulfillment request record. See Example 1.

If the Fulfillment Request feature is not enabled, you transform the sales order record to a store pickup fulfillment record. See Example 2.

In both cases, the Fulfillment Choice field must be set to Store Pickup at the line-item level in the sales order.

Code Samples

Example 1: Create a store pickup fulfillment from a fulfillment request. The sales order ID is 321654. The fulfillment request ID is 221100 and is passed to the nlapiTransformRecord API as a hash table.

```
var fulfillmentRequestRecord = {"fftreqid" : 221100};
var storePickupFulfillRecord = nlapiTransformRecord("salesorder",321654, "storepickupfulfillmen
```

```
t", fulfillmentRequestRecord);
var storePickupFulfillRecordId = nlapiSubmitRecord(storePickupFulfillRecord);
```

Example 2: Create a store pickup fulfillment from a sales order.

```
var storePickupFulfillRecord = nlapiTransformRecord("salesorder",321654, "storepickupfulfillmen
t");
var storePickupFulfillRecordId = nlapiSubmitRecord(storePickupFulfillRecord);
```

Subscription

Note: The content in this help topic pertains to all versions of SuiteScript.

The subscription record is used to build recurring subscriptions.

The subscription record is available only when the SuiteBilling feature is enabled at Setup > Company > Enable Features (Administrator), on the Transactions subtab. When the feature is enabled, you can access the subscription record in the UI by going to Transactions > Subscriptions > Create Subscriptions.

The internal ID for this record is `subscription`.

See the SuiteScript Records Browser for all internal IDs associated with this record. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The subscription record is scriptable in server and client SuiteScript.

Supported Functions

The subscription record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Subscription Line

Note: The content in this help topic pertains to all versions of SuiteScript.

The subscription line record is used to provide subscription item information for subscriptions.

The subscription line record is available only when the SuiteBilling feature is enabled at Setup > Company > Enable Features (Administrator), on the Transactions subtab.

The internal ID for this record is `subscriptionline`.

See the SuiteScript Records Browser for all internal IDs associated with this record. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The subscription line record is scriptable in server and client SuiteScript.

Supported Functions

The subscription line record is not fully scriptable — create and delete are not supported. Subscription lines can be created and deleted only from a subscription plan in the user interface.

Time

Note: The content in this help topic pertains to all versions of SuiteScript.

A time transaction, also known as TimeBill, records the hours worked by an employee. This transaction can be used to record billable hours and invoice customers. This transaction is available when the Time Tracking feature is enabled.

For details about this type of transaction, see the help topic Managing Time Tracking.

The internal ID for this record is `timebill`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Type

The time record is scriptable in both client and server SuiteScript.

Supported Functions

The time record is fully scriptable, which means that the record can be created, updated, copied, deleted, and searched using SuiteScript.

Transaction Search

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Nearly all transaction record types use the TransactionSearch record for search.

The internal ID for this record is `transaction`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The transaction search is scriptable in both client and server SuiteScript.

Supported Functions

Only search is supported for the transaction search record.

Usage Notes

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
Search Filters and Search Columns		
ccnumber	Credit Card #	To prevent users from accessing sensitive information such as password and credit card data, this field cannot be read in beforeSubmit user event scripts for external role users (for example, shoppers, online form users (anonymous users), customer center).
entity	Name	The search filter entity is synonymous for the search filter name. Either filter can be used when searching the value of the Name / ID field in the UI.

Transfer Order

Note: The content in this help topic pertains to all versions of SuiteScript.

The Transfer Order transaction is used to move inventory between locations when the Multi-Location Inventory (MLI) feature is enabled. Existing integrations with external warehouse management systems can leverage this transaction to manage data about inventory movement between locations.

Transfer orders can initialize item fulfillment and item receipt transactions. See the help topics Item Fulfillment and Item Receipt for details about these transactions.

The internal ID for this record is `transferorder`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Usage Notes

Fields

estgrossprofit	Est. Gross Profit	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search.

		When this field appears on the sublist line level, this field is not scriptable.
estgrossprofitpercent	Est. Gross Profit Percent	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
totalcostestimate	Est. Extended Cost	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search.

Usage

Note: The content in this help topic pertains to all versions of SuiteScript.

The usage record is used to calculate usage information for subscriptions.

The usage record is available only when the SuiteBilling feature is enabled at Setup > Company > Enable Features (Administrator), on the Transactions subtab. When the feature is enabled, you can access the usage record in the UI by going to Transactions > Subscriptions > Create Usages.

The internal ID for this record is `usage`.

See the SuiteScript Records Browser for all internal IDs associated with this record. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The usage record does not support client scripts, server scripts, or workflows.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

The usage record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Vendor Bill

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The vendor bill transaction records payables as they arrive from vendors, allowing you to pay bills from the payables list as they are due, and providing an accurate picture of payables at all points of the billing cycle.

For details about this type of transaction, see the help topic Vendor Bills.

The internal ID for this record is `vendorbill`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The vendor bill record is scriptable in both client and server SuiteScript.

Supported Functions

The vendor bill record is fully scriptable, which means that the record can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
estgrossprofit	Est. Gross Profit	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
estgrossprofitpercent	Est. Gross Profit Percent	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
totalcostestimate	Est. Extended Cost	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search.
usertotal	Amount	This field is not available via search or lookup for any transactions.

Using Landed Cost Fields

When you create a landed cost category, the associated field IDs for the first category are landedcostamount1 and landedcostsource1. If you create a second category, the IDs will be landedcostamount2 and landedcostsource2.

This pattern increments by one with each additional category. For example, the IDs for the next landed cost category will be landedcostamount3 and landedcostsource3, and so on.

Using Bill Address

You can use the Bill Address field on the vendor bill. For more information about creating and accessing subrecords, see the help topic Working with Subrecords in SuiteScript.

Vendor Credit

Note: The content in this help topic pertains to all versions of SuiteScript.

A vendor credit transaction creates a credit, from a vendor, that can be applied to a payables account. For example, a vendor credit transaction may occur when items are returned to a vendor or when a discount is negotiated with a vendor.

For details about this type of transaction, see the help topic Vendor Credits.

The internal ID for this record is `vendorcredit`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The vendor credit record is scriptable in both client and server SuiteScript.

Supported Functions

The vendor credit record is fully scriptable, which means that the record can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
usertotal	Amount	This field is not available via search or lookup for any transactions.

Vendor Payment

> **Note:** The content in this help topic pertains to all versions of SuiteScript.

A vendor payment transaction posts to the general ledger as an expense and the amount of the payment is deducted from your accounts payable total. A vendor payment can be applied to one or more vendor bills. Vendor payments can help to track expenditures and total payables due.

For details about this type of transaction, see Vendor Payments.

The internal ID for this record is `vendorpayment`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The vendor payment record is scriptable in both client and server SuiteScript.

Supported Functions

The vendor payment record is fully scriptable, which means that the record can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

The Credits sublist, scriptable for the vendor payment record, is a **list** sublist. In the NetSuite Help Center, see the help topic List Sublists for information on this sublist type.

Vendor Return Authorization

Note: The content in this help topic pertains to all versions of SuiteScript.

A vendor return authorization is a non-posting transaction that tracks a return to a vendor, including the items to be returned, their quantities, the approval status, the shipment status, and the amount refunded or credited from the vendor. This type of transaction is available when the Vendor Return Authorizations feature is enabled.

The vendor return process includes four steps: creating a vendor return authorization record, approving or canceling the authorization, shipping items authorized to be returned, and crediting an authorized vendor return. For more details, see the help topic Vendor Return Authorization Overview.

The internal ID for this record is `vendorreturnauthorization`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The vendor return authorization record is scriptable in both client and server SuiteScript.

Supported Functions

The vendor return authorization record is fully scriptable, which means that the record can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
usertotal	Amount	This field is not available via search or lookup for any transactions.

Work Order

Note: The content in this help topic pertains to all versions of SuiteScript.

Work order transactions track the production of assembly items needed for stock or to fill orders. Work orders track the quantities of assemblies that need to be built and the quantities of components, or member items, needed to do so. This type of transaction is available when the Assembly Items and

Work Orders features are enabled and is used when the Allow Purchase of Assembly Items accounting preference is not enabled.

Special order work orders track assemblies for a particular sale, and Production work orders track assemblies to increase stock. Both use the same work order form, but Production work orders do not link to a sales transaction. Production work orders are generated when the back ordered quantity of an assembly reaches its assigned build point. After the build point is reached, a work order is added in the Mass Create Work Orders queue. For more details, see Assembly Work Orders.

The internal ID for this record is `workorder`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The work order record is scriptable in both client and server SuiteScript.

Supported Functions

The work order record is fully scriptable, which means that the record can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

Fields

estgrossprofit	Est. Gross Profit	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
estgrossprofitpercent	Est. Gross Profit Percent	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search. When this field appears on the sublist line level, this field is not scriptable.
totalcostestimate	Est. Extended Cost	When this field is on the body of the form in edit mode, this field is scriptable and can be returned in a transaction search.

The Multi-Partner Management feature must be enabled in your account for the Partners sublist to appear.

Work Order Close

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

If the Manufacturing Work In Process (WIP) feature has been enabled, you can use SuiteTalk to interact with work order close records. You can check to see if WIP is enabled by going to Setup > Company > Enable Features, and reviewing the Items & Inventory tab.

With WIP, instead of creating a single assembly build record to denote that a work order has been addressed, you track progress of the work using three records: work order issue, work order completion, and work order close. This approach lets you manage the assembly process in a more granular way, and to keep the General Ledger up to date as materials move through the different phases of assembly.

For details on the benefits of WIP, see the help topic Manufacturing Work In Process (WIP). For information about the work order close process in the UI, see the help topic Entering Work Order Closes. For information about the process of closing a work order without WIP enabled, see the help topic Marking Work Orders Closed.

The internal ID for this record is `workorderclose`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The work order close record is scriptable in server SuiteScript only.

Supported Functions

The work order close record is partially scriptable. It can be updated, deleted, and searched using SuiteScript. It cannot be copied or created.

Usage Notes

To use this record you must have the following features enabled: Work Orders and Manufacturing Work in Process.

In the UI, this record is accessed by going to Transactions > Manufactoring > Close Work Order.

Note these additional details:

- You must use nlapiTransformRecord to create a new instance of this record. In this case *workorder* is the originating record type. For more details, see “Prerequisites for Creating a Record,” below.
- Assembly item should have scrap account, WIP account, and WIP Cost Variance Account specified.
- nlapiCreateRecord and nlapiCopyRecord are not supported on this record.

Prerequisites for Creating a Work Order Issue Record

Before you can create a work order issue record, a work order record must already exist, and the work order must be configured to use WIP (the WIP box on the work order record must be selected). This is true regardless of whether you are creating the work order issue record using initialize and add, or add by itself. If you try to create a work order issue record referencing a work order that has not been configured to use WIP, the system generates an error reading in part, “One of the following problems exists: You have an invalid work order < work order ID >, the work order does not use WIP, or the work order is already closed.” You can create and modify work orders by choosing Transactions > Manufacturing > Enter Work Orders.

You can also interact with work orders using SuiteScript, as described in Work Order.

Note also that the assembly item referenced in the work order must be properly set up for WIP, as described in the Setting Up Items as WIP Assemblies.

Work Order Completion

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

If the Manufacturing Work In Process (WIP) feature has been enabled, you can use SuiteTalk to interact with work order completion records. You can check to see if WIP is enabled by going to Setup > Company > Enable Features, and reviewing the Items & Inventory tab.

With WIP, instead of creating a single assembly build record to denote that a work order has been addressed, you track progress of the work using three records: work order issue, work order completion, and work order close. This approach lets you manage the assembly process in a more granular way, and to keep the General Ledger up to date as materials move through the different phases of assembly. For more on the benefits of WIP, refer to Manufacturing Work In Process (WIP).

The work order completion record is used to indicate that assemblies have been built. You can also optionally use this record to record that raw materials — items in the componentList sublist — have been consumed as part of the assembly process. This latter option is called entering a completion with backflush. For example, you might enter a completion with backflush if previous records (such as work order issue) did not record the consumption of all the materials you ended up using.

In the UI, you can view the form used for creating the work order completion record by choosing Transactions > Manufacturing > Enter Completions, selecting a Subsidiary (for OneWorld accounts), then clicking the Complete link that corresponds with one of the listed work orders. An alternate method is to view the work order and click one of two buttons: Enter Completions or Enter Completions With Backflush. For help filling out the form manually, refer to Entering Work Order Completions.

The internal ID for this record is `workordercompletion`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The work order completion record is scriptable in server SuiteScript only.

Supported Functions

The work order completion record is partially scriptable. It can be updated, deleted, and searched using SuiteScript. It cannot be copied or created.

Usage Notes

To use this record you must have the following features enabled: Work Orders and Manufacturing Work in Process.

In the UI, this record is accessed by going to Transactions > Manufactoring > Enter Completions.

Note these additional details:

- You must use nlapiTransformRecord to create a new instance of this record. In this case *workorder* is the originating record type. For more details, see "Prerequisites for Creating a Record," below.
- Assembly item should have scrap account and WIP account specified.
- nlapiCreateRecord and nlapiCopyRecord are not supported on this record.
- The Component sublist is available only when backflush = true

Prerequisites for Creating a Work Order Issue Record

Before you can create a work order issue record, a work order record must already exist, and the work order must be configured to use WIP (the WIP box on the work order record must be selected). This is true regardless of whether you are creating the work order issue record using initialize and add, or add by itself. If you try to create a work order issue record referencing a work order that has not been configured to use WIP, the system generates an error reading in part, "One of the following problems exists: You have an invalid work order < work order ID >, the work order does not use WIP, or the work order is already closed." You can create and modify work orders by choosing Transactions > Manufacturing > Enter Work Orders.

You can also interact with work orders using SuiteScript, as described in Work Order.

Note also that the assembly item referenced in the work order must be properly set up for WIP, as described in the Setting Up Items as WIP Assemblies.

Work Order Issue

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

If the Manufacturing Work In Process (WIP) feature has been enabled, you can use SuiteTalk to interact with work order completion records. You can check to see if WIP is enabled by going to Setup > Company > Enable Features, and reviewing the Items & Inventory tab.

With WIP, instead of creating a single assembly build record to denote that a work order has been addressed, you track progress of the work using three records: work order issue, work order completion, and work order close. This approach lets you manage the assembly process in a more granular way, and to keep the General Ledger up to date as materials move through the different phases of assembly. For more on the benefits of WIP, refer to Manufacturing Work In Process (WIP).

The work order issue record is used to indicate that particular quantities of raw materials, or component items, have been gathered for the production of the assembly item (or items).

In the UI, you can view the form used for creating the work order issue record by choosing Transactions > Manufacturing > Issue Components, selecting a Subsidiary (for OneWorld accounts), then clicking the Issue link that corresponds with one of listed work orders. An alternate method is to view the work order and click the Issue Components button. For help filling out the form manually, refer to Entering Work Order Issues.

The internal ID for this record is `workorderissue`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Supported Script Types

The work order issue record is scriptable in server SuiteScript only.

Supported Functions

The work order issue record is partially scriptable. It can be updated, deleted, and searched using SuiteScript. It cannot be copied or created.

Usage Notes

To use this record you must have the following features enabled: Work Orders and Manufacturing Work in Process.

In the UI, this record is accessed by going to Transaction > Manufacturing > Issue Components.

Note these additional details:

- You must use nlapiTransformRecord to create a new instance of this record. In this case *workorder* is the originating record type. For more details, see "Prerequisites for Creating a Record," below.
- nlapiCreateRecord and nlapiCopyRecord are not supported on this record.

Prerequisites for Creating a Work Order Issue Record

Before you can create a work order issue record, a work order record must already exist, and the work order must be configured to use WIP (the WIP box on the work order record must be selected). This is true regardless of whether you are creating the work order issue record using initialize and add, or add by itself. If you try to create a work order issue record referencing a work order that has not been configured to use WIP, the system generates an error reading in part, "One of the following problems exists: You have an invalid work order < work order ID >, the work order does not use WIP, or the work order is already closed." You can create and modify work orders by choosing Transactions > Manufacturing > Enter Work Orders.

You can also interact with work orders using SuiteScript, as described in Work Order.

Note also that the assembly item referenced in the work order must be properly set up for WIP, as described in Setting Up Items as WIP Assemblies.

Support

Note: The content in this help topic pertains to all versions of SuiteScript.

The following records are scriptable in SuiteScript:

- Case
- Issue

- Issue Product
- Issue Product Version
- Solution
- Topic

Case

Note: The content in this help topic pertains to all versions of SuiteScript.

The internal ID for this record is `supportcase`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The case record is scriptable in both server and client SuiteScript.

Supported Functions

The case record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Change to Case Editing as of Version 2013 Release 2

Prior to Version 2013 Release 2, two users could edit the same case simultaneously and then both could save their changes. As of Version 2013 Release 2, case records have the same restrictions as other records. If a script is editing a case, and another user or script edits and saves the same case record, the script is unable to save the record. Any existing SuiteScript that updates cases even when they are open for editing by another user may need to be updated to continue to operate as intended.

Issue

Note: The content in this help topic pertains to all versions of SuiteScript.

For information about working with the issue record in the user interface, see the help topic Logging Issues.

The internal ID for this record is `issue`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The issue record is scriptable in both server and client SuiteScript.

Supported Functions

The issue record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Issue Product

The use of Products on issue records helps determine which product team is responsible for the resolution of that issue.

For information about working with the issue product record in the user interface, see the help topic Products and Modules.

The internal ID for this record is `issueproduct`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Supported Script Types

The issue record is scriptable in both server and client SuiteScript.

Supported Functions

The record can be created, read, updated and deleted using SuiteScript. It cannot be copied, transformed or searched.

Code Sample

The following script example illustrates how to create an IssueProduct record.

```
var product = nlapiCreateRecord('issueproduct');
product.setFieldValue('issueproduct', 'new product');
var productId = nlapiSubmitRecord(product);
```

Issue Product Version

If you track multiple versions of the same product, you can create versions and builds using the product version record.

For information about working with issue product versions in the user interface, see the help topic Product Versioning.

The internal ID for this record is `productversion`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Supported Script Types

The issue record is scriptable in both server and client SuiteScript.

Supported Functions

The record can be created, read, updated and deleted using SuiteScript. It cannot be copied, transformed or searched.

Code Sample

The following script example illustrates how to create an IssueProductVersion record for a previously created IssueProduct.

```
var version = nlapiCreateRecord('issueproductversion');
version.setFieldValue('issueproduct', productId);
version.setFieldValue('productversion', 'New version');

Var versionId = nlapiSubmitRecord(version);
```

The following script example illustrates how to add IssueProductBuild to IssueProductVersion.

```
var version = nlapiLoadRecord('issueproductversion', versionId);
version.selectNewLineItem('availablebuild');
version.setCurrentLineItemValue('availablebuild', 'build', 'new build');
version.setCurrentLineItemValue('availablebuild', 'description', 'Build description');

version.commitLineItem('availablebuild');
nlapiSubmitRecord(version);
```

Solution

Note: The content in this help topic pertains to all versions of SuiteScript.

The internal ID for this record is `solution`.

This record is available when the Knowledge Base feature is enabled at Setup > Company > Enable Features, on the CRM tab. When the feature is enabled, you can access the inventory cost revaluation record in the UI by choosing Lists > Support > Solutions > New.

For details about working with this record manually, see the help topic Creating Knowledge Base Solutions.

See the SuiteScript Records Browser for all internal IDs associated with this record. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The solution record is scriptable in both server and client SuiteScript.

Supported Functions

The solution record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Topic

Note: The content in this help topic pertains to all versions of SuiteScript.

The internal ID for this record is *topic*.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The topic record is scriptable only in server SuiteScript.

Supported Functions

The topic record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

File Cabinet

Note: The content in this help topic pertains to all versions of SuiteScript.

The following records are scriptable in SuiteScript:

- File
- Folder

File

Note: The content in this help topic pertains to all versions of SuiteScript.

The internal ID for this record is `file`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The file cabinet record is scriptable in server SuiteScript only.

Supported Functions

This record is not fully scriptable. Only search is permitted.

Usage Notes

Client SuiteScript is not supported for this record. It is scriptable in server SuiteScript only.

The file record is different from other NetSuite records. It has its own SuiteScript API. For information about the SuiteScript 1.0 API for files, see the help topic File APIs. for information about the SuiteScript 2.0 API for files, see the help topic N/file Module.

Folder

Note: The content in this help topic pertains to all versions of SuiteScript.

The internal ID for this record is `folder`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The folder record is scriptable in server SuiteScript only.

Supported Functions

The folder record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Lists

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The following list records are scriptable in SuiteScript:

- Account
- Accounting Book
- Accounting Context
- Accounting Period
- Allocation Schedule
- Amortization Schedule
- Amortization Template
- Billing Account
- Billing Class
- Billing Rate Card
- Billing Schedule
- Bin
- BOM
- BOM Revision
- Class
- Consolidated Exchange Rate

- Contact Category
- Contact Role
- Cost Category
- Currency
- Customer Category
- Customer Message
- Customer-Subsidiary Relationship
- Department
- Fair Value Price
- Gift Certificate
- Global Account Mapping
- Government-Issued ID Tracking
  - Driver's License
  - Passport
  - Other Government-Issued ID
- Group
- Inbound Shipment
- Intercompany Allocation Schedule
- Inventory Number
- Item Account Mapping
- Item Location Configuration
- Item Revision
- Kudos Feature Records
  - Kudos Feature Records
  - Organization Value
- Location
- Manufacturing Cost Template
- Manufacturing Routing
- Nexus
- Note Type
- Other Name Category
- Ownership Transfer
- Partner Category
- Payment Method
- Payroll Item
- Price Book
- Price Level
- Price Plan
- Pricing Group
- Project Charge Rule

- Project Expense Type
- Project Revenue Rule
- Revenue Recognition Event
- Revenue Recognition Plan
- Revenue Recognition Schedule
- Revenue Recognition Template
- Role
- Sales Role
- Sales Tax Item
- Subsidiary
- System Note
- Tax Control Account
- Tax Group
- Tax Period
- Tax Type
- Term
- Unit of Measure
- Vendor Category
- Vendor-Subsidiary Relationship
- Workplace

Account

Note: The content in this help topic pertains to all versions of SuiteScript.

For details about working with the account record in the user interface, see the help topic Working with the Chart of Accounts.

The internal ID for this record is `account`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The accounting book record is supported in client SuiteScript only.

Supported Functions

The account record is fully scriptable — it can be created, updated, copied, deleted, and searched using Client SuiteScript.

Accounting Book

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

When the Multi-Book Accounting feature has been enabled, you can use the accounting book record to create secondary books. For details, see the help topic Using Multi-Book Accounting.

The internal ID for this record is `accountingbook`.

In the user interface, you access the accounting book record at Setup > Accounting > Accounting Books > New.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The accounting book record is supported in server SuiteScript only.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

This record is partially scriptable — it can be created, updated, deleted, and searched using SuiteScript.

Code Samples

The following code snippets show how to create accounting book records and perform other basic tasks.

```
// Create Record

var rec = nlapiCreateRecord('accountingbook', null);
rec.setFieldValue('name', 'test');

var subs = new Array();
```

```
subs[0] = '1';
subs[1] = '3';
subs[2] = '5';

rec.setFieldValues('subsidiary', subs);
rec.setFieldValue('isprimary', 'T');

var recCreated = nlapiSubmitRecord(rec);

var id = 0;


// Update Record
var rec = nlapiLoadRecord('accountingbook', '3');

var subs = new Array();
subs[0] = '1';
subs[1] = '3';
subs[2] = '5';

rec.setFieldValues('subsidiary', subs);
rec.setFieldValue('isprimary', 'T');

var recUpdated = nlapiSubmitRecord(rec);

var id = 0;


// Delete Record
var rec = nlapiDeleteRecord('accountingbook', '2');
```

Accounting Context

Note: The content in this help topic pertains to all versions of SuiteScript.

An accounting context can be a one-to-one relationship between a country's local GAAP (Generally Accepted Accounting Principles) reporting requirements and a statutory chart of accounts (COA). It can also be a unique relationship that meets your company's specific needs. Accounting contexts are useful when users prefer to work in a local GAAP context, rather than in the consolidated context with one centralized COA. Accounting contexts are also useful if you have Multi-Book Accounting provisioned in your account. You can set an accounting context specific for your secondary book and use it for your secondary book reports. For more information about accounting contexts, see the help topic Accounting Contexts.

The internal ID for this record is `accountingcontext`.

The accounting context record is available in OneWorld accounts.

The script user must have the Setup Company permission with Full permission level.

In the UI, you configure accounting contexts on the Accounting Contexts subtab on the General Preferences page at Setup > Company > General Preferences.. If you do not have a OneWorld account, this subtab is not visible.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

This record is scriptable in both client and server SuiteScript.

Supported Operations

This record is partially scriptable — it can be read, created, deleted, copied, searched, and edited. It cannot be transformed.

Code Samples

The following code snippets show how to create an accounting context.

```
// SS 1.0
var rec = nlapiCreateRecord("accountingcontext");
rec.setFieldValue("name", "Context 1");
nlapiSubmitRecord(rec);
});
```

```
// SS 2.0
require(["N/record"], function(record){
  var rec = record.create({type: "accountingcontext"});
  rec.setValue({fieldId: "name", value: "Context 1"});
  rec.save();
});
```

Accounting Period

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

When the Accounting Periods feature has been enabled, this record is available to SuiteScript.

In the user interface, you access accounting periods at Setup > Accounting > Manage Accounting Periods.

The internal ID for this record is `accountingperiod`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The accounting period record is supported in client and server SuiteScript, for all script types.

Supported Functions

Only search and read functions are supported.

Code Sample

```
var period = nlapiLoadRecord('accountingperiod', <internal ID>;
var startDate = period.getFieldValue('startdate');
var endDate = period.getFieldValue('enddate');
```

Allocation Schedule

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Allocation schedules transfer balances from expense accounts into one or more other accounts.

The internal ID for this record is `allocationschedule`.

The allocation schedule record is available only when the Accounting Periods and Expense Allocation features are enabled. An administrator can enable these features at Setup > Company > EnableFeatures > Accounting subtab. For information about expense allocation, see the help topic Expense Allocation Overview. For details on working with the allocation schedule in the user interface, see the help topic Creating Expense Allocation Schedules.

If the Statistical Accounting feature is enabled, the allocation schedule record includes fields that enable you to base the weight for the allocation on the balance of a statistical account through statistical journals or as an absolute value. If the Dynamic Allocation feature is also enabled, the weight is dynamically calculated when the allocation journal is generated. An administrator can enable these features at Setup > Company > EnableFeatures > Accounting under Advanced Features. For information about statistical allocation schedules, see the help topic Working with Allocation Schedules Weighted by the Balance of a Statistical Account.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

This record is scriptable in both client and server SuiteScript.

If the Dynamic Allocation feature is enabled, the currency field is not scriptable. None of the fields from the History subtab on the allocation schedule record in the user interface are not scriptable.

Supported Operations

This record is partially scriptable — it can be read, created, deleted, and updated. It cannot be copied or transformed.

Code Samples

The following code snippets show how to create a statistical allocation schedule that divides the balance of one account and transfers the amounts into two other accounts.

```
var record = nlapiCreateRecord("allocationschedule", false)

record.setFieldValue("name", "Allocation Schedule A");
record.setFieldValue("subsidiary", "1");

record.setLineItemValue("allocationsource", "account", 1, "24");

record.setLineItemValue("allocationdestination", "account", 1, "6");
record.setLineItemValue("allocationdestination", "weight", 1, "50.00");

record.setLineItemValue("allocationdestination", "account", 1, "32");
record.setLineItemValue("allocationdestination", "weight", 1, "50.00");

recordId = nlapiSubmitRecord(record);
```

Amortization Schedule

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Amortization schedules are automatically created by the system for transactions that contain item or expense lines associated with amortization templates.

The internal ID for this record is `amortizationschedule`.

The amortization schedule record is available only when the Amortization feature is enabled, at Setup > Enable Features, on the Accounting subtab. In the user interface, you access this record at Lists > Accounting > Amortization Schedules. For details on working with this record in the user interface, see the help topic Amortization Schedule Overview.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

This record is scriptable in server SuiteScript only.

The user events are not supported.

Supported Functions

This record is partially scriptable — it can be read, searched, and edited. It cannot be copied, created, deleted, or transformed.

> **Note:** Edits are allowed only when the Allow Users to Modify Amortization Schedules preference has been selected. For details, see the help topic Setting Amortization Preferences.

Usage Notes

Be aware of the following:

- As in the user interface, you cannot use external ID as part of your criteria when searching for amortization schedule records. You also cannot include external ID in search columns.
- The amortization schedule record is similar to the revenue recognition schedule record — so if you have existing integrations for revenue recognition schedules, you may be able to reuse elements of these scripts. The revenue recognition schedule record is described in Revenue Recognition Schedule.

Amortization Template

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

You use the amortization template record to define the terms of amortization schedules, which are automatically created by the system. For example, on certain transactions, you can associate an amortization template with a line in the item or expense sublist. When the transaction is saved, an amortization schedule based on the template is automatically created. You can use amortization templates in conjunction with journal entry, vendor bill, and vendor credit records.

The internal ID for this record is `amortizationtemplate`.

The amortization template record is available only when the Amortization feature is enabled, at Setup > Enable Features, on the Accounting subtab. In the user interface, you access this record at Lists >

Accounting > Amortization Templates > New. For help working with amortization templates in the user interface, see the help topic Creating Amortization Templates.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The amortization template record is supported in server SuiteScript only.

The user events are not supported.

Supported Operations

This record is fully scriptable, which means that the record can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

Be aware of the following:

- If the Method field (recurrencetype) is set to Custom, then you must include at least one line in the recurrence sublist. In this case, the template cannot be deleted.
- As in the user interface, you cannot use external ID as part of your criteria when searching for amortization template records. You also cannot include external ID in search columns.
- The amortization template record is similar to the revenue recognition template record — so if you have existing integrations for revenue recognition templates, you may be able to reuse elements of these scripts. The revenue recognition template record is described in Revenue Recognition Template.

Code Samples

The following code snippets show how to create amortization templates and perform other basic tasks.

```
// Create Record
var recAmTemp = nlapiCreateRecord('AmortizationTemplate');
recAmTemp.setFieldValue('name', 'Name');
recAmTemp.setFieldValue('externalid', 'externalId');
recAmTemp.setFieldValue('isamortization','T');
recAmTemp.setFieldValue('amortizationtype','STANDARD' ); // Type
recAmTemp.setFieldValue('recurrencetype', 'EVENPERIODSPRORATE' ); // Method
recAmTemp.setFieldValue('recogintervalsrc', 'RECEIPTDATE' ); // Term Source
```

```
recAmTemp.setFieldValue('revrecoffset', 0);
recAmTemp.setFieldValue('periodoffset', 1);
recAmTemp.setFieldValue('acctdeferral','1' );
recAmTemp.setFieldValue('acctcontra', '1');
recAmTemp.setFieldValue('accttarget', '1');
recAmTemp.setFieldValue('amortizationperiod',20);
recAmTemp.setFieldValue('residual','1.00' );
recAmTemp.setFieldValue('initialamount','12.00' );
recAmTemp.setFieldValue('isinactive', 'F');
var recId = nlapiSubmitRecord(recAmTemp);

// Load Record
var rec = nlapiLoadRecord('AmortizationTemplate',recId);

// Copy
var recCopied = nlapiCopyRecord('AmortizationTemplate', recId);

// Delete Record
nlapiDeleteRecord('AmortizationTemplate',recId);
```

Billing Account

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The internal ID for this record is `billingaccount`. Full support is provided for client and server scripting with this record.

For more information on billing accounts, see the help topic Billing Accounts.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The billing account record is scriptable in both client and server SuiteScript.

Supported Functions

The billing account record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Billing Class

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

You can use the billing class record to create different rates that you can use when calculating the cost of a resource's time on a project. You can use SuiteScript to create, update, and delete billing class records.

The internal ID for this record is `billingclass`.

The billing class record is available when the Per-Employee Billing Rates feature is enabled at Setup > Company > Enable Features, on the Employees tab. When the feature is enabled, you can access the billing class record in the UI by choosing Setup > Accounting > Billing Classes > New.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The billing class record is scriptable in server SuiteScript only.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

The billing class record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Field Definitions

See the SuiteScript Records Browser for all internal IDs associated with this record. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Code Samples

The following sample shows how to create a billing class record.

```
var record = nlapiCreateRecord('billingclass');
record.setFieldValue('name', 'Billing Class');
record.setFieldValue('description', 'Billing Class Description');
```

```
record.setFieldValue('isinactive', 'F');
record.setLineItemValue('pricecost', 'price', 1, 2.56);
```

The following sample shows how to delete a billing class record.

```
nlapiDeleteRecord('billingclass', recId);
```

Billing Rate Card

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The internal ID for this record is `billingratecard`.

If you use billing classes, you can also use billing rate cards to define different billing rates for groups of billing classes. These rate cards can then be used to set billing rates on charge-based projects using time-based charge rules. You can use SuiteScript to read, create, update, delete, and search billing rate card records.

The billing rate card record is available when the Billing Rate Cards feature is enabled at Setup > Company > Enable Features, on the Employees tab. The Per-Employee Billing Rates and Charge-Based Billing features are also required to use billing rate cards. When the feature is enabled, you can access the billing rate card record in the UI by choosing Setup > Accounting > Billing Rate Cards > New.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The billing account record is scriptable in both client and server SuiteScript.

Supported Functions

The following SuiteScript actions are supported for this record:

- Read
- Create
- Edit
- Delete
- Search

Usage Notes

There are two sublists for rate card pricing: ratecardpricing and ratecardpricingmulti. These two lists are mutually exclusive. The first list, ratecardpricing, is a simple sublist used when the Multiple Currencies feature is not enabled. The second list, ratecardpricingmulti, is a matrix sublist used when the Multiple Currencies feature is enabled.

When this feature is enabled, the content of ratecardpricing is a subset of ratecardpricingmulti. If you enter information into ratecardpricing and then enable the Multiple Currencies feature, the original content is in a column in the ratecardpricingmulti sublist.

Note: Copy and transform are not supported.

Code Samples

The following sample shows a client script to create a new billing rate card with a single billing class without multiple currencies.

```
b = nlapiCreateRecord('billingratecard')
b.setFieldValue('name', 'abcd');
b.setLineItemValue('ratecardpricing', 'price', 1, 11);
nlapiSubmitRecord(b);
```

The following sample shows a client script to update the billing rate card name and a price in British pounds when multiple currencies is enabled.

```
b = nlapiLoadRecord('billingratecard', 3);
b.setFieldValue('name', b.getFieldValue('name') + ' - updated');
b.setLineItemValue('ratecardpricingmulti', 'price_2_', 1, 5);
nlapiSubmitRecord(b);
```

The following sample shows a user event script to update a price.

```
  var brc = nlapiGetNewRecord();

  var origPrice = brc.getLineItemValue('ratecardpricing', 'price', 1);
  var newPrice = 11;
  brc.setLineItemValue('ratecardpricing', 'price', 1, newPrice);
  nlapiLogExecution('DEBUG', 'price changed from ' + origPrice + ' to ' + newPrice);
}
```

Billing Schedule

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

This record enables you to create a billing schedule that can be applied to a sales order, a line item on a sales order, or a project. SuiteScript supports all five types of billing schedules (charge-based, fixed bid interval, fixed bid milestone, standard, and time and materials).

The internal ID for this record is `billingschedule`.

To use the billing schedule record, you must enable the Advanced Billing feature, at Setup > Enable Features, on the Transactions subtab. In the UI, you access this record at Lists > Accounting > Billing Schedules > New. You can access a billing schedule of fixed bid milestone type through the project record's Financial tab.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

Billing Schedule is scriptable in server SuiteScript only.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

This record is fully scriptable, which means that the record can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

Additionally, note the following:

- To set a value for schedule type, which is a required body field, you must use initializeValues, not setFieldValue. For examples, see the following section.
- If you choose a schedule type of fixed bid milestone, you must identify an existing project record (or job record). You do so using initializeValues. Then, to create a link between the project and the new billing schedule, you must update the project record — this relationship is not established automatically when you create the billing schedule.
- The Recurrence sublist is available only when schedule type is set to Standard and frequency to Custom.
- The Milestone sublist is available only when schedule type is set to Fixed Bid Milestone.

Code Samples

The following examples show how to create different types of billing schedules.

Creating a Charge-Based Billing Schedule

The following sample shows how to create a charge-based billing schedule.

```
var RECORD_TYPE = 'billingschedule';
var SCHEDULE_TYPE = 'CB'; //Charge-Based
```

```
var recId = null;
var recName = SCHEDULE_TYPE + "record";

var initValues = new Array();
initValues.schedtype = SCHEDULE_TYPE;

//Create record
var bs = nlapiCreateRecord(RECORD_TYPE, initValues);

bs.setFieldValue('externalid', 'EXTID001');
bs.setFieldValue('name', recName);
bs.setFieldValue('frequency', 'DAILY');
bs.setFieldValue('dayperiod', '3');
recId =  nlapiSubmitRecord(bs);
```

Creating a Fixed Bid Milestone Billing Schedule

The following sample shows how to create a billing schedule of fixed bid milestone type. Note that this sample references a particular project record during the creation of the billing schedule. However, you still have to establish the relationship between the billing schedule and the project as a separate step.

```
var RECORD_TYPE = 'billingschedule';
var SCHEDULE_TYPE = 'FBM'; //Fixed Bid, Milestone
var P1 = '117';   //Project1
var P1M1 = '112';   //Project1 - Milestone1
var P1M2 = '113';   //Project2 - Milestone2

var recId = null;
var recName = SCHEDULE_TYPE + " record";

var initValues = new Array();
initValues.schedtype = SCHEDULE_TYPE;
initValues.project = P1;

//Create the record
var bs = nlapiCreateRecord(RECORD_TYPE, initValues);

bs.setFieldValue('name', recName);
bs.setFieldValue('initialamount', '10%');

//Create the sublist
bs.selectNewLineItem('milestone');
bs.setCurrentLineItemValue('milestone', 'milestoneamount', '25%');
bs.setCurrentLineItemValue('milestone', 'milestonedate', '11/21/2013');
bs.setCurrentLineItemValue('milestone', 'projecttask', P1M1);
bs.commitLineItem('milestone');
recId = nlapiSubmitRecord(bs);

//Update the project (link it with the newly created Billing Schedule)
var project = nlapiLoadRecord('job', P1);
project.setFieldValue('jobbillingtype', SCHEDULE_TYPE);
project.setFieldValue('billingschedule', recId);
var pId = nlapiSubmitRecord(project);
```

Bin

Note: The content in this help topic pertains to all versions of SuiteScript.

Bins help you track on-hand quantities within a warehouse. Tracking items by bins can help organize receiving items and simplify picking items to fulfill orders.

For example, if you use bins, when you receive a purchase order, the order can tell you which bin to put the items away in. Then, your stock level of that item in that bin is tracked.

To use the bin record, the Bin Management feature must be enabled at Setup > Company > Setup Tasks > Enable Features. On the Items & Inventory tab, check the Bin Management box.

For details about using the bin record in the user interface, see the help topic Bin Management.

The internal ID for this record is `bin`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

Client SuiteScript is not supported for the bin record. It is scriptable in server SuiteScript only.

Supported Functions

The bin record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

BOM

Note: The content in this help topic pertains to all versions of SuiteScript.

A Bill of Materials, or BOM, lists the quantities of raw materials, assemblies, sub-components, and parts needed to manufacture a product. A BOM can be used to communicate between manufacturing partners, multiple facilities within the organization, or with a single manufacturing plant. The BOM that is scriptable in NetSuite is the Advanced BOM.

This record is supported when the Advanced BOM feature is enabled at Setup > Company > Enable Features, on the Items & Inventory tab. After this feature has been enabled, the Advanced BOM replaces the Assembly/Bill of Materials record. For information about this feature, see the help topic Advanced Bill of Materials.

In the UI, this record is accessed by going to Lists > Supply Chain > Bill of Materials.

The internal ID for this record is `bom`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The BOM record is scriptable in client and server SuiteScript.

Supported Functions

The BOM record is fully scriptable, which means that it can be read, created, edited, copied, deleted, and searched using SuiteScript.

BOM Code Samples

- Create a BOM
- Copy a BOM
- Delete a BOM
- Full BOM Code Sample

Create a BOM

The following example shows how to create a new bill of materials record.

```
require(['N/record'],
   function(record)
   {
      function createBom(values)
      {
         var rec = record.create({type: 'bom', isDynamic: true});

         for (var key in values) {
            if (values.hasOwnProperty(key)) {
               rec.setValue({
                  fieldId: key,
                  value: values[key]
               });
            }
         }

         return rec.save({enableSourcing: false, ignoreMandatoryFields: false});
      }

      var bomId = createBom({
```

```
        'name': 'SS 2.0 BoM',
        'subsidiary': [4,5],
        'includechildren' : true,
        'usecomponentyield': true
      });
   });
```

Copy a BOM

The following example shows how to copy a bill of materials record.

```
require(['N/record'],
    function(record)
   {
      function copyBom(bomId, values)
      {
         var rec = record.copy({type: 'bom', id: bomId, isDynamic: true});

         for (var key in values) {
            if (values.hasOwnProperty(key)) {
               rec.setValue({
                  fieldId: key,
                  value: values[key]
               });
            }
         }

         return rec.save({enableSourcing: false, ignoreMandatoryFields: false});
      }

      var bomId = copyBom(1, {
         'name': 'SS 2.0 BoM Copy',
         'subsidiary': [4],
         'includechildren' : false,
         'usecomponentyield': false
      });
   });
```

Delete a BOM

The following example shows how to delete a bill of materials record.

```
require(['N/record'],
   function(record)
   {
      function deleteBom(bomId)
      {
         record.delete({type: 'bom', id: bomId});
      }

      deleteBom(2);
   });
```

Full BOM Code Sample

The following example shows how to create, copy, and delete a bill of materials record, including edits to BOM revisions. For samples specific to BOM revisions, see BOM Revision Code Samples.

```
require(['N/record', 'N/search'],
   function(record, search)
   {
      function createBom(values)
      {
         var rec = record.create({type: 'bom', isDynamic: true});

         for (var key in values) {
            if (values.hasOwnProperty(key)) {
               rec.setValue({
                  fieldId: key,
                  value: values[key]
               });
            }
         }

         return rec.save({enableSourcing: false, ignoreMandatoryFields: false});
      }

      function createBomRevision(values, components)
      {
         var rec = record.create({type: 'bomrevision', isDynamic: true});

         for (var key in values) {
            if (values.hasOwnProperty(key)) {
               rec.setValue({
                  fieldId: key,
                  value: values[key]
               });
            }
         }

         for (var i in components) {
            rec.selectNewLine({sublistId: 'component'});

            for (var key in components[i]) {
               if (components[i].hasOwnProperty(key)) {
                  rec.setCurrentSublistValue({
                     sublistId: 'component',
                     fieldId: key,
                     value: components[i][key]
                  });
               }
            }

            rec.commitLine({sublistId: 'component'});
         }

         return rec.save({enableSourcing: false, ignoreMandatoryFields: false});
      }
```

```
function updateBomRevision(revisionId, values)
{
   var rec = record.load({type: 'bomrevision', id: revisionId, isDynamic: true});

   for (var key in values) {
      if (values.hasOwnProperty(key)) {
         rec.setValue({
            fieldId: key,
            value: values[key]
         });
      }
   }

   return rec.save({enableSourcing: false, ignoreMandatoryFields: false});
}

function copyBom(bomId, values)
{
   var rec = record.copy({type: 'bom', id: bomId});

   for (var key in values) {
      if (values.hasOwnProperty(key)) {
         rec.setValue({
            fieldId: key,
            value: values[key]
         });
      }
   }

   return rec.save({enableSourcing: false, ignoreMandatoryFields: false});
}

function deleteBom(bomId)
{
   record.delete({type: 'bom', id: bomId});
}

function getBomRevisions(bomId)
{
   //search ids for all revisions for given BoM
   var searchObj = search.create({
      type: 'bomrevision',
      filters: [
         ['billofmaterials', 'is', [bomId]]
      ],
      columns: ['name']
   });

   var revisions = [];
   searchObj.run().each(function(result) {
      revisions.push({'id': result.id, 'name': result.getValue('name')});
      return true;
   });
   return revisions;
```

```
}

/* create new BoM */

var bom1Id = createBom({
   'name': 'SS 2.0 BoM',
   'subsidiary': [4,5],
   'includechildren': true,
   'usecomponentyield': true
});

/* create new BoM Revision */

var bom1Rev1Id = createBomRevision({
   'billofmaterials': bom1Id,
   'name': 'Rev.1',
   'effectivestartdate': new Date('1/1/2017')
}, [
   {'item': 63, 'bomquantity': 1},
   {'item': 64, 'bomquantity': 2}
]);

/* update BoM Revision Effective End Date to prevent a collision */

bom1Rev1Id = updateBomRevision(bom1Rev1Id, {
   'effectiveenddate': new Date('12/31/2018')
});

/* create another BoM Revision */

var bom1Rev2Id = createBomRevision({
   'billofmaterials': bom1Id,
   'name': 'Rev.2',
   'effectivestartdate': new Date('1/1/2019')
}, [
   {'item': 64, 'bomquantity': 3},
   {'item': 62, 'bomquantity': 1}
]);

/* copy BoM and update some values */

var bom2Id = copyBom(bom1Id, {
   'name': 'SS 2.0 BoM Copy',
   'subsidiary': [4],
   'usecomponentyield': false,
   'includechildren': false
});

/* find BoM Revisions of BoM */

var revisions = getBomRevisions(bom1Id);

/* update first component's Component Yield value of each found BoM Revision */

for (var i = 0; i < revisions.length; i++) {
```

```
        rec = record.load({type: 'bomrevision', id: revisions[i].id});
        rec.setSublistValue({
           sublistId: 'component',
           fieldId: 'componentyield',
           line: 0,
           value: 50
        });
        rec.save();
     }

     /* delete BoMs */

     deleteBom(bom1Id);
     deleteBom(bom2Id);
  });
```

BOM Revision

Note: The content in this help topic pertains to all versions of SuiteScript.

A BOM revision enables you to update a BOM's details throughout the product lifecycle. A BOM revision also provides an accessible revisions history. Use revisions to compare and track cost savings when many BOM revisions are used in production. This record is available when the Advanced BOM feature is enabled at Setup > Company > Enable Features, on the Items & Inventory tab. For information about this feature, see the help topic Advanced Bill of Materials.

In the UI, this record is accessed by going to Lists > Supply Chain > Bill of Materials.

The internal ID for this record is `bomrevision`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The BOM Revision record is scriptable in server SuiteScript only.

Supported Functions

The BOM Revision record is fully scriptable, which means that it can be read, created, edited, copied, deleted, and searched using SuiteScript.

BOM Revision Code Samples

- Create a BOM Revision
- Search for a BOM Revision
- Update a BOM Revision
- Full BOM Code Sample

Create a BOM Revision

The following example shows how to create a new bill of materials revision record.

```
require(['N/record'],
   function(record)
   {
      function createBomRevision(values, items)
      {
         var rec = record.create({type: 'bomrevision', isDynamic: true});

         for (var key in values) {
            if (values.hasOwnProperty(key)) {
               rec.setValue({
                  fieldId: key,
                  value: values[key]
               });
            }
         }

         for (var itemIndex in items) {
            rec.selectNewLine({sublistId: 'component'});

            for (var key in items[itemIndex]) {
               if (items[itemIndex].hasOwnProperty(key)) {
                  rec.setCurrentSublistValue({
                     sublistId: 'component',
                     fieldId: key,
                     value: items[itemIndex][key]
                  });
               }
            }

            rec.commitLine({sublistId: 'component'});
         }

         return rec.save({enableSourcing: false, ignoreMandatoryFields: false});
      }

      var bomRevisionId = createBomRevision({
         'billofmaterials': 1,
         'name': 'Rev.1',
         'effectivestartdate': new Date('1/1/2017')
      }, [
         {'item': 63, 'bomquantity': 1},
         {'item': 62, 'bomquantity': 2}
```

```
      ])
   });
```

Search for a BOM Revision

The following example shows how to search for a bills of materials revision record.

```
require(['N/search'],
   function(search)
   {
      function getBomRevisions(bomId)
      {
         //search ids for all revisions for given BoM
         var searchObj = search.create({
            type: 'bomrevision',
            filters: [
               [name: 'billofmaterials', operator: 'is', values: [bomId]]
            ],
            columns: ['name']
         });

         var revisions = [];
         searchObj.run().each(function(result) {
            revisions.push({'id': result.id, 'name': result.getValue('name')});
            return true;
         });
         return revisions;
      }

      var revisions = getBomRevisions(1);
   });
```

Update a BOM Revision

The following example shows how to update a bills of materials revision record.

```
require(['N/record'],
   function(record)
   {
      function updateBomRevision(revisionId, values)
      {
         var rec = record.load({type: 'bomrevision', id: revisionId});

         for (var key in values) {
            if (values.hasOwnProperty(key)) {
               rec.setValue({
                  fieldId: key,
                  value: values[key]
               });
            }
         }

         return rec.save({enableSourcing: false, ignoreMandatoryFields: false});
```

```
    }

    updateBomRevision(1, {
       'effectiveenddate': new Date('12/31/2018')
    });
});
```

Class

Note: The content in this help topic pertains to all versions of SuiteScript.

Classes are categories that you can create to track records such as financials, transactions, and employees. For example, you own a janitorial service and you want to track income and expenses separately for household and commercial accounts. You set up a class for each account type and track the financial performance of each class over any time period.

For details about working with the class record in the user interface, see the help topic Creating Classes.

The internal ID for this record is `classification`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The class record is scriptable in both client and server SuiteScript.

Supported Functions

The class record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Consolidated Exchange Rate

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

If you use NetSuite OneWorld and your subsidiaries have different base currencies, you can maintain a table of consolidated exchange rates. This table is used to ensure that for consolidation purposes, currency amounts properly roll up from child to parent subsidiaries. Each consolidated exchange

rate translates between the base currency of a subsidiary and the base currency of its parent or grandparent subsidiary, for a specified accounting period. Consolidated exchange rates include three different rate types per period and subsidiary pair: Current, Average, and Historical.

The internal ID for this record is `consolidatedexchangerate`.

The consolidated exchange rate record is available only when the Multiple Currencies feature is enabled. An administrator can enable this feature at Setup > Company > EnableFeatures > Company subtab. For information about consolidated exchange rates, see the help topic Using Consolidated Exchange Rates.

The script user must have the Currency permission with Full permission level. To edit the different rate types per period and subsidiary pair, the script user must also have access to both the From and To subsidiary.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The consolidated exchange rate record is scriptable in both client and server SuiteScript.

The editable elements of the record are the currentrate, averagerate, historicalrate, and externalid. The remaining record elements can be read and searched.

Important: Rates are created and deleted when subsidiaries or secondary accounting books are created and deleted. Script users can edit the record only if it is not derived (Element field-> isderived) and only when the accounting period to which this rate belongs is not closed (Element field-> isperiodclosed). In addition, the script user cannot edit a consolidated exchange rate of 1 for an elimination subsidiary and its direct parent (Element field-> iseliminationsubsidiary), with the following exception. Those customers who already have a non–1 consolidated exchange rate (current, average, or historical) between and elimination subsidiary and its direct parent subsidiary can edit the consolidated exchange rates for the elimination subsidiary.

Supported Functions

This record is partially scriptable. It can be read, updated, and searched using SuiteScript. It cannot be created, deleted, or transformed.

Code Samples

The following code snippets show how to update a consolidated rate between two subsidiaries.

```
var PARENT_SUBSIDIARY = 1;
```

```
var CHILD_SUBSIDIARY = 3;
var PERIOD = 212;

// Search for the consolidated rate record from CHILD_SUBSIDIARY to PARENT_SUBSIDIARY for a spe
cific PERIOD
var filters = new Array();
var columns = new Array();
filters[0] = new nlobjSearchFilter('fromsubsidiary', null, 'is', CHILD_SUBSIDIARY);
filters[1] = new nlobjSearchFilter('tosubsidiary', null, 'is', PARENT_SUBSIDIARY);
filters[2] = new nlobjSearchFilter('period', null, 'is', PERIOD);
columns[0] = new nlobjSearchColumn('internalid');
var results = nlapiSearchRecord('consolidatedexchangerate', null, filters, columns);

// Update the consolidated rate's current, average, and historical rate
var rateRecord = nlapiLoadRecord('consolidatedexchangerate', results[0].getValue('internalid'),
 true);
rateRecord.setFieldValue('currentrate', 1.2);
rateRecord.setFieldValue('averagerate', 1.3);
rateRecord.setFieldValue('historicalrate', 1.4);
nlapiSubmitRecord(rateRecord);
```

Contact Category

Note: The content in this help topic pertains to all versions of SuiteScript. .

Contact category defines a list of values that are used by the contact record to set the type of contact. In the UI, this is a user defined list at Setup > Sales > CRM Lists > Contact Category.

For details about working with the contact category record in the user interface, see the help topic Setting Up Accounting Lists.

The internal ID for this record is `contactcategory`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The contact category record is scriptable in both client and server SuiteScript.

Supported Functions

The contact category record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Contact Role

Note: The content in this help topic pertains to all versions of SuiteScript.

Contact roles can be defined for each contact listed on various records including opportunity, customer, vendor, and partner.

The internal ID for this record is `contactrole`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The contact role record is scriptable in both client and server SuiteScript.

Supported Functions

The contact role record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Cost Category

Note: The content in this help topic pertains to all versions of SuiteScript.

Cost category records are used to classify different types of costs associated with your items. Using cost categories helps you to track costs and variances in the manufacturing process. These categories are available when the Standard Costing or Landed Cost feature is enabled

Material or service cost categories track standard costs for items. For details, see the help topic Creating Cost Categories.

Landed cost category values are used for items on bills and item receipts to categorize the different kinds of expenses incurred when making purchases. For details, see Landed Cost Categories.

The internal ID for this record is `costcategory`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The cost category record is scriptable in both client and server SuiteScript.

Supported Script Functions

The cost category record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Currency

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The internal ID for this record is `currency`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The currency record is scriptable in server SuiteScript only.

Supported Functions

When the Multiple Currencies feature is enabled, full server-side scripting is supported for this record type.

When the Multiple Currencies feature is not enabled, scripting does not support create, edit, delete, or search of currency records. Search-based functions such as nlapiLookupField are not supported, because search is not supported. Loading of a currency record to get field values is supported, as long as the currency ID is known, as shown in the following example:

```
var rec = nlapiLoadRecord('currency', 1);
var symbol=rec.getFieldValue('symbol');
var a = 1;
```

Usage Notes

As of 2016.1, the `currencyprecision` field is available for scripting, even when the ALLOWCURRENCYPRECISIONCHANGE preference is disabled and the field is read-only. In scripting, this field is always a number and never a list box, behavior which differs from this field's behavior in the UI.

Customer Category

Note: The content in this help topic pertains to all versions of SuiteScript.

Customer category defines a list of values that are used by the customer record to set the type of customer. In the UI, reference an existing customer category list by going to Setup > Accounting > Accounting Lists.

To create a new customer category list, set the Type dropdown list at the top of the page to Customer Category. Next, click the New button that appears at the top of the page.

The internal ID for this record is `customercategory`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The customer category record is scriptable in both client and server SuiteScript.

Supported Functions

The cost category record is partially scriptable — it can be created, updated, copied, and deleted using SuiteScript. Search is not available for the customer category record.

Customer Message

Note: The content in this help topic pertains to all versions of SuiteScript.

Standardized customer messages can be created to be available for selection on transaction records. A selected message is sent to the customer for the transaction. For example, a message of "Thank you for your order!" can be selected on an invoice record to be displayed on the invoice sent to the customer. For information about how to define customer messages, see the help topic Setting Up Accounting Lists.

The internal ID for this record is `customermessage`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The customer message record is scriptable in both client and server SuiteScript.

Supported Functions

The customer message record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Customer-Subsidiary Relationship

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

If you use NetSuite OneWorld and have enabled the Multi-Subsidiary Customer feature, you can share the customer and sub-customer records with multiple subsidiaries, and then select those subsidiaries on core transactions. An administrator can enable these features at Setup > Company > EnableFeatures, on the Company subtab. For help working with Multi-Subsidiary Customers in the user interface, see the help topic Assigning Subsidiaries to a Customer.

The internal ID for this record is `customersubsidiaryrelationship`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For customer-subsidiary and vendor-subsidiary relationships belonging to the same entity, both share the external ID of the entity. For example, Company ABC can be both a vendor and customer. If Company ABC has a vendor and customer relationship with the US subsidiary, both share the same external ID.

Supported Script Types

The customer-subsidiary relationship is scriptable in both client and server SuiteScript.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

The customer-subsidiary relationship can be read, created, updated, and deleted.

Code Sample

The following example shows how to create a customer-subsidiary relationship.

```
var csr = nlapiCreateRecord('customersubsidiaryrelationship');
csr.setFieldValue('entity', 42);
csr.setFieldValue('subsidiary', 1);
nlapiSubmitRecord(csr);
```

Department

Note: The content in this help topic pertains to all versions of SuiteScript.

Departments are categories that you can create to separate and track records such as financials, transactions, and employees. For example, you can create a department for each team of employees dedicated to a certain area of business, and then track income and expenses by each department over any time period.

For details about working with the department record in the user interface, see the help topic Creating Departments.

The internal ID for this record is `department`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The department record is scriptable in both client and server SuiteScript.

Supported Functions

The department record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Expense Category

Note: The content in this help topic pertains to all versions of SuiteScript.

Expense categories are used to group expenses. Popular categories include transportation, lodging, mileage, and entertainment. Each expense category is linked to an account. When an employee enters

an expense report, he or she selects a category for each expense, and the expense automatically posts to the associated expense account. Note that new expense categories cannot be created at the time an expense report is entered.

If you use NetSuite OneWorld, be aware that an expense category is available to only those subsidiaries assigned to the account linked with the expense category. If you want to enable intercompany expense transactions, it is recommended that you set up expense categories linked to expense accounts that are available to all subsidiaries, for use in these transactions. For intercompany expense transactions, users cannot save expense lines unless they contain expense categories available to both the employee subsidiary and customer subsidiary.

To enter expense categories, go to Setup > Accounting > Setup Tasks > Expense Categories > New (Administrator). For more information, see the help topic Creating an Expense Category.

The internal ID for this record is `expensecategory`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The expense category record is scriptable in server SuiteScript only.

Supported Functions

The expense category record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Fair Value Price

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Fair Value Price is a list of the records that define the fair value for items. Fair value price is used to allocate revenue in revenue arrangements

This record is part of advanced revenue management. To use advanced revenue management, the Accounting Periods feature and Advanced Revenue Management feature must be enabled. Before you begin working with advanced revenue management programmatically, see the help topic Setup for Advanced Revenue Management.

For help working with this record in the user interface, see the help topic Fair Value Setup.

The internal ID for this record is `fairvaluepricelist`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The fair value price record is scriptable in server SuiteScript only.

Supported Functions

The fair value price record is partially scriptable. It can be created, deleted, and searched using SuiteScript. You cannot copy the fair value price record.

Gift Certificate

Note: The content in this help topic pertains to all versions of SuiteScript.

You can create gift certificate items that allow customers to purchase store credit they can send to someone as a gift. The recipient uses the gift certificate code when placing an order through your Web store or entering a transaction with a sales representative. You can set a preference in NetSuite for how you want to generate the gift certificate codes: you can create them yourself, or use a random hash code automatically generated by the system.

Gift certificate codes are not active until the order used to purchase the gift certificate is billed.

For details about working with the gift certificate record in the user interface, see the help topic Gift Certificates.

The internal ID for this record is `giftcertificate`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The gift certificate record is scriptable in both client and server SuiteScript.

Supported Functions

The gift certificate record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Global Account Mapping

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

For accounts using Multi-Book Accounting, the global account mapping record enables you to configure secondary accounting books to post to accounts different from the primary book. These mappings are used by transactions where the user can manually select the account to which the transaction posts.

The internal ID for this record is `globalaccountmapping`.

Using this record requires that, as part of your Multi-Book configuration, you select the Chart of Accounts Mapping option at Setup > Enable Features, on the Accounting subtab, in addition to the other setup steps required for Multi-Book Accounting.

In the user interface, you access this record at Setup > Accounting > Global Account Mappings > New.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

This record is scriptable in server SuiteScript only.

All three user events are supported: before Load, before Submit, and afterSubmit.

Supported Functions

This record is fully scriptable, which means it can be created, updated, copied, deleted, and searched using SuiteScript.

Code Samples

The following code snippets show how to create global account mapping records and perform other basic tasks.

```
// Create Record
var rec = nlapiCreateRecord('globalaccountmapping', null);
rec.setFieldValue('effectivedate', '4/4/2004');
rec.setFieldValue('accountingbook', '2');
```

```
rec.setFieldValue('sourceaccount', '6');
rec.setFieldValue('subsidiary', '1');
rec.setFieldText('class', 'Class US');
rec.setFieldText('department', 'Department  US');
rec.setFieldText('location', 'Location US');
rec.setFieldValue('destinationaccount', '6');
var id = nlapiSubmitRecord(rec);
var x = 0;

// Update Record
var rec = nlapiLoadRecord('globalaccountmapping', '102');
rec.setFieldValue('effectivedate', '5/5/2005');
rec.setFieldText('sourceaccount', 'Advances Paid');
rec.setFieldText('destinationaccount', 'ABN Withholding');

var id = nlapiSubmitRecord(rec);


// Delete Record
var rec = nlapiDeleteRecord('globalaccountmapping', '102');


// Copy Record
var rec = nlapiCopyRecord('globalaccountmapping', '103');
rec.setFieldValue('effectivedate', '5/5/2005');
rec.setFieldText('sourceaccount', 'Advances Paid');
rec.setFieldText('destinationaccount', 'ABN Withholding');
var id = nlapiSubmitRecord(rec);
```

Government-Issued ID Tracking

Note: The content in this help topic pertains to all versions of SuiteScript.

The Government-Issued ID Tracking feature enables you to track a variety of government-issued identification information about your employees to ensure that you have accurate documentation for your employees' travel, driving qualifications, and tax compliance. For information about this feature, see the help topic Government-Issued ID Tracking

The following record types are related to the Government-Issued ID Tracking feature:

- Driver's License
- Passport
- Other Government-Issued ID

Driver's License

Note: The content in this help topic pertains to all versions of SuiteScript.

The driver's license record captures the driver's license number for a particular employee.

The internal ID for this record is `driverslicense`. For more details about this record and how to work with it in the user interface, see the help topic Government-Issued ID Tracking.

This record becomes available when either the Basic Government-Issued ID Tracking or Advanced Government-Issued ID Tracking features are enabled.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

This record is scriptable in both server and client SuiteScript. However, user event scripts are not supported.

Supported Functions

This record is fully scriptable, which means it can be created, edited, copied, deleted, and searched.

Code Sample

The following example shows how to create a new driver's license for an employee.

```
var driversLicenseRecord = nlapiCreateRecord('driverslicense');

driversLicenseRecord.setFieldValue('employee', '123');
driversLicenseRecord.setFieldValue('numbervalue', '123456789');
driversLicenseRecord.setFieldValue('nameondocument', 'John Doe');
driversLicenseRecord.setFieldValue('issuer', 'United States Government');
driversLicenseRecord.setFieldValue('class', 'M3');
driversLicenseRecord.setFieldValue('dateofissue', '10/12/2017');
driversLicenseRecord.setFieldValue('expirationdate', '10/12/2027');
driversLicenseRecord.setFieldValue('description', 'This is a Drivers License');

var driversLicenseInternalId = nlapiSubmitRecord(driversLicenseRecord);
```

Passport

Note: The content in this help topic pertains to all versions of SuiteScript.

The passport record captures the passport number for a particular employee.

The internal ID for this record is `passport`. For more details about this record and how to work with it in the user interface, see the help topic Government-Issued ID Tracking.

This record becomes available when either the Basic Government-Issued ID Tracking or Advanced Government-Issued ID Tracking features are enabled.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

This record is scriptable in both server and client SuiteScript. However, user events are not supported.

Supported Functions

This record is fully scriptable, which means it can be created, edited, copied, deleted, and searched.

Code Sample

The following example shows how to create a new passport number for an employee.

```
var passportRecord = nlapiCreateRecord('passport');  passportRecord.setFieldValue('employee', '
123');

passportRecord.setFieldValue('employee', '123');
passportRecord.setFieldValue('numbervalue', '123456789');
passportRecord.setFieldValue('issuer', 'United States Government');
passportRecord.setFieldText('nationality', 'American');
passportRecord.setFieldValue('nameondocument', 'John Doe');
passportRecord.setFieldValue('dateofissue', '10/12/2017');
passportRecord.setFieldValue('expirationdate', '10/12/2027');
passportRecord.setFieldValue('description', 'This is a Passport');

var passportInternalId = nlapiSubmitRecord(passportRecord);
```

Other Government-Issued ID

Note: The content in this help topic pertains to all versions of SuiteScript.

The other government-issued ID record captures ID types other than driver's license and passport.

The internal ID for this record is `othergovernmentissuedid`. For more details about this record and how to work with it in the user interface, see the help topic Government-Issued ID Tracking.

This record becomes available when the Advanced Government-Issued ID Tracking feature is enabled.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

This record is scriptable in both server and client SuiteScript. However, user events are not supported.

Supported Functions

This record is fully scriptable, which means it can be created, edited, copied, deleted, and searched.

Code Sample

The following example shows how to create a new government-issued ID type.

```
var otherGovIdRecord = nlapiCreateRecord('othergovernmentissuedid');

otherGovIdRecord.setFieldValue('employee', '123');
otherGovIdRecord.setFieldText('idtype', 'Social Insurance Number (SIN)');
otherGovIdRecord.setFieldValue('numbervalue', '123456789');
otherGovIdRecord.setFieldValue('nameondocument', 'John Doe');
otherGovIdRecord.setFieldValue('issuer', 'United States Government');
otherGovIdRecord.setFieldValue('expirationdate', '10/12/2027');
otherGovIdRecord.setFieldValue('description', 'This is a Other Government ID');

var otherGovIdInternalId = nlapiSubmitRecord(otherGovIdRecord);
```

Group

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

You use the group record to define groups of contacts, customers, employees, partners, or vendors.

The internal ID for this record is `entitygroup`.

In the UI, you access this record at Lists > Relationships > Group > New.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The group record is scriptable in server SuiteScript only. None of the user events are supported.

Supported Functions

This record is fully scriptable, which means it can be created, read, updated, deleted, and searched.

Usage Notes

Be aware of the following details when working with this record:

- The grouptype field is required and must be set either to “static” or “dynamic.”
- If the grouptype is dynamic, you can never manually add to the Members sublist (because the group members are determined by a saved search). This behavior is the same as in the user interface.
- If the grouptype is static, you can create lines in the Members sublist during the time that you are creating the record, but you cannot modify the sublist during updates. This behavior differs from the user interface, which does allow manual updates after the record is created.

  To add members to a group in SuiteScript after it has been created, you can use code like the following:

  ```
  nlapiAttachRecord('customer', <someCustomerId>, 'entitygroup', <someGroupId>);
  nlapiDetachRecord('customer', <someCustomerId>, 'entitygroup', <someGroupId>);
  ```

Code Sample

The following sample shows how to create a static group record.

```
var initValues = new Array();
initValues.grouptype = 'Employee';
initValues.dynamic = 'F';
var wc1 = nlapiCreateRecord('entitygroup', initValues);
wc1.setFieldValue('groupname', 'WC1');
wc1.setFieldValue('subsidiary', '1');
wc1.setFieldValue('ismanufacturingworkcenter', 'T');
wc1.setFieldValue('machineresources', '11');
wc1.setFieldValue('laborresources', '22');
var wc1Id = nlapiSubmitRecord(wc1);
```

Inbound Shipment

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Inbound shipments provide visibility of in-transit inventory and the status of a shipment. Items from multiple purchase orders can be assigned to an incoming shipment and bulk received and billed from within the record. Accounting support is also provided for the transfer of ownership of inbound inventory prior to receipt, through the ownership transfer record. For information about this record type, see Ownership Transfer.

The internal ID for this record is `inboundshipment`.

The inbound shipment record is available in NetSuite and NetSuite OneWorld accounts when the Inbound Shipment Management feature is enabled.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The inbound shipment record is supported in client and server SuiteScript.

Supported Operations

This record can be created, read, updated, deleted, and searched using SuiteScript. It cannot be copied or transformed.

Usage Notes

A script that adds a line item to an inbound shipment should include both the purchase order ID and the line number for the item in the purchase order items sublist.

Field Definitions

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Code Sample

The following script example illustrates a typical workflow for an inbound shipment. This example does the following:

- Creates the inbound shipment record.
- Adds line items from a purchase order.
- Enters information into the primary information fields.
- Updates the status field.
- Updates the expected shipping date.
- Creates a partial ownership transfer.
- Receives the inbound shipment.

```
var inboundShipment = nlapiCreateRecord('inboundshipment');
var purchaseOrder = nlapiLoadRecord('purchaseorder', 1718);
var itemLineCount = purchaseOrder.getLineItemCount('item')
for (var i = 1; i <= itemLineCount; i++) {
   inboundShipment.selectNewLineItem('items');
   inboundShipment.setCurrentLineItemValue('items', 'purchaseorder', purchaseOrder.getId());
```

```
   inboundShipment.setCurrentLineItemValue('items', 'shipmentitem', purchaseOrder.getLineItemVa
lue('item', 'lineuniquekey', i));
   inboundShipment.commitLineItem('items');
}
var inboundShipmentId = nlapiSubmitRecord(inboundShipment);

var inboundShipmentUpdate = nlapiLoadRecord('inboundshipment', inboundShipmentId);
inboundShipmentUpdate.setFieldValue('shipmentstatus', 'inTransit');
inboundShipmentUpdate.setFieldValue('externaldocumentnumber', 'EDN645');
inboundShipmentUpdate.setFieldValue('expectedshippingdate', '8/2/2017');
inboundShipmentUpdate.selectLineItem('items', 1);
inboundShipmentUpdate.setCurrentLineItemValue('items', 'receivinglocation', 6);
inboundShipmentUpdate.setCurrentLineItemValue('items', 'quantityexpected', 1);
inboundShipmentUpdate.setCurrentLineItemValue('items', 'expectedrate', 10.5);
inboundShipmentUpdate.commitLineItem('items');
nlapiSubmitRecord(inboundShipmentUpdate);

var takeOwnership = nlapiLoadRecord('bulkownershiptransfer', inboundShipmentId);
takeOwnership.selectLineItem('items', 2);
takeOwnership.setCurrentLineItemValue('items', 'process', 'F');
takeOwnership.commitLineItem('items');
nlapiSubmitRecord(takeOwnership);

var bulkReceive = nlapiLoadRecord('receiveinboundshipment', inboundShipmentId);
bulkReceive.selectLineItem('receiveitems', 2);
bulkReceive.setCurrentLineItemValue('receiveitems', 'quantitytobereceived', 1);
bulkReceive.commitLineItem('receiveitems');
nlapiSubmitRecord(bulkReceive);
```

Intercompany Allocation Schedule

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Intercompany allocation schedules transfer a balance from one source subsidiary account to multiple destination subsidiaries for costs that are shared between subsidiaries.

The internal ID for this record is `intercompallocationschedule`.

The intercompany allocation schedule record is available only in OneWorld accounts and when the Accounting Periods and Expense Allocation features are enabled. An administrator can enable these features at Setup > Company > EnableFeatures > Accounting subtab. For information about expense allocation, see the help topic Expense Allocation Overview. For details on working with the intercompany allocation schedule in the user interface, see the help topic Creating Intercompany Allocation Schedules.

If the Statistical Accounting feature is enabled, the intercompany allocation schedule record includes fields that enable you to base the weight for the allocation on the balance of a statistical account through statistical journals or as an absolute value. If the Dynamic Allocation feature is also enabled, the weight is dynamically calculated when the intercompany allocation journal is generated. An administrator can enable these features at Setup > Company > EnableFeatures > Accounting under Advanced Features. For information about statistical allocation schedules, see the help topic Working with Allocation Schedules Weighted by the Balance of a Statistical Account.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

This record is scriptable in both client and server SuiteScript.

If the Dynamic Allocation feature is enabled, the currency field is not scriptable. None of the fields from the History subtab on the intercompany allocation schedule record in the user interface are not scriptable.

Supported Functions

This record is partially scriptable. It can be created, read, updated, and deleted using Suitescript. It cannot be copied, updated, searched, or transformed.

Code Samples

The following code snippets show how to create a statistical intercompany allocation schedule that divides the balance of one account and transfers the amounts into two other subsidiaries.

```
var record = nlapiCreateRecord("intercompallocationschedule", false)

record.setFieldValue("name", "Allocation Schedule A");
record.setFieldValue("subsidiary", "1");

record.setLineItemValue("allocationsource", "account", 1, "24");

record.setLineItemValue("allocationdestination", "subsidiary", 1, "6");
record.setLineItemValue("allocationdestination", "weight", 1, "50.00");

record.setLineItemValue("allocationdestination", "subsidiary", 1, "32");
record.setLineItemValue("allocationdestination", "weight", 1, "50.00");

recordId = nlapiSubmitRecord(record);
```

Inventory Number

Note: The content in this help topic pertains to all versions of SuiteScript.

The term inventory number refers to a serial number or a lot number. An inventory number record uniquely identifies an item in physical inventory with a serial number, or uniquely identifies a group of items with a lot number. For details about these types of items, see Serial Numbered Items and Lot Numbered Items.

The internal ID for this record is `inventorynumber`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

Client SuiteScript is not supported for the inventory number record. It is scriptable in server SuiteScript only.

Supported Functions

The inventory number record is partially scriptable. It can be updated and searched using SuiteScript. Copy, Create, and Delete are not allowed for this record.

Item Account Mapping

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

For accounts using Multi-Book Accounting, the item account mapping record enables you to configure secondary accounting books to post to accounts different from the primary book. These mappings are used by transactions where the item determines the account to which the transaction posts.

Using this record requires that, as part of your Multi-Book configuration, you select the Chart of Accounts Mapping option at Setup > Enable Features, on the Accounting subtab, in addition to the other setup steps required for Multi-Book Accounting.

In the user interface, you access this record at Setup > Accounting > Item Account Mappings > New.

The internal ID for this record is `itemaccountmapping`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

This record is scriptable in server SuiteScript only.

All three user events are supported: before Load, before Submit, and afterSubmit.

Supported Functions

The item account mapping record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Code Samples

The following code snippets show how to create item account mapping records and perform other basic tasks.

```
// Get Record
var rec = nlapiLoadRecord('itemaccountmapping', '105');
var id = 0;


// Create Record
var rec = nlapiCreateRecord('itemaccountmapping', null);
rec.setFieldValue('effectivedate', '4/4/2004');
rec.setFieldValue('accountingbook', '2');
rec.setFieldValue('sourceaccount', '118');
rec.setFieldText('itemaccount', 'Asset');
rec.setFieldValue('subsidiary', '3');
rec.setFieldText('class', 'Class US');
rec.setFieldText('department', 'Department  US');
rec.setFieldText('location', 'Location US');
rec.setFieldValue('destinationaccount', '118');
var id = nlapiSubmitRecord(rec);
var x = 0;


// Update Record
var rec = nlapiLoadRecord('itemaccountmapping', '201');
rec.setFieldValue('effectivedate', '5/5/2005');
rec.setFieldValue('subsidiary', '1');
rec.setFieldText('itemaccount', 'Discount');
rec.setFieldText('sourceaccount', 'Advertising');
rec.setFieldText('destinationaccount', 'Advertising');
var id = nlapiSubmitRecord(rec);


// Copy Record
var rec = nlapiCopyRecord('itemaccountmapping', '201');
rec.setFieldValue('effectivedate', '6/6/2005');
rec.setFieldText('itemaccount', 'Asset');
rec.setFieldValue('sourceaccount', '118');
rec.setFieldValue('destinationaccount', '118');
var id = nlapiSubmitRecord(rec);


// Delete Record
var rec = nlapiDeleteRecord('itemaccountmapping', '201');
```

Item Location Configuration

The internal ID for the item location configuration record is `itemlocationconfiguration`.

The item location configuration record is available only when the following features are enabled:

- Multi-Location Inventory
- Advanced Inventory Management
- Advanced Item Location Configuration

The Advanced Item Location Configuration feature supports tracking of location attributes data for items in accounts where the Multi-Location Inventory and Advanced Inventory Management features are enabled. For information about these features, see the help topics Advanced Inventory Management and Multi-Location Inventory.

For details about using Advanced Item Location Configuration, read the following:

- Advanced Item Location Configuration
- Enabling Advanced Item Location Configuration
- Using Advanced Item Location Configuration
- Working with Item Location Attributes
- Duplicate Item Attributes from a Single Location
- Duplicate Item Attributes from Multiple Locations
- Edit Item Location Attributes

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The item location configuration record is scriptable in both client and server SuiteScript.

Supported Operations

The item location configuration record can be created, read, edited, deleted, and searched using SuiteScript. It cannot be copied or transformed.

Code Sample

The following code snippets illustrate how to make changes on an item location configuration record.

```
// OW account, subsidiary (id=1) "Parent Company", locations: (id=6) "US ONLY Location"
// Create Record (OW account, )
var rec = nlapiCreateRecord('itemlocationconfiguration', {'item':'18'});
rec.setFieldValue('subsidiary', '1');
rec.setFieldValue('location', '6');
rec.setFieldValue('itemid', '18');
rec.setFieldValue('name', 'ilc record1');
```

```
nlapiSubmitRecord(rec);

// Update Record
var rec = nlapiLoadRecord('itemlocationconfiguration', '10');
rec.setFieldValue('defaultreturncost', '4.3');
rec.setFieldValue ('poreceiptcost', '7.54');
rec.setFieldValue ('safetystocklevel', '12');
nlapiSubmitRecord(rec);

// Delete Record
var rec = nlapiDeleteRecord('itemlocationconfiguration', '10');
```

Item Revision

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The internal ID for this record is `itemrevision`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

Client SuiteScript is not supported for this record. It is scriptable in server SuiteScript only.

Supported Functions

The item revision record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Sample Code

The following example creates an item revision record:

```
function afterSubmit(type)
{

var itemrev = nlapiCreateRecord("itemrevision");

itemrev.setFieldValue("name", "revision name 222");
```

```
itemrev.setFieldValue("item", "109");
itemrev.setFieldValue("memo", "revision memo");
itemrev.setFieldValue("effectivedate", "3/4/2012");

var id = nlapiSubmitRecord(itemrev, true);

}
```

Kudos Feature Records

Note: The content in this help topic pertains to all versions of SuiteScript.

The Kudos feature enables your employees to give their colleagues social-based recognition for projects completed or a job well done. For more information about this feature, see the help topic Kudos.

The following record types are related to the Kudos feature:

- Kudos
- Organization Value

Kudos

Note: The content in this help topic pertains to all versions of SuiteScript.

A Kudos is an act of recognition from one employee to another.

The internal ID for this record is `kudos`. For more details about this record and how to work with it in the user interface, see see the help topic Kudos.

This record becomes available when the Kudos feature is enabled.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

This record is scriptable in server SuiteScript only.

Supported Functions

This record can be read, created, edited, copied, deleted, and searched using SuiteScript. Note that only the inactive field can be edited.

Organization Value

Note: The content in this help topic pertains to all versions of SuiteScript.

An organization value is a value that you associate with a Kudos when you give a Kudos to a colleague.

The internal ID for this record is `organizationvalue`. For more details about this record and how to work with it in the user interface, see the help topic Kudos.

This record becomes available when the Kudos feature is enabled.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

This record is scriptable in server SuiteScript only.

Supported Functions

This record can be read, created, edited, copied, deleted, and searched using SuiteScript.

Location

The Locations feature enables you to track information about employees and transactions for multiple offices or warehouses. For example, you can create locations for the corporate office and all of the sales offices, or for warehouses in several states. You can associate transactions and employees with each location, and then filter report data by location. For more information, see the help topic Locations Overview.

The internal ID for this record is `location`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

Client SuiteScript is not supported for this record. It is scriptable in server SuiteScript only.

Supported Functions

The location record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Manufacturing Cost Template

Note: The content in this help topic pertains to all versions of SuiteScript.

To work with this record, the Manufacturing Routing and Work Center feature must be enabled at Setup > Company > Enable Features, on the Items & Inventory tab.

In the UI, this record is accessed by going to Lists > Supply Chain > Manufacturing Cost Template.

The internal ID for this record is `manufacturingcosttemplate`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The manufacturing cost template record is scriptable in server SuiteScript only.

Supported Functions

The manufacturing cost template record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Manufacturing Routing

Note: The content in this help topic pertains to all versions of SuiteScript.

To work with this record, the Manufacturing Routing and Work Center feature must be enabled at Setup > Company > Enable Features, on the Items & Inventory tab.

In the UI, this record is accessed by going to Lists > Supply Chain > Manufacturing Routing.

The internal ID for this record is `manufacturingrouting`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The manufacturing routing record is scriptable in server SuiteScript only.

Supported Functions

The manufacturing routing record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Nexus

Note: The content in this help topic pertains to all versions of SuiteScript.

A nexus is a tax jurisdiction, usually defined at the country level. This type of entity is available when the Advanced Taxes feature is enabled, to allow users to manage and calculate taxes for different jurisdictions within the same NetSuite account. For details, see the help topic Advanced Taxes.

The internal ID for this record is `nexus`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The nexus record is scriptable in server SuiteScript only.

Supported Functions

The nexus record is partially scriptable. It can be created, updated, deleted, and searched using SuiteScript. It cannot be copied.

Note Type

Note: The content in this help topic pertains to all versions of SuiteScript.

This type defines a list of values that are used by the note record to set the type of note. In the user interface, this is a user defined list at Setup > Sales > Setup Tasks > CRM Lists > New > Note Type.

The internal ID for this record is `notetype`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The note type record is scriptable in both client and server SuiteScript.

Supported Functions

The note type record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Other Name Category

> **Note:** The content in this help topic pertains to all versions of SuiteScript.

Other name category values are used on other name records to categorize them. The list of other name records is a collection of records for people or companies who are not vendors, customers, or employees. You can use other name records to track other people or companies to whom you write checks or from whom you receive deposits.

For example, your company might donate money to a favorite charity, so you can create an other name record for the charity. You might also list your owners and partners here if they contribute or withdraw equity.

The internal ID for this record is `othernamecategory`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The other name category record is scriptable in both client and server SuiteScript.

Supported Functions

The other name category record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Ownership Transfer

You can use the ownership transfer record to transfer ownership of an inbound shipment before it has been physically received by a receiving location. This transfer makes it possible to take ownership of an inbound shipment as soon as it has been marked as in-transit. Ownership transfers can be applied to individual items as well as entire shipments. A posting transaction, Ownership Transaction, reflects the impact of the transfer on the general ledger, summarized as: DR External In Transit Inventory account; CR Inventory Received Not Billed account.

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The internal IDs for the two different ownership transfer records are:

- `bulkownershiptransfer` (the posting transaction bulk creation page)
- `ownershiptransfer` (the resulting posting transaction)

The ownership transfer record is available in NetSuite and NetSuite OneWorld accounts when the Inbound Shipment Management feature is enabled.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The ownership transfer record is supported in client and server SuiteScript.

Supported Operations

The `ownershiptransfer` record can be created, deleted and searched. This record type cannot be edited in SuiteScript.

Field Definitions

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Code Sample

The following code snippets illustrate how to create and delete an ownership transfer.

```
var takeOwnership = nlapiLoadRecord("bulkownershiptransfer", 267);  // set id of inbound shipme
```

```
nt you want transferred
takeOwnership.setLineItemValue("process","receivingLocation", 2, "F");  // second line won't be
 transferred
nlapiSubmitRecord(takeOwnership);
```

```
nlapiDeleteRecord("ownershiptransfer", 720);
```

Partner Category

Note: The content in this help topic pertains to all versions of SuiteScript.

To create new partner categories, go to Setup > Sales > Setup Tasks > CRM Lists (Administrator).

The note type record is scriptable in both client and server SuiteScript.

The internal ID for this record is `partnercategory`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The partner category record is scriptable in both client and server SuiteScript.

Supported Functions

The partner category record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Payment Method

Note: The content in this help topic pertains to all versions of SuiteScript.

You can create payment methods to add selections to the Payment Method field on transactions and in your Web store. For details, see the help topic Creating a Payment Method.

The internal ID for this record is `paymentmethod`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The payment method record is scriptable in both client and server SuiteScript.

Supported Functions

The payment method record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Payroll Item

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Payroll items store values for payroll transaction line items. Payroll items are used with payroll transactions generated by the Payroll feature, but these transactions are managed by NetSuite and are generally not used in integrations.

The internal ID for this record is `payrollitem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The payroll item record is scriptable in both client and server SuiteScript.

Supported Functions

The payroll item record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

The exposure of this record to SuiteScript is intended to enable global payroll solutions. You can use it along with the Paycheck Journal record to create custom payroll solutions and to support integrations with external payroll systems. For details, see the help topic Using the Paycheck Journal Feature.

Code Samples

The following samples create Deduction type payroll items.

```
function createPayrollItemdeductionMinimal()
   {
       var pi = nlapiCreateRecord('payrollitem');
      pi.setFieldValue('subsidiary', 1);
      pi.setFieldValue('itemtype', 16);
      pi.setFieldValue('liabilityaccount', 27);
      pi.setFieldValue('name', 'SSSitem-Deduction-Minimal');
      pi.setFieldValue('custrecord_payroll_item', 'Cust_field');
      pi.setFieldValue('externalid', 'testingexternalID');
      nlapiSubmitRecord(pi);
   }

   function createPayrollItemdeductionComplete()
   {
       var pi = nlapiCreateRecord('payrollitem');
      pi.setFieldValue('externalid', 'SSSitem-Deduction-Completed');
      pi.setFieldValue('subsidiary', 1);
      pi.setFieldValue('itemtype', 16);
      pi.setFieldValue('liabilityaccount', 150);
      pi.setFieldValue('name', 'SSSitem-Deduction');
      pi.setFieldValue('vendor', 1);
      pi.setFieldValue('employeepaid', true);
      pi.setFieldValue('custrecord_payroll_item', 'Cust_field');

      nlapiSubmitRecord(pi);
   }
```

The following samples create Earning:Addition type payroll items.

```
   function createPayrollItemAdditionMinimal()
   {
       var pi = nlapiCreateRecord('payrollitem');
      pi.setFieldValue('subsidiary', 1);
      pi.setFieldValue('itemtype', 6);
      pi.setFieldValue('expenseaccount', 114);
      pi.setFieldValue('name', 'SSSitem-addition-Minimal');
      pi.setFieldValue('custrecord_payroll_item', 'Cust_field');

      nlapiSubmitRecord(pi);
   }

   function createPayrollItemAdditionComplete()
   {
       var pi = nlapiCreateRecord('payrollitem');
      pi.setFieldValue('externalid', 'SSSitem-Addition-Completed');
      pi.setFieldValue('subsidiary', 1);
      pi.setFieldValue('itemtype', 6);
      pi.setFieldValue('expenseaccount', 114);
      pi.setFieldValue('name', 'SSSitem-Addition');
      pi.setFieldValue('custrecord_payroll_item', 'Cust_field');
```

```
        nlapiSubmitRecord(pi);
    }
```

The following code updates a payroll item.

```
    function updatePayrollItem()
    {
        var recordpi = nlapiLoadRecord('payrollitem', 110);

       recordpi.setFieldValue('liabilityaccount', 29);
       recordpi.setFieldValue('vendor', 6);
       recordpi.setFieldValue('custrecord_payroll_item', 'Updated');
       recordpi.setFieldValue('inactive', 'T');

       nlapiSubmitRecord(recordpi);
    }
```

The following code deletes a payroll item.

```
   function deletePayrollItem()
    {
   var savedSearchInternalId = 2
           var searchresults = nlapiSearchRecord('payrollitem', savedSearchInternalId, null, nu
ll);
    for ( var i = 0; searchresults != null && i < searchresults.length; i++ )
   {
      var searchresult = searchresults[ i ];
      if ( 0 < searchresults[i].getId())
      nlapiDeleteRecord(searchresults[i].getRecordType(), searchresults[i].getId());
   }

}
```

Price Book

Note: The content in this help topic pertains to all versions of SuiteScript.

The price book record, along with the price plan record, is used to build pricing information for subscriptions.

The price book record is available only when the SuiteBilling feature is enabled at Setup > Company > Enable Features (Administrator), on the Transactions subtab.

The internal ID for this record is `pricebook`.

See the SuiteScript Records Browser for all internal IDs associated with this record. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The price book record is scriptable in server and client SuiteScript.

Supported Functions

The price book record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Price Level

Note: The content in this help topic pertains to all versions of SuiteScript.

Price level defines a list of values that are used by the opportunity and item records to set the price level for a specific item. Items can be assigned different price levels — such as Employee Price or Corporate Discount Price.

When the Multiple Pricing feature is enabled, you can create up to 1000 price levels for items. Setting up multiple price levels allows greater flexibility to set different pricing for different customers. For information, see Creating Price Levels and Setting Up Items for Multiple Price Levels.

The internal ID for this record is `pricelevel`. Search is not available on this record type.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The price level record is scriptable in both client and server SuiteScript.

Supported Functions

The price level record is partially scriptable. It can be created, updated, copied, and deleted using SuiteScript. It cannot be searched.

Price Plan

Note: The content in this help topic pertains to all versions of SuiteScript.

The price plan record, along with the price book record, is used to build pricing information for subscriptions.

The price plan record is available only when the SuiteBilling feature is enabled at Setup > Company > Enable Features (Administrator), on the Transactions subtab.

The internal ID for this record is `priceplan`.

See the SuiteScript Records Browser for all internal IDs associated with this record. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The price plan record is scriptable in server and client SuiteScript.

Supported Functions

The price plan record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Pricing Group

Note: The content in this help topic pertains to all versions of SuiteScript.

Price groups allow you to assign customer-specific price levels for groups of items. For example, you could create a pricing group called Laptops and associate the pricing group with all of your laptop items. For details, see the help topic Pricing Groups.

The internal ID for this record is `pricinggroup`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The pricing group record is scriptable in both client and server SuiteScript.

Supported Functions

The pricing group record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Project Charge Rule

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The project charge rule record is available only when the Project Management and Charge-Based Billing features are enabled. In the user interface, you access this record on the Financial subtab of charge-based billing projects.

The internal IDs for this record is `chargerule`.

Project charge rules determine the billing rate, the timing of project charges, and the stage of a charge when it is generated.

For more information, see the help topic Understanding Charge Rules.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The project revenue rule record is supported in client and server SuiteScript.

Supported Operations

This record can be created, read, updated, deleted, copied, and searched using SuiteScript. It cannot be transformed.

Usage Notes

If charges generated from a project charge rule have been billed, the project charge rule record is read-only. The `chargeruletype` field is mandatory.

Field Definitions

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Code Samples

The following code snippets show how to create a time-based project charge rule.

```
var record= nlapiCreateRecord('ChargeRule', {'ChargeRuleType' : 'TIMEBASED'});
record.setFieldValue('Name', 'ChargeRuleName');
record.setFieldValue('Project', 99);
var id = nlapiSubmitRecord(n);
```

Project Expense Type

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Important: The project expense type record is part of the Job Costing and Project Budgeting feature. For information on the availability of this feature, please contact your account representative.

This record is available when the Job Costing and Project Budgeting feature is enabled at Setup > Company > Enable Features, on the Company tab. If you do not see this option in the UI, contact your account representative for assistance.

When the feature is enabled, you can access the project expense type record in the UI by choosing Setup > Accounting > Setup Tasks > Project Expense Types > New. For help manually creating a project expense type record, see the help topic Creating Project Expense Types.

The project expense type record lets you create expense classifications that you can apply to projects. The internal ID for this record is `projectexpensetype`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The project expense type record is not scriptable in client SuiteScript. It is scriptable in server SuiteScript only.

Supported Functions

The project expense type record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

The Language sublist is available only when the Multi-Language feature is enabled at Setup > Company > Enable Features, on the Company tab. The sublist's records can be updated, but you cannot add records to the sublist through SuiteScript.

Project Revenue Rule

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The project revenue rule record is available only when the Project Management, Charge-Based Billing, and Advanced Revenue Management features are enabled. In the user interface, you access this record on the Financial subtab of charge-based billing projects. There are three types of project revenue rules.

The internal IDs for these records are `laborbasedprojectrevenuerule`, `fixedamountprojectrevenuerule`, `pctcompleteprojectrevenuerule`.

Project revenue rules rely on charge rules to determine the amount of revenue to be distributed. The type of project revenue rule you create is determined by how you want the revenue to be recognized. A labor based project revenue rule recognizes the revenue created from time tracked on your project. A fixed amount project revenue rule can recognize revenue based on a date, task, or milestone. A percent complete project revenue rule recognizes revenue at a rate equivalent to the project's progress. Expense-based charge rules are not available to use with Project Revenue Recognition.

For more information, see the help topics Charge-Based Project Billing and Advanced Revenue Management.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The project revenue rule record is supported in client and server SuiteScript.

Supported Functions

The project revenue rule record is partially scriptable. It can be created, read, updated, deleted, and searched using SuiteScript. It cannot be copied or transformed.

Usage Notes

For labor based project revenue rules, the filters sublist is not exposed to SuiteScript. For fixed amount project revenue rules, the recurrence field is not exposed.

When you create a new project revenue rule you must set the Project (company) field to the project/job ID. Project revenue rules are not standalone records, and therefore must be associated with a specific project.

Code Samples

The following code snippet illustrates how to create a project revenue rule.

```
var record = nlapiCreateRecord('pctcompleteprojectrevenuerule');
record.setFieldValue('name', 'Percent Complete Rule');
record.setFieldValue('project', '25');
record.setFieldValue('serviceitem', '1');
var internalId = nlapiSubmitRecord(record);
```

Revenue Recognition Event

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The revenue recognition event record is available only when the Advanced Revenue Management feature is enabled. In the user interface, you access this record at Lists > Accounting > Revenue Recognition Events > New.

The internal ID for this record is `billingrevenueevent`.

Revenue recognition events trigger the creation of revenue recognition plans. Each revenue recognition event must have a revenue recognition event type. Revenue recognition event types are not scriptable. To create a revenue recognition event type, go to Lists > Accounting > Revenue Recognition Event Types > New. You must create at least one revenue recognition event type before you can create a revenue recognition event.

For a description of the revenue recognition event and revenue recognition event type records in the UI, see the help topic Creating a Custom Revenue Recognition Event.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The revenue recognition event record is supported in client and server SuiteScript.

The user events are not supported.

Supported Functions

The revenue recognition event record is partially scriptable. It can be created, read, updated, deleted, and searched using SuiteScript. It cannot be copied or transformed.

Usage Notes

The fields of this record that must be used in combination are fully described in the Revenue Recognition Event section of the help topic for the user interface. For details, see the help topic Creating a Custom Revenue Recognition Event.

Code Samples

The following code snippets show how to create revenue recognition events and perform other basic tasks.

```
// Create Record
var recRevRecEvent = nlapiCreateRecord('billingRevenueEvent');
recRevRecEvent.setFieldValue('transactionline', 12); //Transaction Line ID
recRevRecEvent.setFieldValue('eventtype', 2); //Event Type internal ID
recRevRecEvent.setFieldValue('eventpurpose', 'FORECAST'); //Event Purpose
recRevRecEvent.setFieldValue('eventdate', '12/31/2016'); //Event Date
recRevRecEvent.setFieldValue('quantity', 1); //Quantity
var recID = nlapiSubmitRecord(recRevRecEvent);

//Load Record
var rec = nlapiLoadRoad('billingRevenueEvent', recID);
```

Revenue Recognition Plan

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

This record is used to indicate the posting periods in which revenue should be recognized and the amount to be recognized in each period. Revenue plans are derived from revenue recognition rules. Each revenue element has a forecast plan and one or more actual plans. The actual revenue plans control the posting of revenue.

This record is part of advanced revenue management. To use advanced revenue management, the Accounting Periods feature and Advanced Revenue Management feature must be enabled. Before you begin working with advanced revenue management programmatically, see the help topic Setup for Advanced Revenue Management.

The internal ID for this record is `revrecplan`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The revenue recognition plan record is supported in server SuiteScript only.

The user events are not supported.

Supported Functions

The revenue recognition plan record is partially scriptable. It can be read, updated, and searched using SuiteScript. It cannot be created, copied, deleted, or transformed.

Planned Revenue Sublist

The internal id for this sublist is `plannedrevenue`. The sublist type is inline editor.

The following table lists the scriptable sublist fields associated with the Planned Revenue sublist.

Field	Type	Internal ID
Amount	Decimal Number	`amount`
Planned Period	List/Record	`plannedperiod`

For help working with this record in the user interface, see the help topic Revenue Recognition Plans.

Revenue Recognition Schedule

Note: The content in this help topic pertains to all versions of SuiteScript.

A revenue recognition schedule indicates the posting periods in which revenue should be recognized, and the amount to be recognized in each period, for an item sale. A revenue recognition schedule is generated for any sales transaction item that has an associated revenue recognition template. The point at which a revenue recognition schedule is generated for an item sale depends upon the type of sales transaction and enabled features and preferences set in your account. The schedule could be generated when a transaction is first saved, when it is approved, or when it is billed. Revenue recognition schedules provide a basis for the generation of journal entries that record the impact of item sales. This record is available when the Revenue Recognition feature is enabled. For more details, see the help topic Working with Revenue Recognition Schedules.

Revenue recognition schedules are system-generated.

The internal ID for this record is `revrecschedule`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

Client SuiteScript is not supported for the revenue recognition schedule record. It is scriptable in server SuiteScript only. Also, user event scripts are not supported.

Supported Functions

The revenue recognition schedule record is partially scriptable. It can be read, updated, and searched using SuiteScript. It cannot be created, copied, deleted, or transformed.

Revenue Recognition Template

Note: The content in this help topic pertains to all versions of SuiteScript.

A revenue recognition template indicates how revenue from associated items should be posted. For each template: you can select from a choice of standard terms or define your own custom terms, set the time period over which recognition occurs, define an offset to delay the start of recognition, and set up an initial amount to be recognized. This record is available when the Revenue Recognition feature is enabled. For more information, see the help topics Revenue Recognition Template and Creating Revenue Recognition Templates.

The internal ID for this record is `revrectemplate`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

Client SuiteScript is not supported for this record. It is scriptable in server SuiteScript only. Also, user event scripts are not supported.

Supported Functions

The revenue recognition template record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Role

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

A role is a defined access configuration. To set up and manage user access to your NetSuite account, you need to set up roles that can be assigned to users. Roles include sets of permissions for viewing and/or editing data. Roles and their permissions determine the pages that users can see in the NetSuite interface and the tasks that they can complete. Each role is associated with a center, a user interface designed for a particular business area. For details, see the help topic NetSuite Roles Overview.

The internal ID for this record is `role`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

Client SuiteScript is not supported for the role record. It is scriptable in server SuiteScript only.

Supported Functions

This record is not fully scriptable. Only search is permitted. The Role record is supported in all search APIs, excluding the duplicate search APIs.

Usage Notes

To work with this record, a user must have Manage Roles permissions, at Setup > Users/Roles > Manage Roles > Search.

When passing role into the supported SuiteScript 1.0 search functions, the internal ID is case-sensitive and must be capitalized. This is unique to the search APIs.

```
var roles = nlapiSearchRecord('Role', SavedSearchToBeLoaded, filters, columns);
```

In SuiteScript 2.0, role is called as `Role=search.Type.ROLE`.

Sales Role

> **Note:** The content in this help topic pertains to all versions of SuiteScript.

Sales roles can be defined for each member of a sales team. To use this record the Team Selling feature must be enabled. To enable team selling, go to Setup > Company > Enable Features > CRM. Click the Team Selling checkbox, and then click **Save**.

The internal ID for this record is `salesrole`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The sales role record is supported in client and server SuiteScript.

Supported Functions

The sales role record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Sales Tax Item

Note: The content in this help topic pertains to all versions of SuiteScript.

The internal ID for this record is `salestaxitem`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

Client SuiteScript is not supported for the sales tax item record. It is scriptable in server SuiteScript only.

Supported Functions

The sales tax item record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Subsidiary

Note: The content in this help topic pertains to all versions of SuiteScript.

A NetSuite OneWorld account enables you to manage data for a hierarchical structure of separate legal entities, or subsidiaries. This structure is organized as a tree that rolls up to a root, or top-level parent subsidiary. The root subsidiary is the highest-level subsidiary in your account, and all other subsidiaries are below it in the hierarchy. If an account that is upgraded to OneWorld has preexisting data, this data is used for the root subsidiary.

Each subsidiary represents a separate company within your global organization. Subsidiaries can be international or domestic. When you create a subsidiary record in NetSuite, the country you define for its address determines the NetSuite edition and tax nexus associated with that subsidiary.

For more details about the subsidiary record, see the help topics Subsidiaries Overview and Setting Up Subsidiaries.

The internal ID for this record is `subsidiary`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The subsidiary record is supported in client and server SuiteScript.

Supported Functions

The subsidiary record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

The Company Information page and the root Subsidiary record share information. In OneWorld accounts, if you update shared fields on the Company Information page, the corresponding fields on the root Subsidiary record are also updated. An update to the Company Information page also triggers all user event scripts deployed on the Subsidiary record. If your script updates a shared field on the Subsidiary record, the corresponding field on the Company Information page is also updated.

The Subsidiary record is hidden in accounts that are not OneWorld, but the record still exists. You cannot deploy a stand-alone user event script on the Subsidiary record in accounts that are not OneWorld. You can, however, bundle your script and distribute it to these accounts. In this scenario, updates to the Company Information page trigger the execution of the bundled script.

You can create up to 124 subsidiary records in addition to the root subsidiary, for a total of 125.

On some records, the Subsidiary field is a multi-select field type. When executing a search containing multi-select fields, SuiteScript treats commas as a delimiter of values. It is therefore recommended that you avoid using commas in subsidiary record names. If you do not, subsidiary record names containing a comma can skew your search results.

System Note

> **Note:** The content in this topic pertains to all versions of SuiteScript.

System notes track changes made to a record and to general configuration settings that have a financial impact. A system note for a change on a record captures the date when the change was made, who made the change, the role of the user who made the change, the type of change, and the old

and new value in the record. System notes are logged on custom records as well as standard records. Anyone with view access to a record can view the system notes for that record. When searching system notes, you can search for certain records, use advanced search filters, create a saved search, or export the information for additional analysis. For more information about system notes, see the help topics System Notes Overview and Searching System Notes.

System notes are only exposed as a search.

In the UI, users who are not Administrators can only see system notes for which they are the authors. The same restriction applies to scripting.

The internal ID for this record is systemnote.

In SuiteScript 2.0, when using the search.Type enum, use the value SYSTEM_NOTE

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The system note record is scriptable in both client and server-side scripts.

Supported Functions

The system note record is partially scriptable. Only search scripting is permitted.

Code Samples

The following examples show how you can script with system note records.

The following SuiteScript 2.0 code snippet creates a saved search:

```
require(['N/search'],
    function(search) {
        function createSearch() {
            var systemNotesSearch = search.create({
                type: search.Type.SYSTEM_NOTE,
                title: 'Changes made in Employee Center',
                id: 'customsearch_emp_center_changes',
                columns: ['date', 'recordtype', 'record', 'field', 'oldvalue', 'newvalue'],
                filters: [['role', 'is', '15']]
            });
            systemNotesSearch .save();
        }
        createSearch();
    });
```

The following SuiteScript 1.0 code snippet does the same search as the preceding example inline:

```
var result = nlapiSearchRecord('SystemNote', null,
[ new nlobjSearchFilter('ROLE', null, 'is', '15')],
[ new nlobjSearchColumn('DATE'), new nlobjSearchColumn('RECORDTYPE'), new nlobjSearchColumn('RE
CORD'),
new nlobjSearchColumn('FIELD'), new nlobjSearchColumn('OLDVALUE'), new nlobjSearchColumn('NEWVA
LUE')]);
```

Tax Control Account

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

A tax control account is an account to which the amounts computed for indirect taxes, such as sales tax and VAT, are posted.

In the UI, you can create a tax control account, and view existing ones, at Setup > Accounting > Tax Control Accounts. Note also that a tax control account is essentially an account record, with a few differences. Because they are accounts, tax control accounts also show up in the full list of accounts at Lists > Accounting > Accounts.

The internal ID for this record is `taxacct`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The tax control account record is scriptable in both client and server SuiteScript. However, user event scripts are *not* supported.

Supported Functions

The tax control account record is partially scriptable. Only creating, copying, and some updates are permitted.

Updating

Updating is permitted, but only of certain fields: Description, External ID, IsInactive, and Name. However, you may be able to change other fields when interacting with the tax control account as an account. For details, see Account.

Alternatives to Deleting

nlapiDeleteRecord is not supported, because a tax control account cannot be deleted (either through the UI or through SuiteScript). If the Advanced Taxes feature is enabled, you can effectively remove the tax control account by deleting the nexus that the account is associated with. If the Advanced Taxes feature is *not* enabled, you cannot remove the account.

An alternative to deleting the account is to make it account inactive, by setting the IsInactive field to true.

For more details on accounts that cannot be deleted in NetSuite, refer to the help topic Deleting Accounts and Making Accounts Inactive.

Field Definitions

This section describes some of the key fields on the tax control account.

Country

The value for country is derived from the nexus value and is read-only. If your account does not have the Advanced Taxes feature enabled, the value for country will always be the same for all your tax control accounts.

Description

The description of the record can include only up to 50 characters. Otherwise, the operation fails with an error reading “The field description contained more than the maximum number (50) of characters allowed.”

Description is one of the few fields that can be modified during an update.

External ID

ExternalID is one of the few fields that can be modified during an update.

IsInactive

Because a tax control account cannot be deleted, you might want to make it inactive.

IsInactive is one of the few fields that can be modified during an update.

Name

The name of the record must be unique. If you try to add a tax control account using a non-unique value for the name field, the system returns an error reading “This record already exists,” even if you included a unique external ID.

Name is one of the few fields that can be modified during an update.

Nexus

You can set a value for nexus only if the Advanced Taxes feaure is enabled. If your NetSuite account does not use the Advanced Taxes feature, the value for nexus is set automatically and will always be the same for all your tax control accounts.

When Advanced Taxes is enabled, initializiation of a nexus value is required. (For an example, refer to Code Samples.) However, after the account is created, the value for nexus cannot be changed.

Note that Advanced Taxes is enabled in all OneWorld accounts and cannot be turned off.

Tax Account Type

All tax control accounts have a tax account type, but you only actively choose a tax account type for accounts in certain countries. For example:

- If you are creating a tax code for the United Kingdom, valid choices include “sales” and “purchase.”
- When creating a tax code for the United States, only one option exists (“sales”), so you do not set a value for this field.

In countries where both “sales” and “purchase” are valid choices, the tax account type field is required. After an account has been created, the tax account type cannot be changed.

In countries where only one choice is allowed, tax account type is automatically set, and you cannot change it through SuiteScript (or through the UI).

Note that if your account uses Advanced Taxes and has nexuses in many countries, you may need to set this field for some tax control accounts and not others.

For More Information

See the SuiteScript Records Browser for all internal IDs associated with this record.

Usage Notes

This section includes additional details on interacting with the tax control account record.

Finding the Internal ID

If you need the internal ID for an existing tax control account, note that you cannot find it through Setup > Accounting > Tax Control Accounts. However, you can find it at Lists > Accounting > Accounts. Make sure you have configured your NetSuite preference to “Show Internal IDs.” (You can find this choice at Home > Set Preferences.)

When sorting accounts at Lists > Accounting > Accounts, be aware that:

- Tax control accounts of type “sale” show up as Other Current Liability.
- Tax control accounts of type “purchase” show up as Other Current Asset.

Be aware that you can also add and update values for the external ID field.

Relationship to the Account Record

You can interact with the tax control account in many of the same ways you can interact with any account. However, some exceptions exist. For example, a tax control account cannot be deleted, and there are limits to what you can update, as described in Updating. However, when you interact with the record as an account, you may be able to update additional fields.

For details on interacting with the account record, see Account.

Code Samples

The following examples show how you can script with tax control account records.

Get

The following sample shows how to retrieve a tax account record with the internal ID of “186”:

```
var taxAcct = nlapiLoadRecord('taxacct', '186');
;
```

Add

The following example shows how to create a tax control account when the Advanced Taxes feature is enabled. In this scenario, a value for nexus is required:

```
var taxAcct = nlapiCreateRecord('taxacct', {'nexus', 'internal ID of nexus'});
taxAcct.setFieldValue('name', 'Name of account');
taxAcct.setFieldValue('description', 'Description of account');
taxAcct.setFieldValue('externalid', 'Your external ID value');
taxAcct.setFieldValue('taxaccttype', 'Type');
nlapiSubmitRecord(taxAcct);
```

Update

This example shows how to update the name, description, and external ID of a tax control account, and to make it active. Note that these fields are the only ones that can be updated.

```
var taxAcct = nlapiLoadRecord('taxacct', 'internal ID of tax control account');
taxAcct.setFieldValue('name', 'Updated name');
taxAcct.setFieldValue('description', 'Updated description');
taxAcct.setFieldValue('isinactive', 'F');
taxAcct.setFieldValue('externalid', 'Updated external ID of tax control account');
nlapiSubmitRecord(taxAcct);
;
```

Tax Group

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

You can use the tax group record to combine several tax codes, even if the taxes are paid to different jurisdictions. For example, a tax group in the US might include a state tax, a city tax, and a transit tax. The advantage of using a tax group is that, when you create a sales invoice, you can apply one tax group to the transaction, instead of several separate tax codes.

In the UI, you navigate to this record by choosing Setup > Accounting > Tax Groups.

The internal ID for the tax group record is `taxgroup`.

See the SuiteScript Records Browser for the internal IDs of fields, search filters, and search columns associated with this record.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The tax group record is scriptable in both client SuiteScript and Server SuiteScript.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

The tax group record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser.

Code Samples

The following sample shows how to create a US tax group.

```
var initValues = new Array();
initValues.nexuscountry = 'US';
var tg = nlapiCreateRecord('taxgroup', initValues);
tg.setFieldValue('itemid', 'Test US Tax Group');
tg.setFieldValue('description', 'Tax group description');
tg.selectNewLineItem('taxitem');
tg.setCurrentLineItemValue('taxitem', 'taxname', '-1425'); // -1425 is ID of TX_BUFFALO Tax Cod
e
tg.commitLineItem('taxitem');
recId = nlapiSubmitRecord(tg);
;
```

The following code sample shows how to create a tax group for Canada.

```
var newgroup = nlapiCreateRecord('taxgroup', {nexuscountry: 'CA'});
newgroup.setFieldValue('itemid', 'CA-T5');
newgroup.setFieldValue('piggyback', 'T');
newgroup.setFieldValue('taxitem1', 21); //Canadian tax code
newgroup.setFieldValue('taxitem2', 24); //Canadian tax code
newgroup.setFieldValue('subsidiary', 2); //Canadian subsidiary
newgroup.setFieldValue('state', 'AB'); //Canadian subsidiary
newgroup.setFieldValue('description', 'New Canada Tax Group');

nlapiSubmitRecord(newgroup);
;
```

Tax Period

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

A tax period defines a period of time over which your company tracks tax. A tax period can be a month, a quarter, or a year, depending on the frequency of your tax submissions or what your business has agreed with the local tax agency. You can run monthly, quarterly, or annual VAT/GST reports in NetSuite. For details, see the help topic Tax Periods Overview.

The internal ID for this record is `taxperiod`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The tax period record is scriptable in server SuiteScript only.

Supported Functions

The tax period record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Tax Type

Note: The content in this help topic pertains to all versions of SuiteScript.

A tax type determines where the tax paid or collected is tracked on the balance sheet. The balance sheet account to which NetSuite posts the collection or payment of tax is called the tax control account.

In NetSuite, the tax types are either already set up by default, or set up for you by Professional Services. When you add a subsidiary in a new country, the tax types for that country become available in the system automatically. However, an administrator can create new tax types if necessary. For details, see the help topic Tax Types Overview.

The internal ID for this record is `taxtype`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords

- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The tax type record is scriptable in server SuiteScript only.

Supported Functions

The tax type record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Term

Note: The content in this help topic pertains to all versions of SuiteScript.

Terms are used to specify when payment is due on your customers' invoices. Define the specific requirements of a term of payment by creating a term record. You can create different payment terms for different customers. In the UI, this is a user defined list at Setup > Accounting > Setup Tasks > Accounting Lists > Term > New. For more information about payment terms, see the help topic Creating Terms of Payment.

The internal ID for this record is `term`. Search is not available on this record type.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The term record is scriptable in both client and server SuiteScript

Supported Functions

The term record is partially scriptable. It can be created, updated, copied, and deleted using SuiteScript. It cannot be searched.

Unit of Measure

Note: The content in this help topic pertains to all versions of SuiteScript.

The Multiple Units of Measure feature enables you to define various units used to stock, purchase, and sell inventory items, and track non-monetary accounts. For details, see the help topics Multiple Units of Measure and Setting Up Units of Measure.

The internal ID for this record is `unitstype`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The unit of measure record is scriptable in server SuiteScript only.

Supported Functions

The unit of measure record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Vendor Category

> **Note:** The content in this help topic pertains to all versions of SuiteScript.

The internal ID for this record is `vendorcategory`. Search is not available on this record type.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The vendor category record is scriptable in client and server SuiteScript.

Supported Functions

The vendor category record is fully scriptable, which means that it can be created, updated, copied, deleted, and searched using SuiteScript.

Vendor-Subsidiary Relationship

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

If you use NetSuite OneWorld, you can share the vendor record with multiple subsidiaries, and then select those subsidiaries on core transactions. For help working with Multi-Subsidiary Vendors in the user interface, see the help topic Assigning Subsidiaries to a Vendor.

The internal ID for this record is `vendorsubsidiaryrelationship`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For customer-subsidiary and vendor-subsidiary relationships belonging to the same entity, both share the external ID of the entity. For example, Company ABC can be both a vendor and customer. If Company ABC has a vendor and customer relationship with the US subsidiary, both share the same external ID.

Supported Script Types

The vendor-subsidiary relationship is scriptable in both client and server SuiteScript.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

The vendor-subsidiary relationship can be read, created, updated, and deleted.

Code Sample

The following example shows how to create a vendor-subsidiary relationship.

```
var vsr = nlapiCreateRecord('vendorsubsidiaryrelationship');
vsr.setFieldValue('entity', 42);
vsr.setFieldValue('subsidiary', 1);
nlapiSubmitRecord(vsr);
```

Workplace

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

To use NetSuite Payroll, you must create workplace records for each location where you have a tax ID and maintain paid employees. NetSuite uses workplace records to define the taxes the employer must withhold and pay. Each employee you include in a payroll must have a designated workplace. For details, see the help topic Entering Workplace Records for Payroll.

This record is available only when the Payroll feature is enabled at Setup > Company > Enable Features on the Employees subtab.

In the user interface, you can access this record at Lists > Employees > Workplaces. For help working with this record in the user interface, see the help topic Entering Workplace Records for Payroll.

The internal ID for this record is `workplace`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The workplace record is scriptable in both client and server SuiteScript.

Supported Functions

The workplace record is fully scriptable, which means that the record can be created, updated, copied, deleted, and searched using SuiteScript.

Customization

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The following customization records are scriptable in SuiteScript:

- Custom List
- Custom Record
- Custom Segment Fields and Values
- Custom Transaction
- Scheduled Script Instance
- Script
- Script Deployment

Custom List

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

See the following for information about scripting with custom lists:

- Custom List IDs

- Search Filters
- Search Columns

Custom List IDs

Each custom list has a unique ID. This value shown in the ID field on the custom list record. When creating or interacting with an instance of a custom record type, you must use this ID.

Every custom list ID is prefaced by `customlist`. If the user who created the custom record type permitted the value to be entirely system generated, the ID ends with a numeric value (for example, `customlist100`). If the user who created the list customized the ID, it may be more descriptive. To view a list of all these IDs, navigate to Customization > Lists, Records, & Fields > Lists. These values are shown in the ID column.

Search Filters

Field Internal ID	Field UI Label	Field Type
internalid	Internal ID	select
internalidnumber	Internal ID (Number)	integer
isinactive	Inactive	checkbox
name	Name	text

Search Columns

Field Internal ID	Field UI Label	Field Type
internalid	Internal ID	select
isinactive	Inactive	checkbox
name	Name	text

Custom Record

The NetSuite UI enables you to create custom record types and work with instances of those types. The process of creating a custom record type is described in Creating Custom Record Types.

You cannot use SuiteScript to create a custom record type or make changes to an existing type. However, you can use SuiteScript to interact with instances of an existing type.

For example, suppose your system had a custom record type called Feature. Each instance of that record type would be a feature record. You could use SuiteScript to update an existing feature record, delete a feature record, or create a new feature record. However, you could not use SuiteScript to alter the Feature record type.

For a list of fieldscommonly available on instances of all custom records, see the custom record reference page in the SuiteScript Records Browser. These fields include owner, which identifies the user who created the record instance, and isinactive, which indicates whether the record is active.

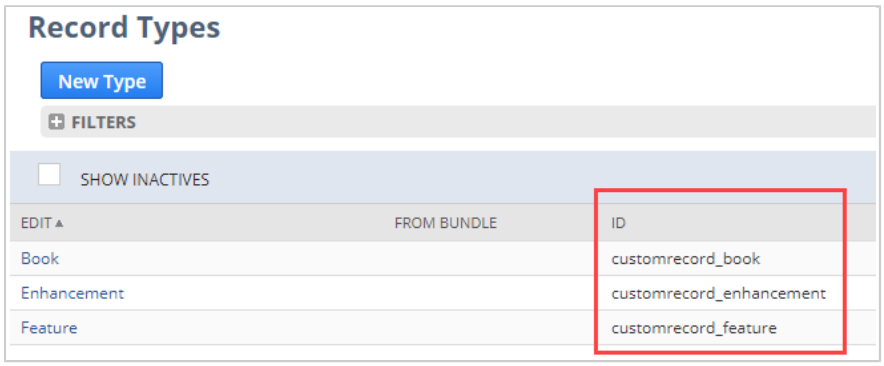
To interact with fields that were created for a specific record custom type, use the IDs for those fields. These IDs have a prefix of `custrecord`. For help finding these IDs, see How do I find a field's internal ID?

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

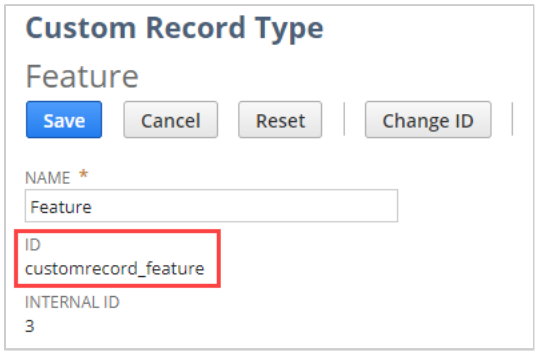
Custom Record IDs

Each custom record type has a unique ID. This value is shown in the ID field on the record that represents the custom record type. When creating or interacting with an instance of a custom record type, you must use this ID.

Every custom record ID is prefaced by `customrecord`. If the value was entirely system generated, the ID ends with a number (for example, `customrecord100`). If the ID was customized when the record type was created, the ID may be more descriptive. To view a list of all IDs, navigate to Customization > Lists, Records, & Fields > Record Types. The values are shown in the ID column.



You can also see the ID on the record that represents the custom record type.



If you are using SuiteScript 2.0, you use this value to set the Record.type or CurrentRecord.type field. Note that this guidance differs from the way you set this field if you are working with a standard NetSuite record type. When working with a standard record type, you set this field by using either the record.Type or CurrentRecord.type enum.

See also SuiteScript 2.0 Sample.

Supported Script Types

Custom record instances are supported in both client and server SuiteScript.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

Custom records are fully scriptable. They can be created, updated, copied, searched, and deleted using SuiteScript.

Script Samples

The following examples and snippets show how to work with instances of a custom record type with the script ID `customrecord_feature`. These examples assume that this custom record type has fields with the field IDs `custrecord_priority` and `custrecord_risklevel`.

SuiteScript 2.0 Sample

In SuiteScript 2.0, you use the N/record Module and N/currentRecord Module to interact with custom records, as you would with a standard record type. Note that you use the custom record type's string ID when setting the Record.type field, as shown in the following example. For help finding this ID, see Custom Record IDs.

```
/**
 * @NApiVersion 2.x
 * @NScriptType UserEventScript
 */

define([ 'N/record' ],

    function(record) {

        function afterSubmit (context) {


            // Use a string internal ID to identify the custom record type.

            var rec = record.create({
                type: 'customrecord_feature',
                isDynamic : true
            });

            rec.setValue({
                fieldId: 'name',
                value: 'entry form redesign'
            });

            rec.setValue({
                fieldId: 'custrecord_priority',
                value: 1
            });

            rec.setValue({
                fieldId: 'custrecord_risklevel',
                value: 3
            });


        try {
```

```
                var callId = rec.save();
                log.debug('Call record created successfully', 'Id: ' + callId);
            } catch (e) {
                log.debug('sigh');
                log.error(e.name);
            }
        }

        return {
            afterSubmit: afterSubmit
        };

});
```

SuiteScript 1.0 Sample

In SuiteScript 1.0, you use standard record APIs such as nlapiCreateRecord(type, initializeValues) to interact with a custom record.

```
function createFeatureRecord() {

    var obj = nlapiCreateRecord('customrecord_feature', {
        recordmode : 'dynamic'
    });

    obj.setFieldValue('name', 'entry form redesign');
    obj.setFieldValue('customrecord_priority', '1');
    obj.setFieldValue('customrecord_risklevel', '3');

    var id = nlapiSubmitRecord(obj);
}
```

The following example shows how to search among instances of a specific custom record type. In this example, the script searches for the instance of the `customrecord_feature` type that has the name `Entry Form Redesign`. The script writes the feature's priority and risk level to the execution log on the script deployment record.

```
function searchFeatureRecord() {

    var items = [];

    var filters = new Array();
    filters[0] = new nlobjSearchFilter("name", null, "is", "Entry Form Redesign", null);

    var columns = new Array();
    columns[0] = new nlobjSearchColumn("custrecord_priority");
    columns[1] = new nlobjSearchColumn("custrecord_riskLevel");

    var results = nlapiSearchRecord("customrecord_feature", null, filters, columns);

    try {
       var redesignPriority = results[0].getValue(columns[0]);
       var redesignRisk = results[0].getValue(columns[1]);

       nlapiLogExecution('DEBUG', 'redesign priority', redesignPriority);
```

```
        nlapiLogExecution('DEBUG', 'redesign risk', redesignRisk);
    }

     catch (e) {
        nlapiLogExecution('ERROR', ''+arguments.callee.name +' failed. ',e);
    }
}
```

Custom Segment Fields and Values

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Custom segments are custom classification fields similar to class, department, and location. When the Custom Segments feature is enabled, an administrator working in the UI can create custom segments, define possible values for each segment, and apply the segments to specific record types. A custom segment appears as a dropdown list or multiselect field on instances of record types where the segment was applied. Users can then classify records by selecting the appropriate value for each segment.

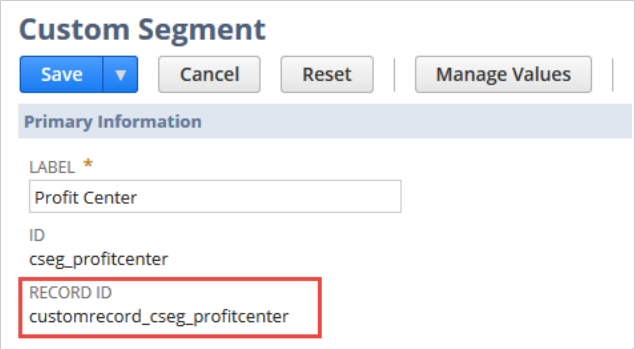
You cannot use SuiteScript to create a custom segment. However, you can use SuiteScript to create values for existing custom segments. You can also use SuiteScript to set values for segments that appear as fields on record instances. For details, see the following sections:

- Using SuiteScript to Create Values for Existing Custom Segments
- Using SuiteScript to Set Values for Custom Segment Fields

Using SuiteScript to Create Values for Existing Custom Segments

Your NetSuite account contains a custom record type for every custom segment that has been defined. To add a value to the segment, you add an instance of this record type.

You can find the internal ID of the appropriate custom record type on the custom segment definition, under the Record ID label.



The following SuiteScript 1.0 snippet shows how to create a value for a segment called Profit Center.

```
...

    // Create new segment value
```

```
    var newSegmentValue = nlapiCreateRecord('customrecord_cseg_profitcenter');

    // Set a name value on segment value record
    newSegmentValue.setFieldValue('name', stItemIdValue);

    // If appropriate, you may want to set a parent value for the new value.
    // For example, you might have a static value that you use for all
    // segment values created through a script.
    newSegmentValue.setFieldText('parent', 'Created from automated script');

...
```

Using SuiteScript to Set Values for Custom Segment Fields

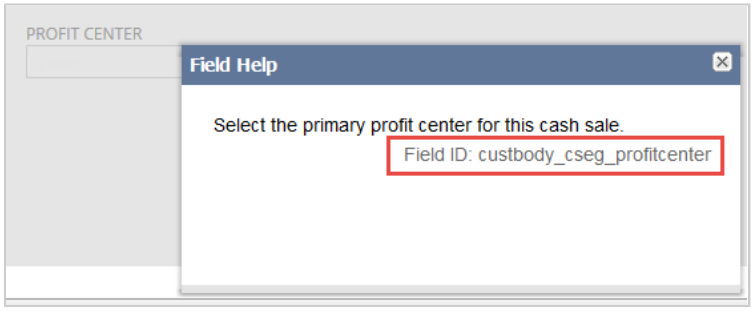
When a segment is applied to a record type, it appears as a field on instances of that type. You can interact with this field as you would a custom field. To do this, you need to be able to identify both the segment and the relevant values.

For details, see the following topics:

- Identifying a Segment Field
- Identifying a Segment Value
- Example

Identifying a Segment Field

You identify a segment field in the same way you would a custom field: by using its field ID. To find the field ID, view an instance of the record type that you want to interact with. Click the label of the field to display a window that includes the field ID.



If you have permission to view custom segment definitions, you might notice that the segment definition lists a script ID and record ID. You should **not** refer to either of these IDs when setting a value for the segment on a record instance.

Additionally, be aware that a single custom segment can appear on instances of multiple record types. The field ID for the segment can vary among record types. You must use the field ID as it appears on an instance of the **same record type** where you want your script to set a segment value.

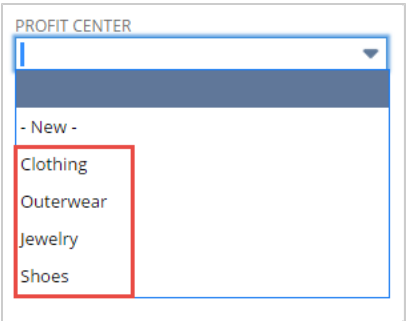
Identifying a Segment Value

Typically, each custom segment includes a list of possible values. To set a value for a segment by using SuiteScript, at least one value must already have been defined for the segment.

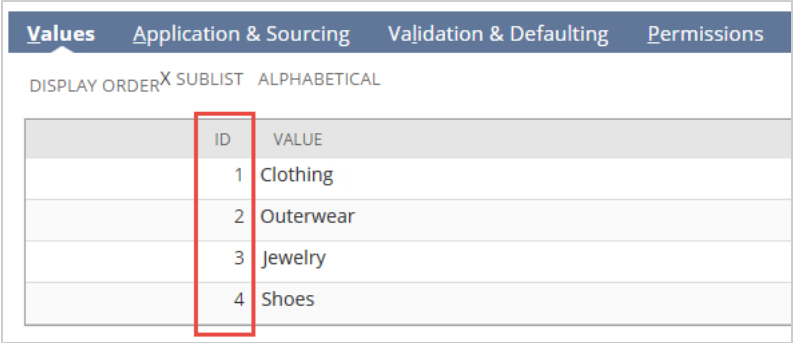
You can reference a custom segment value by specifying either of the following:

- The value's text label
- The value's internal ID

The text label is shown in the custom segment definition and on instances of the record type where the segment has been applied. On the record instance, the segment's values are shown in either a multiselect box or a dropdown list.



To find a segment value's internal ID, open the segment definition at Customization > Lists, Records, & Fields > Custom Segments. To see a list of all possible values, refer to the Values sublist. This list includes a column labeled ID, which identifies each value's internal ID.



Example

The following snippet shows how you could use SuiteScript 1.0 to set a value for a segment with the field ID of custbody_cseg_profitcenter:

```
nlapiSetFieldText('custbody_cseg_profitcenter', 'Outerwear');
```

Custom Transaction

Note: The content in this help topic pertains to all versions of SuiteScript.

You use the custom transaction record to interact with instances of existing custom transaction types. For example, suppose you had a custom transaction type called Non-Operational Income Entry. In this case, you could use SuiteScript to create and modify non-operational income entries.

The internal ID for this record varies depending on the ID value of the transaction type. For example, if your transaction type had an ID value of `_noie`, the SuiteScript internal ID would be `customtransaction_noie`. You can view the transaction type's ID value by opening the type for

editing. To open the type for editing, go to Customization > Lists, Records, & Fields > Transaction Types, and select the appropriate type.

Supported Script Types

Custom transaction record instances are supported in both client and server SuiteScript.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

Custom transaction records are fully scriptable. They can be created, updated, copied, searched, and deleted using SuiteScript.

For more details see the following table.

Function	Supported?	Notes
nlapiCopyRecord	Yes	
nlapiCreateRecord	Yes	
nlapiDeleteRecord	Yes	
nlapiLoadRecord	Yes	
nlapiSearchRecord	Yes	You use the standard transaction search to search custom transactions. For more information, see Transaction Search.
nlapiSubmitRecord	Yes	
nlapiTransformRecord	No	
nlapiVoidTransaction	Yes	nlapiVoidTransaction is supported only for transaction types configured to support the void option. In other words, on the transaction type definition, the box labeled **Allow Void Transactions Using Reversal Journals** must be checked. Direct voids are never permitted with custom transactions. Only voids through reversal journals are permitted.

Usage Notes

In the UI, custom transactions can have a field labeled Status. In SuiteScript, this field has an internal ID of transtatus. Values for transtatus are set by the system when the statuses are created. They cannot be modified. Possible values are A through Z. In the UI, you can view a transaction type's available statuses (and their transtatus values) by opening the transaction type and navigating to the Statuses subtab. The transtatus value for each status is listed in the Code column.

Code Sample

The following sample shows how to create an instance of a custom transaction type.

```
var transaction = nlapiCreateRecord('customtransaction_mytype',true);
transaction.setFieldValue('subsidiary','1');

// Add debit line
transaction.selectNewLineItem('line');
transaction.setCurrentLineItemValue('line', 'account', '1');
```

```
transaction.setCurrentLineItemValue('line','debit',10);
transaction.setCurrentLineItemValue('line','credit',0);
transaction.setCurrentLineItemValue('line', 'memo', 'My first custom transaction line');
transaction.commitLineItem('line');

// Add credit line
transaction.selectNewLineItem('line');
transaction.setCurrentLineItemValue('line', 'account', '2');
transaction.setCurrentLineItemValue('line','debit',0);
transaction.setCurrentLineItemValue('line','credit',10);
transaction.setCurrentLineItemValue('line', 'memo', 'My second custom transaction line');
transaction.commitLineItem('line');

nlapiSubmitRecord(transaction);
```

Scheduled Script Instance

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

You can use the scheduled script instance (`scheduledscriptinstance`) search to return information about individual scheduled script jobs and map/reduce script jobs. A subset of this information is displayed on the following pages in the NetSuite UI:

- **Scheduled Scripts:** Each entry on the Scheduled Script Status page contains information about a single scheduled script instance. Each scheduled script instance is handled by one scheduled script job. To access the Scheduled Script Status page in the UI, go to .
- **Map/Reduce Scripts:** Each entry on the Map/Reduce Script Status page contains information about a single map/reduce script instance. Each map/reduce script instance is handled by multiple map/reduce script jobs. To access the Map/Reduce Script Status page in the UI, go to . You can access information about the individual map/reduce jobs for a particular map/reduce script instance from the map/reduce script details page. To access this page from the Map/Reduce Script Status page, click the Details link for the appropriate script instance.

For more information on the Scheduled Script Status page, see the help topic Monitoring a Scheduled Script's Runtime Status or SuiteScript 2.0 Scheduled Script Status Page. For more information on the Map/Reduce Script Status page, see the help topic SuiteScript 2.0 Map/Reduce Script Status Page.

For additional information, see:

- Usage Notes
- Code Samples

Usage Notes

You can script searches on `scheduledscriptinstance` with all server-side script types. Scheduled script instance searches are not exposed to client-side scripts.

Important: You can only use the scheduled script instance record type to perform searches.

For a list of SuiteScript supported `scheduledscriptinstance` fields, see the Records Browser. In addition, see the following table for supplemental information on three supported fields.

Field	Notes
Original Priority (`originalpriority`)	This field holds the original priority setting (low, standard, or high) for the scheduled script job or map/reduce job. The priority setting is the value set when the scheduled script job or map/reduce job is initially created. ▪ If a job is created when a deployment is submitted for processing, the original priority is the priority set on the deployment record at the time it was submitted for processing. ▪ If a job is created during script processing, the original priority is the current priority of the job that created the new job. For more information on deployment record priority settings, see SuiteCloud Processors – Priority Levels. **Note:** For SuiteScript 1.0 and 2.0 scheduled scripts, one scheduled script job is created when a script deployment is submitted for processing. In addition, a SuiteScript 1.0 scheduled script can yield. When yielding occurs, an additional scheduled script job is created during script processing. For map/reduce scripts, the initial map/reduce jobs are created when the script deployment is submitted for processing. In addition, more map/reduce jobs can be created during script processing.
Original Priority Number (`originalprioritynumber`)	This field holds a numeric value that corresponds to the original priority value: ▪ High priority: 1 ▪ Standard priority: 2 ▪ Low priority: 3 You cannot group your search results by the original priority field because it holds a text value. The original priority number field is provided as an alternative.
Priority Timestamp (`prioritytimestamp`)	This field holds the time that the current priority is set for the scheduled script job or map/reduce job. This field can reflect the timestamp of a user-defined priority setting change or a system-defined priority setting change. User-defined priority changes do not impact the value of this field for existing jobs. However, system-defined priority changes may impact this value for existing jobs. For example, if you create a scheduled script deployment and immediately submit the deployment for processing, the priority timestamp for that job reflects the deployment submission time. If you then change the priority on the deployment record, neither the priority nor the priority timestamp for the existing job is changed. However, If you enable priority elevation, and the priority for the existing job is changed by the system, the priority timestamp value is changed accordingly. For more information on priority elevation, see SuiteCloud Processors – Priority Elevation and Processor Reservation (Advanced Settings).

Code Samples

The following SuiteScript 1.0 code filters the search for all schedule script instance records that are in progress in queue 2. It uses search return columns to return all available information on the filtered results. Note that the first two search return columns are joins on the script and script deployment record types.

```
var colArr = new Array();
var filterArr = new Array();

colArr.push(new nlobjSearchColumn('name', 'script'));
colArr.push(new nlobjSearchColumn('internalid', 'scriptdeployment'));
```

```
colArr.push(new nlobjSearchColumn('datecreated'));
colArr.push(new nlobjSearchColumn('status'));
colArr.push(new nlobjSearchColumn('startdate'));
colArr.push(new nlobjSearchColumn('enddate'));
colArr.push(new nlobjSearchColumn('queue'));
colArr.push(new nlobjSearchColumn('percentcomplete'));
colArr.push(new nlobjSearchColumn('queueposition'));
colArr.push(new nlobjSearchColumn('percentcomplete'));
filterArr.push(new nlobjSearchFilter('queue', null, 'is' , "2"));
filterArr.push(new nlobjSearchFilter('status', null, 'anyof' , ['PROCESSING']));
var searchResults =  nlapiSearchRecord('scheduledscriptinstance', null, filterArr, colArr);
```

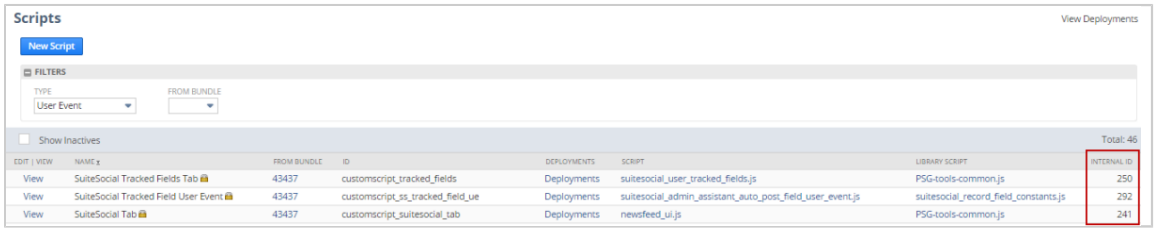
Script

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

To load and submit a script record, use one of the following case-insensitive script types for the type argument:

- 'bundleinstallationscript'
- 'clientscript'
- 'massupdatescript'
- 'portlet'
- 'restlet'
- 'scheduledscript'
- 'suitelet'
- 'usereventscript'
- 'workflowactionscript'

For the id argument, you must use the script record's internal ID (for example, 191); the script record's ID (for example, `customscript_csalert` is not supported. You can find a script record's internal ID at Customization > Scripting > Scripts. If you have the **Show Internal IDs** preference enabled at Home > Set Preferences, internal IDs are listed in the Internal ID column.



See the SuiteScript Records Browser for all other internal IDs associated with this record. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

Use nlapiSearchRecord(type, id, filters, columns), nlapiLoadSearch(type, id) and nlapiCreateSearch(type, filters, columns) to perform searches on script records. Note that when you use one of these APIs to search for script records, you must pass in 'script' as the type parameter. This differs from how you load script records with nlapiLoadRecord; in those instances you pass in the individual script type.

For additional information about the script record, see:

- Usage Notes
- Supported Functions
- Code Samples

Usage Notes

You can read and edit script records with all server-side script types. Script records are not exposed to client-side scripts.

Important: Script records cannot be created, copied or deleted programmatically.

The following objects are not scriptable, or are read only, on the script record:

- The Change ID button (in Edit view of UI) is not scriptable
- The Unhandled Errors tab is not scriptable
- The History tab is not scriptable
- The System Notes subtab is not scriptable
- The Execution Log tab is not scriptable
- The Libraries tab is not scriptable
- The Script Type field is read only; editing is not supported
- The Parameters tab is read only; editing is not supported

To access script records within a script, the script owner must belong to a role assigned with SuiteScript permissions.

Important: Scripts can only access script records when the records are part of the same bundle.

Scripts that are not part of a bundle cannot access script records that are part of a bundle.

Supported Functions

Use the following APIs to load and submit script records.

- nlapiLoadRecord(type, id, initializeValues)
- nlapiSubmitRecord(record, doSourcing, ignoreMandatoryFields)

Use the following nlobjRecord methods to read and edit script records:

- commitLineItem(group, ignoreRecalc)
- findLineItemValue(group, fldnam, value)
- getAllFields()
- getAllLineItemFields(group)
- getCurrentLineItemDateTimeValue(type, fieldId, timeZone)
- getCurrentLineItemValues(type, fldnam)
- getField(fldnam)
- getFieldText(name)

- getFieldValue(name)
- getId()
- getLineItemCount(group)
- getLineItemDateTimeValue(type, fieldId, lineNum, timeZone)
- getLineItemField(group, fldnam, linenum)
- getLineItemText(group, fldnam, linenum)
- getLineItemValue(group, name, linenum)
- getLineItemValues(type, fldnam, linenum)
- getRecordType()
- insertLineItem(group, linenum, ignoreRecalc)
- removeLineItem(group, linenum, ignoreRecalc)
- selectLineItem(group, linenum)
- selectNewLineItem(group)
- setCurrentLineItemDateTimeValue(type, fieldId, dateTime, timeZone)
- setCurrentLineItemValue(group, name, value)
- setFieldText(name, text)
- setFieldValue(name, value)
- setLineItemDateTimeValue(type, fieldId, lineNum, dateTime, timeZone)
- setLineItemValue(group, name, linenum, value)

Use the following Search APIs to perform searches on script records:

- nlapiCreateSearch(type, filters, columns)
- nlapiLoadSearch(type, id)
- nlapiSearchRecord(type, id, filters, columns)

Code Samples

```
//read a suitelet script record
var rec = nlapiLoadRecord('suitelet',605);
var a = rec.getFieldValue('scriptfile');
var count = rec.getLineItemCount('parameters');
var pLabel = rec.getLineItemValue('parameters', 'label', 1);
var pId = rec.getLineItemValue('parameters', 'internalid', 1);
var pType = rec.getLineItemValue('parameters', 'fieldtype', 1);
var pRecordType = rec.getLineItemValue('parameters', 'selectrecordtype', 1);
```

```
//edit a user event script record
var rec = nlapiLoadRecord('usereventscript', 302);
rec.setFieldValue('name', 'userevent_001');
rec.setFieldValue('scriptfile', '227');
rec.setFieldValue('notifyadmins', 'T');
rec.setFieldValue('aftersubmitfunction', 'afterSubmitFunction');
nlapiSubmitRecord(rec);
```

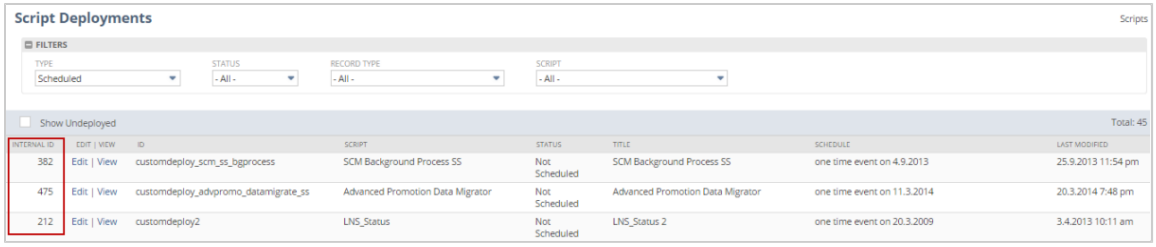
```
//search a script record
```

```
var filters = [new nlobjSearchFilter('defaultfunction', null, 'is', 'myfunctionname')];
var columns = [new nlobjSearchColumn('name')];
var results = nlapiSearchRecord('script', null, filters, columns);
```

Script Deployment

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

To load, submit, copy, create and delete a script deployment record, use 'scriptdeployment' for the type argument. Note that 'scriptdeployment' is case-insensative. For the id argument, you must use the script deployment record's internal ID (for example, 88); the script deployment record's ID (for example, 'customdeploy_newsfeed_email') is not supported. You can find a script deployment record's internal ID at Customization > Scripting > Script Deployments. If you have the `Show Internal IDs` preference enabled at Home > Set Preferences, internal IDs are listed in the Internal ID column.



See the SuiteScript Records Browser for all other internal IDs associated with this record. For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For additional information about the script deployment record, see:

- Usage Notes
- Supported Functions
- Code Samples

Usage Notes

You can access script deployment records with all server-side script types. Script deployment records are not exposed to client-side scripts.

The following objects are not scriptable, or are read only, on the script deployment record:

- Change ID button (in Edit view of UI) is not scriptable
- The History subtab is not scriptable
- The System Notes subtab is not scriptable
- The Execution Log subtab is not scriptable
- The Scheduling subtab is not scriptable
- The Parameters subtab is read only; editing is not supported

To access script deployment records within a script, the script owner must belong to a role assigned with SuiteScript permissions.

Important: Scripts can only access script deployment records when the records are part of the same bundle.

Scripts that are not part of a bundle cannot access script deployment records that are part of a bundle.

Supported Functions

Use the following APIs to copy, create, delete, load and submit script deployment records.

- nlapiCopyRecord(type, id, initializeValues)
- nlapiCreateRecord(type, initializeValues)
- nlapiDeleteRecord(type, id, initializeValues)
- nlapiLoadRecord(type, id, initializeValues)
- nlapiSubmitRecord(record, doSourcing, ignoreMandatoryFields)

Use the following nlobjRecord methods to read and edit script deployment records:

- commitLineItem(group, ignoreRecalc)
- findLineItemValue(group, fldnam, value)
- getAllFields()
- getAllLineItemFields(group)
- getCurrentLineItemDateTimeValue(type, fieldId, timeZone)
- getCurrentLineItemValues(type, fldnam)
- getField(fldnam)
- getFieldText(name)
- getFieldValue(name)
- getId()
- getLineItemCount(group)
- getLineItemDateTimeValue(type, fieldId, lineNum, timeZone)
- getLineItemField(group, fldnam, linenum)
- getLineItemText(group, fldnam, linenum)
- getLineItemValue(group, name, linenum)
- getLineItemValues(type, fldnam, linenum)
- getRecordType()
- insertLineItem(group, linenum, ignoreRecalc)
- removeLineItem(group, linenum, ignoreRecalc)
- selectLineItem(group, linenum)
- selectNewLineItem(group)
- setCurrentLineItemDateTimeValue(type, fieldId, dateTime, timeZone)
- setCurrentLineItemValue(group, name, value)
- setFieldText(name, text)
- setFieldValue(name, value)

- setLineItemDateTimeValue(type, fieldId, lineNum, dateTime, timeZone)
- setLineItemValue(group, name, linenum, value)

Use the following Search APIs to perform searches on script deployment records:

- nlapiCreateSearch(type, filters, columns)
- nlapiLoadSearch(type, id)
- nlapiSearchRecord(type, id, filters, columns)

Code Samples

```
//create a script deployment record
var rec = nlapiCreateRecord('scriptdeployment', {'script':'2'});
```

```
//edit a script deployment record
var rec = nlapiLoadRecord('scriptdeployment',65);
rec.setFieldValue('allroles', 'T');
nlapiSubmitRecord(rec);
```

```
//read a script script deployment record
var rec = nlapiLoadRecord('scriptdeployment', 203);
var a = rec.getFieldValue('status')
```

```
//copy and edit a script deployment record
var rec = nlapiCopyRecord('scriptdeployment', 41);
rec.setFieldValue('isdeployed', 'F');
rec.setFieldValue('title', 'Feature 4141 - Suitelet Scratch 4');
nlapiSubmitRecord(rec);
```

```
//search a script deployment record
var filters = [new nlobjSearchFilter('internalidnumber', null, 'anyof', '43')];
var columns = [new nlobjSearchColumn('title')];
var results = nlapiSearchRecord('scriptdeployment', null, filters, columns);
```

Marketing

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The following marketing records are scriptable in SuiteScript:

- Campaign
- Campaign Response
- Campaign Template
- Coupon Code
- Email Template

- Promotion

Campaign

Note: The content in this help topic pertains to all versions of SuiteScript.

Campaign records are used to manage your marketing initiatives. With campaign records, you can manage your online marketing as well as your offline efforts. You can create and send email marketing messages, manage paid keywords, send promotions through the mail, and manage your advertising campaigns. For details, see the help topics Campaign Overview and Creating a Campaign Record.

The internal ID for this record is `campaign`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The campaign record is scriptable in both client and server SuiteScript.

Supported Functions

The campaign record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Campaign Response

Note: The content in this help topic pertains to all versions of SuiteScript.

The campaign response record helps you refine how you deliver marketing campaigns. For details, see the help topic Tracking Campaign Responses.

To view this record in the user interface, navigate to an entity record and click the **Marketing** subtab. On the Campaigns subtab, click the **Add Response** button.

The internal ID for this record is `campaignresponse`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The campaign response record is scriptable in both client and server SuiteScript.

Supported Functions

The campaign response record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Campaign Template

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The campaign template record is used for both CRMSDK templates and scriptable templates. In the UI, you find this record by choosing Documents > Templates > Marketing Templates

Scriptable templates are not supported with SuiteScript.

The internal ID for this record is `campaigntemplate`.

See the SuiteScript Records Browser for the internal IDs of all fields associated with this record, their corresponding labels in the UI, and more details. Fields not listed in the Records Browser are not supported and should not be used.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The campaign template record is scriptable in server SuiteScript only.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

You can use nlapiMergeRecord(id, baseType, baseId, altType, altId, fields) with CRMSDK templates to perform a mail merge. You can also use SuiteScript to read and delete CRMSDK templates. You cannot create, edit, copy, search, or transform CRMSDK templates with SuiteScript.

Field Definitions

This section describes some of the key fields on the campaign template record.

Content and Mediaitem

When accessing this record, only one of the following fields contains a value:

- content — This field maps to the text editor that appears on the Template tab in the UI. This field contains plain text or HTML.
- mediaitem — This field maps to the File text box that appears on the Template tab in the UI. This field identifies the file that is used as the basis for the template.

Code Sample

The following sample shows how to access a campaign template email.

Accessing Template Content

The following sample shows how to access template content:

```
var template = nlapiLoadRecord('campaigntemplate', 154);

if (template.getFieldValue('mediaitem') != null) {
      var media = template.getFieldValue('mediaitem'));
      // do something with the media...
} else {
      var content = template.getFieldValue('content');
      // do something with the content...
}
```

Coupon Code

Note: The content in this help topic pertains to all versions of SuiteScript.

Promotions allow you to track the source of revenue and to offer discounts in the form of coupons. Each promotion has a promotion code that can be applied to transactions and campaigns. Enable Promotion Codes on the Transactions subtab at Setup > Company > Setup Tasks > Enable Features (Administrator). For details, see the help topic Selling With Promotion Codes

The internal ID for this record is `couponcode`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords

- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The coupon code record is scriptable in both client and server SuiteScript.

Supported Functions

The coupon code record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Email Template

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The email template record is used for both CRMSDK templates and scriptable templates. In the UI, you find this record by choosing Documents > Templates > Email Templates. This record is available only if the Customer Relationship Management feature has been enabled at Setup > Company > Enable Features, on the CRM tab. For details, see the help topic Working with Email Templates.

Scriptable templates are not supported with SuiteScript.

The internal ID for this record is `emailtemplate`.

See the SuiteScript Records Browser for the internal IDs of all fields associated with this record, their corresponding labels in the UI, and more details. Fields not listed in the Records Browser are not supported and should not be used.

> **Note:** For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

This record is only supported in server SuiteScript.

All three user events are supported: beforeLoad, beforeSubmit, and afterSubmit.

Supported Functions

You can use nlapiMergeRecord(id, baseType, baseId, altType, altId, fields) with CRMSDK templates to perform a mail merge. You can also use SuiteScript to read and delete CRMSDK templates. You cannot create, edit, copy, search, or transform CRMSDK templates with SuiteScript.

Field Definitions

This section describes some of the key fields on the email template record.

Content and Mediaitem

When accessing this record, only one of the following fields contains a value:

- content — This field maps to the text editor that appears on the Template tab in the UI. This field contains plain text or HTML.
- mediaitem — This field maps to the File text box that appears on the Template tab in the UI. This field identifies the file that is used as the basis for the template.

Code Sample

The following example shows how you can script with email template records.

Accessing Template Content

The following sample shows how to access template content:

```
var template = nlapiLoadRecord('emailtemplate', 124);

if (template.getFieldValue('mediaitem') != null) {
      var media = template.getFieldValue('mediaitem'));
      // do something with the media...
} else {
      var content = template.getFieldValue('content');
      // do something with the content...
}
```

Promotion

Note: The content in this help topic pertains to all versions of SuiteScript.

Promotions allow you to create special offers to motivate your customers to purchase products in higher quantities. Promotions can help you to move out-of-date stock, increase sales, and reward valuable clients. Promotions provide discounts that customers can apply to Web store orders and orders placed with sales reps. Furthermore, NetSuite promotions enable you to target specific customers, locations or channels, and time periods. For details, see the help topics Promotions Overview, Configuring Promotions, and The Promotion Record.

The internal ID for this record is `promotioncode`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords

- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The promotion record is scriptable in both client and server SuiteScript.

Supported Functions

The promotion record is fully scriptable — it can be created, updated, copied, deleted, and searched using SuiteScript.

Usage Notes

The following table provides usage notes for specific fields on this record.

Field Internal ID	Field UI Label	Note
Body Fields		
discounttype	radio buttons	Valid values in scripts are: ▪ percent ▪ flat

When applying a free gift promotion to a transaction using SuiteScript, the free gift is not automatically added or removed from the order.

The following SuiteScript example shows how to apply the free gift and remove it when the transaction is no longer eligible:

```
// Create Sales Order
var params = {};
params.recordmode = 'dynamic';
var rec = nlapiCreateRecord('salesorder', params);
rec.setFieldValue('entity', '41');

// Add free gift promotion to the promotion machine
rec.selectNewLineItem('promotions');
rec.setCurrentLineItemValue('promotions', 'promocode', '103');
rec.commitLineItem('promotions');

// Add eligible item with quantity 4 to the item machine
rec.setCurrentLineItemValue('item', 'item', '233');
rec.setCurrentLineItemValue('item', 'quantity', '4');
rec.commitLineItem('item');


// *********** Manually add free gift and link it to the free gift promotion ***********
rec.setCurrentLineItemValue('item', 'item', '244');
rec.setCurrentLineItemValue('item', 'quantity', '2');
rec.setCurrentLineItemValue('item', 'freegiftpromotion', '103');
rec.commitLineItem('item');
```

```
 // Check Free Gift Promotion colums on Promotion Machine:
 // firstFreeGiftQuantity should say 2
 // firstEligibleFreeGifts should say 2
 // firstFreeGiftsAdded should say 2
var firstFreeGiftQuantity= rec.getLineItemValue('item', 'quantity', 2);
var firstEligibleFreeGifts = rec.getLineItemValue('promotions', 'eligiblefreegifts', 1);
var firstFreeGiftsAdded = rec.getLineItemValue('promotions', 'freegiftsadded', 1);


// *********** Reduce eligible item quantity to 3 ***********
rec.selectLineItem('item', '1');
rec.setCurrentLineItemValue('item', 'quantity', '3');
rec.commitLineItem('item');

// Verify that the columns have been updated but the user still needs to manually remove 1 Free
 Gift
// secondFreeGiftQuantity should say 2
// secondEligibleFreeGifts should say 1
// secondFreeGiftsAdded should say 2
var secondFreeGiftQuantity= rec.getLineItemValue('item', 'quantity', 2);
var secondEligibleFreeGifts = rec.getLineItemValue('promotions', 'eligiblefreegifts', 1);
var secondFreeGiftsAdded = rec.getLineItemValue('promotions', 'freegiftsadded', 1);


// *********** Manually reduce Free Gift quantity ***********
rec.selectLineItem('item', '2');
rec.setCurrentLineItemValue('item', 'quantity', '1');
rec.commitLineItem('item');

// Verify that the promotions machine columns have been updated again
// Verify that the columns have been updated but the user still needs to manually remove 1 Free
 Gift
// firstFreeGiftQuantity should say 1
// firstEligibleFreeGifts should say 1
// firstFreeGiftsAdded should say 1
var thirdFreeGiftQuantity= rec.getLineItemValue('item', 'quantity', 2);
var thirdEligibleFreeGifts = rec.getLineItemValue('promotions', 'eligiblefreegifts', 1);
var thirdFreeGiftsAdded = rec.getLineItemValue('promotions', 'freegiftsadded', 1);
```

Working with the Discount Amount Search Column

When you use the Discount Amount search column, instead of using the discountamount internal ID, use discountrate.

Website

> **Note:** The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The following website records are scriptable in SuiteScript:

- CMS Content

- CMS Content Type
- CMS Page
- Commerce Category
- Website Setup
- Shopping Cart

CMS Content

Note: The content in this help topic applies to all versions of SuiteScript. Currently, it may only include links or examples for SuiteScript 1.0.

CMS content enables you to create things such as text and images for your website. For more information, see the help topic Site Management Tools.

The internal ID for this record is `cmscontent`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The CMS content record is scriptable in server SuiteScript only.

Supported Functions

The CMS content record is partially scriptable. It can be created, updated, and searched using SuiteScript. It cannot be copied, attached, or transformed.

Usage Notes

You must enable the Site Management Tools feature on the Web Presence subtab at Setup > Company > Enable Features to be able to script with this record.

Script Sample

The following sample code snippet creates a script that will search for a CMS content record.

```
function searchCmsContentRecords(siteId)
{
   var searchFlters = [new nlobjSearchFilter('site', null, 'is', siteId)];

   var searchColumns = [new nlobjSearchColumn('name'),
```

```
                    new nlobjSearchColumn('site'),
                    new nlobjSearchColumn('cmscontenttype')];

   return nlapiSearchRecord('cmscontent', null, searchFlters, searchColumns);
}
```

CMS Content Type

Note: The content in this help topic applies to all versions of SuiteScript. Currently, it may only include links or examples for SuiteScript 1.0.

CMS page enables you to create different content types for your website. For more information, see the help topic Site Management Tools.

The internal ID for this record is `cmspage`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The CMS content type record is scriptable in server SuiteScript only.

Supported Functions

The CMS content type record is partially scriptable. It can be created, updated, and searched using SuiteScript. It cannot be copied, attached, or transformed.

Usage Notes

You must enable the Site Management Tools feature on the Web Presence subtab at Setup > Company > Enable Features to be able to script with this record.

Script Samples

The following sample code snippets show how to create and search for CMS content type records. All CMS content type records are associated with a custom record. For more information, see Custom Record.

Create CMS Content Type Record for SuiteScript 1.0

The following sample code snippet creates a basic CMS content type record and associates it with the specified custom record.

```
var cmsContentTypeRecord = nlapiCreateRecord('cmscontenttype', false);

cmsContentTypeRecord.setFieldValue('name', 'cms content type name');
cmsContentTypeRecord.setFieldValue('label', 'content type label');
cmsContentTypeRecord.setFieldValue('description', 'My cms content type description');
cmsContentTypeRecord.setFieldValue('customrecordid', <customRecordID>);

contentTypeID = nlapiSubmitRecord(cmsContentTypeRecord);
```

Create CMS Content Type Record for SuiteScript 2.0

The following sample code snippet creates a basic CMS content type record and associates it with the specified custom record.

```
var recordObj = record.create({
    type: record.Type.CMS_CONTENT_TYPE,
    isDynamic: false
});

recordObj.setValue('name', 'cms content type name');
recordObj.setValue('label', 'content type label');
recordObj.setValue('description', 'My cms content type description');
recordObj.setValue('customrecordid', <customRecordID>);

var recordId = recordObj.save({
      enableSourcing: false,
      ignoreMandatoryFields: false
});
```

Search CMS Content Type Records for SuiteScript 1.0

The following sample code snippet searches for all CMS content type records for a given SCA site. It returns the name, site, pagetype, and url fields as search result columns.

```
function searchCmsContentTypeRecords()
{
   var searchColumns = [new nlobjSearchColumn('name'),
                    new nlobjSearchColumn('label'),
                    new nlobjSearchColumn('customrecordid')];

   return nlapiSearchRecord('cmscontenttype', null, null, searchColumns);
}
```

CMS Page

Note: The content in this help topic applies to all versions of SuiteScript. Currently, it may only include links or examples for SuiteScript 1.0.

CMS page enables you to create different pages for hosting content on your website. For more information, see the help topic Landing and Enhanced Pages.

The internal ID for this record is `cmspage`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The CMS page record is scriptable in server SuiteScript only.

Supported Functions

The CMS page record is partially scriptable. It can be created, updated, and searched using SuiteScript. It cannot be copied, attached, or transformed.

Usage Notes

You must enable the Site Management Tools feature on the Web Presence subtab at Setup > Company > Enable Features to be able to script with this record.

Script Samples

The following sample code snippets show how to create and search for CMS page records.

Create CMS Page Record for SuiteScript 1.0

The following sample code snippet creates a basic landing page record (pagetype =1) for the specified SCA site.

```
var cmsPageRecord = nlapiCreateRecord('cmspage', false);

cmsPageRecord.setFieldValue('name', 'Cms Page Name');
cmsPageRecord.setFieldValue('site', <siteID>);
cmsPageRecord.setFieldValue('pagetype', 1); // 1 = landing & 2 = enhanced
cmsPageRecord.setFieldValue('url', 'PageUrl');

var cmsPageId = nlapiSubmitRecord(cmsPageRecord);
```

Create CMS Page Record for SuiteScript 2.0

The following sample code snippet creates a basic landing page record (pagetype = 1) for the specified SCA site.

```
var recordObj = record.create({
    type: record.Type.CMS_PAGE,
    isDynamic: false
});
```

```
recordObj.setValue('name', 'Cms Page Name');
recordObj.setValue('site', <siteID>);
recordObj.setValue('pagetype', 1);
recordObj.setValue('url', 'PageUrl');

return = recordObj.save({
    enableSourcing: false,
    ignoreMandatoryFields: false
});
```

Search CMS Page Records for SuiteScript 1.0

The following sample code snippet searches for all CMS page records for a given SCA site. It returns the name, site, pagetype, and url fields as search result columns.

```
function searchCmsPageRecords(siteId)
{
   var searchFlters = [new nlobjSearchFilter('site', null, 'is', siteId)];

   var searchColumns = [new nlobjSearchColumn('name'),
                   new nlobjSearchColumn('site'),
                   new nlobjSearchColumn('pagetype'),
                   new nlobjSearchColumn('url')];

   return nlapiSearchRecord('cmspage', null, searchFlters, searchColumns);
}
```

Commerce Category

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

Commerce Categories enable you to create a hierarchical structure of product categories, subcategories, and products. For details, see the help topic Commerce Categories.

In the UI, you can go to Lists > Web Site > Commerce Categories to access this record.

The internal ID for this record is `commercecategory`.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The commerce category record is scriptable in server SuiteScript only.

Supported Functions

The commerce category record is partially scriptable. It can be created, updated, and searched using SuiteScript. It cannot be copied, attached, or transformed.

Usage Notes

You must enable the Commerce Category feature on the Web Presence subtab at Setup > Company > Enable Features to be able to script with this record.

Script Sample

The following sample code snippet searches for a commerce category record with a fullurl of "/cat-0" and siteid of 2. It returns fullurl, addtohead, and subcataddtoheadoverride as search result columns.

```
var fullurl = '/cat-0';
var siteid = 2;

var searchFilters = [
        new nlobjSearchFilter('fullurl', null, 'is', fullurl)
        , new nlobjSearchFilter('site', null, 'is', siteid)
];
var searchColumns = [
        new nlobjSearchColumn('fullurl')
        , new nlobjSearchColumn('addtohead')
        , new nlobjSearchColumn('subcataddtoheadoverride')
                                        ];
var search = nlapiSearchRecord('commercecategory', null, searchFilters, searchColumns);
```

Website Setup

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The internal ID for this record is `website`. In the UI, you can find this record by going to Setup > Web Site > Set Up Web Site.

See the SuiteScript Records Browser for all internal IDs associated with this record.

Note: For information on using the SuiteScript Records Browser, see Using the SuiteScript Records Browser in the NetSuite Help Center.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The web site setup record is scriptable in server SuiteScript only.

The web site record is not supported in beforeLoad user event scripts.

Supported Functions

The web site setup record is fully scriptable. It can be created, copied, updated, deleted, and searched using SuiteScript.

Usage Notes

Developers can use SuiteScript with the web site setup record whether they are working in accounts using Site Builder or SuiteCommerce Advanced.

Creating a Web Site Setup Record in SuiteScript

To create a Web Site Setup record in a **Site Builder** site, developers can write code using one of two approaches:

The first approach is:

```
var recWebSite = nlapiCreateRecord('website');
```

The second approach is:

```
var initvalues = new Array();
initvalues.sitetype = 'STANDARD';
var recWebSite = nlapiCreateRecord('website', initvalues);
```

In the first approach, the initvalues are defaulted to STANDARD. However, it is recommended that developers use the second approach because there are two website types. To clearly distinguish between the two types in your code, the second approach is cleaner.

To create a Web Site Setup record in a **SuiteCommerce Advanced** site, the code **must** look like the following:

```
var initvalues = new Array();
initvalues.sitetype = 'ADVANCED';
var recWebSite = nlapiCreateRecord('website', initvalues);
```

Setting Values for Web Site Setup Dropdown Fields

The following table lists the dropdown fields on the Web Site Setup record. When writing SuiteScript, if you are setting the value of a dropdown field, use the IDs listed in the column called Internal IDs for Dropdown Values.

Tab	Dropdown Field	UI Labels for Dropdown Values	Internal IDs for Dropdown Values
Setup tab			
	Web Site Scope (websitescope)	▪ Full Web Store ▪ Information And Catalog, With Pricing ▪ Information And Catalog	▪ FULL_WEB_STORE ▪ INFO_CATALOG_PRICING ▪ INFO_CATALOG

Tab	Dropdown Field	UI Labels for Dropdown Values	Internal IDs for Dropdown Values
		▪ Information Only	▪ INFO_ONLY
	Default Customer Category (defaultcustomercategory)	▪ Corporate ▪ Individual ▪ Employee	▪ CORPORATE ▪ INDIVIDUAL ▪ EMPLOYEE
Appearance tab			
	Web Site Logo Alignment (websitelogoalign)	▪ Align Left ▪ Align Right ▪ Align Center	▪ LEFT ▪ RIGHT ▪ CENTER
	Page Alignment (pagealign)	▪ Align Left ▪ Align Right ▪ Align Center	▪ LEFT ▪ RIGHT ▪ CENTER
	Display Order of Cart Items (cartdisplayorder)	▪ Most Recently Added First ▪ Most Recently Added Last	▪ RECENT_FIRST ▪ RECENT_LAST
Upsell tab			
	Items to Upsell (upsellitems)	▪ Show Related Items First and Upsell Items Next ▪ Show Upsell Items First and Related Items Next ▪ Show Only Related Items ▪ Show Only Upsell Items	▪ RELATED_FIRST_UPSELL_NEXT ▪ UPSELL_FIRST_RELATED_NEXT ▪ ONLY_RELATED_ITEMS ▪ ONLY_UPSELL_ITEMS
	Items to Upsell in Cart (cartupsellitems)	▪ Show Related Items First and Upsell Items Next ▪ Show Upsell Items First and Related Items Next ▪ Show Only Related Items ▪ Show Only Upsell Items	▪ RELATED_FIRST_UPSELL_NEXT ▪ UPSELL_FIRST_RELATED_NEXT ▪ ONLY_RELATED_ITEMS ▪ ONLY_UPSELL_ITEMS
Legacy tab			
	Site Tab Alignment (sitetabalignment)	▪ Align Left ▪ Align Right ▪ Align Center	▪ LEFT ▪ RIGHT ▪ CENTER
Shopping tab			
	Sales Order Type (salesordertype)	▪ Per Customer Basis	▪ PER_CUSTOMER

Shopping Cart

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The internal ID for this record is `shoppingcart`. You can use the UI to create saved searches with data from the shopping cart. Go to Lists > Search > Saved Searches > New > Shopping cart.

For information about scripting with this record in SuiteScript, see the following:

- SuiteScript 2.0 Scripting Records and Subrecords
- N/record Module
- SuiteScript 1.0 Scripting Records, Subrecords, Fields, Forms, and Sublists

Supported Script Types

The shopping cart record is scriptable in server SuiteScript only.

Supported Functions

The shopping cart record is not fully scriptable. Only search is permitted.

Usage Notes

You can use SuiteScript to create sophisticated marketing campaigns based upon the data linked to the shopping cart. The following data points are exposed through SuiteScript: customer ID, date created, date edited, item IDs for cart items, and website ID.

Note: Because shopping carts can be created by anonymous shoppers, in some search results the customer ID may be null.

Script Samples

The following example uses the SuiteScript 1.0 `nlapiSearchRecord` method to get shopping cart information.

To get carts from a certain website (id:1) which contains a particular item (item id:352), use the following code sample:

```
var result = nlapiSearchRecord('ShoppingCart', null, [ new nlobjSearchFilter('WEBSITEID', null,
 'equalto', '1'), new nlobjSearchFilter('ITEMID', null, 'equalto', '352') ], [ new nlobjSearchC
olumn('ITEMID'), new nlobjSearchColumn('ITEMQTY'), new nlobjSearchColumn('WEBSITEID'), new lobj
SearchColumn('CUSTOMERID')])
```

You can also use `nlapiLoadSearch` to programmatically execute a previously created shopping cart saved search. Or, you can use methods from the SuiteScript 2.0 N/search module.

SuiteScript IDs

- Permission Names and IDs
- Feature Names and IDs
- Preference Names and IDs
- Task IDs
- Button IDs

Permission Names and IDs

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The following table provides permission names and IDs associated with each NetSuite feature. You can use the permission ID with nlobjContext.getSetting(type, name) to return the permission levels that have been specified in your account.

Permission ID	Permission Name	Feature
ADMI_ACCOUNTING	Set Up Accounting	Accounting
ADMI_ACCOUNTINGBOOK	Accounting Book	Accounting
ADMI_ACCOUNTINGLIST	Accounting Lists	Accounting
ADMI_ACCTPERIODS	Manage Accounting Periods	Accounting Periods
ADMI_ACCTSETUP	Set Up Accounting	Accounting
ADMI_ACH	Set Up ACH Processing	ACH Processing
ADMI_ADVANCED_ORDER_MANAGEMENT	Advanced Order Management	Advanced Order Management
ADMI_APP_DEPLOYMENT	SuiteApp Deployment	SuiteApp
ADMI_APPPUBLISHER	Application Publishers	SSP Applications
ADMI_AUDITLOGIN	View Login Audit Trail	Login Audit Trail
ADMI_BACKUPEXPORT	Backup Your Data	CSV Export
ADMI_BILLPAYSETUP	Set Up Bill Pay	Online Bill Pay
ADMI_BUNDLER	SuiteBundler	SuiteBundler
ADMI_BUNDLERMANUP	SuiteBundler Upgrade Install Base	SuiteBundler
ADMI_CAMPAIGNEMAIL	Set Up Campaign Email Addresses	Marketing Automation
ADMI_CAMPAIGNSETUP	Setup Campaigns	Marketing Automation
ADMI_CASEFORM	Online Case Form	Customer Support and Service
ADMI_CASEISSUE	Support Case Issue	Customer Support and Service
ADMI_CASEORIGIN	Support Case Origin	Customer Support and Service
ADMI_CASEPRIORITY	Support Case Priority	Customer Support and Service

Permission ID	Permission Name	Feature
ADMI_CASERULE	Support Case Territory Rule	Customer Support and Service
ADMI_CASESTATUS	Support Case Status	Customer Support and Service
ADMI_CASETERRITORY	Support Case Territory	Customer Support and Service
ADMI_CASETYPE	Support Case Type	Customer Support and Service
ADMI_CLASSESTOLOCS	Convert Classes to Locations	Locations
ADMI_CLASSSEGMENTMAPPING	Class Segment Mapping	Classes
ADMI_CLOSEPERIOD	Lock Transactions	Accounting Periods
ADMI_COMMISSIONSETUP	Commission Feature Setup	Commissions
ADMI_COMPANY	Company Information	Company Setup
ADMI_CONVERTCLASSES	Convert Classes to Departments	Departments
ADMI_CONVERTLEAD	Lead Conversion Mapping	Sales Force Automation
ADMI_COPYPROJECTTASK	Copy Project Tasks	Project Management
ADMI_CREDITCARD	Credit Card Processing	Credit Card Payments
ADMI_CRMLIST	CRM Lists	Opportunities
ADMI_CSVIMPORTPREF	Set Up CSV Preferences	Accounting
ADMI_CUSTADDRESSFORM	Custom Address Form	Custom Forms
ADMI_CUSTBODYFIELD	Custom Body Fields	Custom Fields
ADMI_CUSTCATEGORY	Custom Center Categories	Custom Centers
ADMI_CUSTCENTER	Custom Centers	Custom Centers
ADMI_CUSTCOLUMNFIELD	Custom Column Fields	Custom Fields
ADMI_CUSTEMAILLAYOUT	Custom HTML Layouts	Custom PDF/HTML Templates
ADMI_CUSTENTITYFIELD	Custom Entity Fields	Custom Fields
ADMI_CUSTENTRYFORM	Custom Entry Forms	Custom Forms
ADMI_CUSTEVENTFIELD	Custom Event Fields	Custom Fields
ADMI_CUSTFIELD	Custom Fields	SuiteFlow Custom Fields
ADMI_CUSTFIELDTAB	Custom Subtabs	Custom Subtabs
ADMI_CUSTFORM	Custom Transaction Forms	Custom Forms
ADMI_CUSTITEMFIELD	Custom Item Fields	Custom Fields
ADMI_CUSTITEMNUMBERFIELD	Custom Item Number Fields	Inventory
ADMI_CUSTLAYOUT	Custom PDF Layouts	Custom PDF/HTML Templates
ADMI_CUSTLIST	Custom Lists	Custom Lists
ADMI_CUSTOMERFORM	Online Customer Form	Sales Force Automation
ADMI_CUSTOMSCRIPT	SuiteScript	Client SuiteScript
ADMI_CUSTOMSUBLIST	Custom Sublists	Custom Sublists

Permission ID	Permission Name	Feature
ADMI_CUSTOTHERFIELD	Other Custom Fields	Custom Fields
ADMI_CUSTRECORD	Custom Record Types	Custom Records
ADMI_CUSTRECORDFORM	Online Custom Record Form	Custom Forms
ADMI_CUSTSECTION	Custom Center Tabs	Custom Records
ADMI_CUSTTASKS	Custom Center Links	Custom Records
ADMI_CUSTTRANSACTION	Custom Transaction Types	Custom Transactions
ADMI_DELETEDRECORD	Deleted Records	Search for Deleted Records
ADMI_DEPTSEGMENTMAPPING	Department Segment Mapping	Departments
ADMI_DEVICE_ID	Device ID Management	Suite Commerce InStore
ADMI_DOMAINS	Set Up Domains	SuiteCommerce
ADMI_DUPLICATESETUP	Duplicate Detection Setup	Duplicate Detection & Merge
ADMI_EMPLCATEGORY	Publish Employee List	SuiteCommerce
ADMI_EMPLOYEELIST	Other Lists	Accounting
ADMI_ENABLEFEATURES	Enable Features	Enable Company Features
ADMI_ENTITYACCOUNTMAPPING	Entity Account Mapping	Accounting
ADMI_ENTITYSTATUS	Customer Status	Customer Relationship Management
ADMI_ESCALATIONRULE	Escalation Assignment Rule	Customer Support and Service
ADMI_ESCALATIONTERRITORY	Escalation Assignment	Customer Support and Service
ADMI_EXPORTIIF	Export as IIF	Accounting
ADMI_FINCHARGEPREF	Finance Charge Preferences	A/R
ADMI_GAINLOSSACCTMAPPING	Foreign Currency Variance Mapping	Accounting
ADMI_GLOBALACCOUNTMAPPING	Global Account Mapping	Accounting
ADMI_IMPORTCSVFILE	Import CSV File	CSV Import
ADMI_IMPORTDEFAULT	Delete All Data	Delete All Data
ADMI_IMPORTQIFFILE	Import Quicken QIF File	Accounting
ADMI_IMPORTXML	Set Up ADP Payroll	Import ADP Payroll
ADMI_INTEGRAPP	Integration Application	Integration
ADMI_ISSUESETUP	Issue Setup	Issue Management
ADMI_ITEMACCOUNTMAPPING	Item Account Mapping	Accounting
ADMI_KNOWLEDGEBASE	Publish Knowledge Base	Knowledge Base
ADMI_KPIREPORT	KPI Scorecards	KPI Scorecards
ADMI_LOCATIONCOSTINGGROUP	Location Costing Group	Item Record Management
ADMI_LOCSEGMENTMAPPING	Location Segment Mapping	Custom Segments

Permission ID	Permission Name	Feature
ADMI_MANAGE_OAUTH_TOKENS	Access Token Management	Token-based Authentication
ADMI_MANAGE_OWN_OAUTH_TOKENS	User Access Tokens	Token-based Authentication
ADMI_MANAGECUSTOMSEGMENTS	Custom Segments	Custom Segments
ADMI_MANAGEROLES	Bulk Manage Roles	Show Role Differences
ADMI_MANAGEUSERS	Manage Users	Managing Users
ADMI_OPENIDSSOSETUP	Set Up OpenID Single Sign-on	OpenID Single Sign-on
ADMI_OUTLOOKINTEGRATION	Outlook Integration	Outlook Integration
ADMI_OUTLOOKINTEGRATION_V3	Outlook Integration 3.0	Outlook Integration
ADMI_PAYROLL	Set Up Payroll	Payroll
ADMI_PROJECT_ACCOUNTING_SETUP	Project Profitability Setup	Project Management
ADMI_PUBLISHER_APP	Publisher Applications	SuiteCommerce
ADMI_PUBLISHER_ID	Publisher ID	SuiteCommerce
ADMI_REVIEW_CUSTOM_GL_RUNS	Review Custom GL plug-in executions	Custom GL Lines Plug-in
ADMI_SALESTERRITORY	Sales Territory	Sales Force Automation
ADMI_SAMLSSOSETUP	Set Up SAML Single Sign-on	SAML Single Sign-on
ADMI_SAVEDASHBOARD	Publish Dashboards	Publishing Dashboards
ADMI_SETUPCOMPANY	Set Up Company	Company Preferences
ADMI_SETUPIMAGERESIZE	Set Up Image Resizing	SuiteCommerce Advanced
ADMI_SETUPYEARSTATUS	Set Up Year Status	Accounting
ADMI_SFASETUP	Sales Force Automation Setup	Sales Force Automation
ADMI_STATETAXIMPORT	Import State Sales Tax	Accounting
ADMI_STORESETUP	Set Up Web Site	Web Site
ADMI_SUITEANALYTICSCONNECT	SuiteAnalytics Connect	SuiteAnalytics Connect
ADMI_SUITESIGNON	SuiteSignOn	Outbound Single Sign-on
ADMI_SUPPORTSETUP	Support Setup	Customer Support and Service
ADMI_SWAPPRICES	Swap Prices Between Price Levels	Multiple Prices
ADMI_SYNCHRONIZATIONSETUP	Set up synchronization	Synchronization
ADMI_TAXPERIODS	Manage Tax Reporting Periods	Accounting
ADMI_TRANSITEMTXT	Translation	Multi-Language
ADMI_TWOFACTORAUTH	Two-Factor Authentication	Two-Factor Authentication
ADMI_TWOFACTORAUTHBASE	Two-Factor Authentication base	Two-Factor Authentication
ADMI_UNCATSITEITEMS	Uncategorized Presentation Items	Web Store

Permission ID	Permission Name	Feature
ADMI_UPDATEPRICES	Update Prices	Item Record Management
ADMI_UPSELLSETUP	Upsell Setup	Upsell Manager
ADMI_USERPREF	User Preferences	Set Preferences
ADMI_WEBSERVICES	Web Services	SuiteTalk
ADMI_WEBSERVICESLOG	Web Services Usage Log	SuiteTalk
ADMI_WEBSERVICESSETUP	Set Up Web Services	SuiteTalk
ADMI_WORKFLOW	Workflow	SuiteFlow
LIST_ACCOUNT	Accounts	Accounting
LIST_AMORTIZATION	Amortization Schedules	Amortization
LIST_BIG_SEARCH	Persist Search	Search
LIST_BILLINGSCHEDULE	Billing Schedules	SuiteBilling
LIST_BILLOFDISTRIBUTION	Bill Of Distribution	Advanced Inventory Management
LIST_BILLOFMATERIALSINQUIRY	Bill Of Materials Inquiry	Assembly Items
LIST_BOM	Bill of Materials	Assembly Items
LIST_BIN	Bins	Accounting
LIST_CALENDAR	Calendar	Calendar Preferences
LIST_CALL	Phone Calls	Phone Calls
LIST_CAMPAIGN	Marketing Campaigns	Marketing Automation
LIST_CAMPAIGNHISTORY	Campaign History	Marketing Automation
LIST_CASE	Cases	Customer Support and Service
LIST_CATEGORY	Expense Categories	Expense Reports
LIST_CHECKITEMAVAILABILITY	Check Item Availability	Advanced Inventory Management
LIST_CLASS	Classes	Classes
LIST_COLORTHEME	Color Themes	SuiteCommerce
LIST_COMMISSIONRULES	Employee Commission Schedules/Plans	Employee Commissions
LIST_COMPANY	Companies	Customer Relationship Management
LIST_COMPETITOR	Competitors	Sales Force Automation
LIST_COMPONENTWHEREUSEDINQUIRY	Component Where Used	Assembly Items
LIST_CONTACT	Contacts	Contacts
LIST_CONTACTROLE	Contact Roles	Sales Force Automation
LIST_COSTEDBOMINQUIRY	Costed Bill Of Materials Inquiry	Assembly Items
LIST_CRMGROUP	CRM Groups	Customer Relationship Management

Permission ID	Permission Name	Feature
LIST_CRMMESSAGE	Track Messages	Customer Relationship Management
LIST_CRMTEMPLATE	Marketing Template	Marketing Automation
LIST_CURRENCY	Currency	Multiple Currencies
LIST_CUSTJOB	Customers	Prospects and Contacts
LIST_CUSTPROFILE	Customer Profile	Customer Profile
LIST_CUSTRECORDENTRY	Custom Record Entries	Custom Records
LIST_DEPARTMENT	Departments	Departments
LIST_DISTRIBUTIONNETWORK	Distribution Network	Advanced Inventory Management
LIST_EMAILTEMPLATE	Email Template	Email Template
LIST_EMPLOYEE	Employees	Employees
LIST_EMPLOYEE_NAVIGATION	Employee Navigation	Employees
LIST_EMPLOYEE_SEARCH	Employee Search	Employees
LIST_EMPLOYEESSN	Employee Social Security Numbers	Employees
LIST_ENTITY_DUPLICATES	Duplicate Entity Management	Duplicate Detection & Merge
LIST_EVENT	Events	Events
LIST_EXPORT	Export Lists	Search Result Export
LIST_FAIRVALUEDIMENSION	Fair Value Dimension	Revenue Recognition
LIST_FAIRVALUEFORMULA	Fair Value Formula	Revenue Recognition
LIST_FAIRVALUEPRICE	Fair Value Price	Revenue Recognition
LIST_FAXMESSAGE	Fax Messages	Communications
LIST_FAXTEMPLATE	Fax Template	Mail Merge
LIST_FILECABINET	Documents and Files	File Cabinet
LIST_FIND	Perform Search	Order Management
LIST_FISCALCALENDAR	Fiscal Calendars	Accounting
LIST_GENERICRESOURCE	Generic Resources	Project Management
LIST_GLLINESAUDITLOG	Custom GL Lines Plug-in Audit Log	Custom GL Lines Plug-in
LIST_GLLINESAUDITLOGSEG	Custom GL Lines Plug-in Audit Log (Segments)	Custom GL Lines Plug-in
LIST_HISTORY	Notes Tab	Communications
LIST_INFOITEM	Store Content Items	SuiteCommerce
LIST_INFOITEMFORM	Publish Forms	SuiteCommerce
LIST_INTEGRAPP	Integration Applications	SuiteTalk
LIST_INVCOSTTEMPLATE	Inventory Cost Template	Multi-Book Accounting

Permission ID	Permission Name	Feature
LIST_INVENTORYSTATUS	Inventory Status	Inventory Management
LIST_ISSUE	Issues	Issue Management
LIST_ITEM	Items	Item Record Management
LIST_ITEM_REVISION	Item Revisions	Item Record Management
LIST_ITEMDEMANDPLAN	Item Demand Plan	Advanced Inventory Management
LIST_ITEMREVENUECATEGORY	Item Revenue Category	Revenue Recognition
LIST_ITEMSUPPLYPLAN	Item Supply Plan	Advanced Inventory Management
LIST_JOB	Jobs	Projects
LIST_KNOWLEDGEBASE	Knowledge Base	Knowledge Base
LIST_LOCATION	Locations	Locations
LIST_MAILMERGE	Mail Merge	Mail Merge
LIST_MAILMESSAGE	Letter Messages	Communications
LIST_MAILTEMPLATE	Letter Template	Mail Merge
LIST_CRMTEMPLATE	Marketing Template	Marketing
LIST_MASSUPDATES	Mass Updates	Mass Update
LIST_MEDIAITEMFOLDER	Media Folders	File Cabinet
LIST_MEMDOC	Memorized Transactions	Transactions
LIST_MFGCOSTTEMPLATE	Manufacturing Cost Template	Inventory Management
LIST_MFGROUTING	Manufacturing Routing	Inventory Management
LIST_NOTIFICATION	Notifications	
LIST_ORDER_REALLOCATION	Commit Orders	Advanced Inventory Management
LIST_OTHERNAME	Other Names	Sales Force Automation
LIST_PARTNER	Partners	Partner Relationship Management
LIST_PARTNERCOMMISSNRULES	Partner Commission Schedules/Plans	Partner Commissions/ Royalties
LIST_PAYCHECK	Paychecks	Payroll
LIST_PAYMETH	Payment Methods	Order Management
LIST_PAYROLLITEM	Payroll Items	Payroll
LIST_PDFMESSAGE	PDF Messages	Communications
LIST_PDFTEMPLATE	PDF Template	Mail Merge
LIST_PLANNEDREVENUE	Planned Revenue	Revenue Recognition
LIST_PLANNEDSTANDARDCOST	Planned Standard Cost	Item Record Management
LIST_PRESCATEGORY	Presentation Categories	SuiteCommerce

Permission ID	Permission Name	Feature
LIST_PROJECTREVENUERULE	Project Revenue Rules	Project Management
LIST_PROJECTTASK	Project Tasks	Project Management
LIST_PROJECTTEMPLATE	Project Templates	Project Management
LIST_PROMOTIONCODE	Promotion Code	Sales Force Automation
LIST_PUBLISHSEARCH	Publish Search	Publishing Search Results
LIST_QUANTITYPRICINGSCHEDULE	Quantity pricing Schedules	Quantity Pricing
LIST_RELATEDITEMS	Related Items	Web Site
LIST_RESOURCE	Resource	Project Management
LIST_REVENUEELEMENT	Revenue Element	Revenue Recognition
LIST_REVENUEPLAN	Revenue Recognition Plan	Revenue Recognition
LIST_REVENUERECOGNITIONRULE	Revenue Recognition Rule	Revenue Recognition
LIST_REVRECSCHEDULE	Revenue Recognition Schedules	Revenue Recognition
LIST_REVRECFIELDMAPPING	Revenue Recognition Field Mapping	Revenue Recognition
LIST_REVRECSCHEDULE	Revenue Recognition Schedules	Revenue Recognition
LIST_REVRECVSOE	Revenue Management VSOE	VSOE
LIST_RSRCALLOCATION	Resource Allocations	Project Management
LIST_RSRCALLOCATIONAPPRV	Resource Allocation Approval	Project Management
LIST_SALESCAMPAIGN	Sales Campaigns	Sales Campaigns
LIST_SALESROLE	Sales Roles	Team Selling
LIST_SHIPITEM	Shipping Items	Accounting
LIST_SHIPPARTPACKAGE	Shipping Partner Package	Order Management
LIST_SHIPPARTREGISTRATION	Shipping Partner Registration	Order Management
LIST_SHIPPARTSHIPMENT	Shipping Partner Shipment	Order Management
LIST_SITEEMAILTEMPLATE	Web Store Email Template	SuiteCommerce
LIST_STANDARDCOSTVERSION	Standard Cost Version	Item Record Management
LIST_STORETAB	Store Tabs	Web Store
LIST_SUBSCRIPTION	Subscriptions	Advanced Subscription Billing
LIST_SUBSCRIPTIONPLAN	Subscription Plan	Advanced Subscription Billing
LIST_SUBSIDIARY	Subsidiaries	Subsidiaries
LIST_SYSTEMEMAILTEMPLATE	System Email Template	Customer Relationship Management
LIST_TASK	Tasks	Customer Relationship Management

Permission ID	Permission Name	Feature
LIST_TAXITEM	Tax Items	Accounting
LIST_TAXSCHEDULE	Tax Schedules	Advanced Taxes
LIST_TEGATAACCOUNT	Tegata Accounts	Accounting
LIST_TEMPLATE_CATEGORY	Template Categories	Email Marketing Campaigns
LIST_TIMEOFFADMIN	Time-Off Administration	Time-Off Management
LIST_TRANNUMBERAUDITLOG	Access to transaction numbering audit log	Transactions
LIST_UNIT	Units	Accounting
LIST_UPSELL	Upsell Assistant	Upsell Manager
LIST_UPSELLWIZARD	Upsell Wizard	Upsell Manager
LIST_VENDOR	Vendors	Accounting
LIST_WEBSITE	Website (External) publisher	Web Site
LIST_WORKCALENDAR	Work Calendar	Project Management
LIST_WORKPLACE	Workplaces	Payroll
REGT_ACCTPAY	Accounts Payable Register	A/P
REGT_ACCTREC	Accounts Receivable Register	A/R
REGT_BANK	Bank Account Registers	Accounting
REGT_COGS	Cost of Goods Sold Registers	Accounting
REGT_CREDCARD	Credit Card Registers	Accounting
REGT_DEFEREXPENSE	Deferred Expense Registers	Amortization
REGT_DEFERREVENUE	Deferred Revenue Registers	Revenue Recognition
REGT_EQUITY	Equity Registers	Accounting
REGT_EXPENSE	Expense Registers	Accounting
REGT_FIXEDASSET	Fixed Asset Registers	Accounting
REGT_INCOME	Income Registers	Accounting
REGT_LONGTERMLIAB	Long Term Liability Registers	Accounting
REGT_NONPOSTING	Non Posting Registers	Accounting
REGT_OTHASSET	Other Asset Registers	Accounting
REGT_OTHCURRASSET	Other Current Asset Registers	Accounting
REGT_OTHCURRLIAB	Other Current Liability Registers	Accounting
REGT_OTHINCOME	Other Income Registers	Accounting

Permission ID	Permission Name	Feature
REGT_UNBILLEDREC	Unbilled Receivable Registers	Revenue Commitments
REPO_1099	Form 1099 - Federal Miscellaneous Income	A/P
REPO_940	Form 940 - Employer's Annual Federal Unemployment Tax Return	Payroll
REPO_941	Form 941 - Employer's Quarterly Federal Tax Return	Payroll
REPO_ACCOUNTDETAIL	Account Detail	Accounting
REPO_AMORTIZATION	Amortization Reports	Amortization
REPO_AP	Accounts Payable	A/P
REPO_AR	Accounts Receivable	A/R
REPO_AUTHPARTNERCOMMISSION	Partner Authorized Commission Reports	Partner Commissions/ Royalties
REPO_BALANCESHEET	Balance Sheet	Accounting
REPO_BOOKINGS	Sales Order Reports	Sales Force Automation
REPO_CASHFLOW	Cash Flow Statement	Accounting
REPO_COMMISSION	Commission Reports	Employee Commissions
REPO_CUSTOMIZATION	Report Customization	Report Customization
REPO_FINANCIALS	Financial Statements	Accounting
REPO_GL	General Ledger	Accounting
REPO_INTEGRATION	Integration	SuiteTalk
REPO_INVENTORY	Inventory	Inventory
REPO_ISSUE	Issue Reports	Issue Management
REPO_MARKETING	Marketing Campaign Reports	Marketing Automation
REPO_NONPOSTING	Sales Order Transaction Report	Order Management
REPO_PANDL	Income Statement	Accounting
REPO_PARTNERCOMMISSION	Partner Commission Reports	Partner Commissions/ Royalties
REPO_PARTNERMONITOR	Partner Monitor Results	Partners
REPO_PAYCHECKDETAIL	Payroll Check Register	Payroll
REPO_PAYROLL	Payroll Summary & Detail Reports	Payroll
REPO_PAYROLLHOURSEARNING	Payroll Hours & Earnings	Payroll
REPO_PAYROLLJOURNAL	Payroll Journal Report	Payroll
REPO_PAYROLLLIAB	Payroll Liability Report	Payroll

Permission ID	Permission Name	Feature
REPO_PAYROLLSTATEWITHHOLD	Payroll State Withholding	Payroll
REPO_PAYROLLW2	Form W-2 - Wage and Tax Statement	Payroll
REPO_PROJECT_ACCOUNTING	Project Accounting	Accounting
REPO_PURCHASEORDER	Purchase Order Reports	Purchase Orders
REPO_PURCHASES	Purchases	Accounting
REPO_QUOTA	Quota Reports	Sales Force Automation
REPO_RECONCILE	Reconcile Reporting	Accounting
REPO_REMINDEREMPLOYEE	Employee Reminders	Employees
REPO_RETURNAUTH	Return Authorization Reports	Return Authorizations
REPO_REVREC	Revenue Recognition Reports	Revenue Recognition
REPO_RSRCALLOCATION	Resource Allocation Reports	Project Management
REPO_SALES	Sales	Sales Force Automation
REPO_SALESORDER	Sales Order Fulfillment Reports	Sales Orders
REPO_SALES_PARTNER	Sales By Partner	Partner Relationship Management
REPO_SALES_PROMO	Sales By Promotion Code	Sales Force Automation
REPO_SCHEDULE	Report Scheduling	Reports
REPO_SFA	Sales Force Automation	Customer Relationship Management
REPO_SUPPORT	Support	Customer Support and Service
REPO_TAX	Tax	Accounting
REPO_TIME	Time Tracking	Time Tracking
REPO_TRAN	Transaction Detail	Transactions
REPO_TRIALBALANCE	Trial Balance	Accounting
REPO_UNBILLED	Accounts Receivable Un-Billed	Accounting
REPO_WEBSITE	Web Site Report	Web Site
REPO_WEBSTORE	Web Store Report	Web Store
TRAN_ADJUSTMENTJOURNAL	Currency Adjustment Journal	Accounting
TRAN_ALLOCSCHEDULE	Create Allocation Schedules	Expense Allocation
TRAN_AMENDW4	Amend W-4	Employees
TRAN_APPROVECOMMISSN	Employee Commission Transaction Approval	Employee Commissions

Permission ID	Permission Name	Feature
TRAN_APPROVEDD	Approve Direct Deposit	Direct Deposit
TRAN_APPROVEEFT	Approve EFT	Electronic Funds Transfer
TRAN_APPROVEPARTNERCOMM	Partner Commission Transaction Approval	Partner Commissions/ Royalties
TRAN_APPROVEVP	Approve Vendor Payments	ACH Vendor Payments
TRAN_AUDIT	Audit Trail	Transactions
TRAN_BILLPAY_APPROVE	Approve Online Bill Payments	Online Bill Pay
TRAN_BILLPAY_STATUS	View Online Bill Pay Status	Online Bill Pay
TRAN_BINTRNFR	Bin Transfer	Bin Management
TRAN_BINWKSHT	Bin Putaway Worksheet	Bin Management
TRAN_BLANKORD	Blanket Purchase Order	Purchasing and Receiving
TRAN_BLANKORDAPPRV	Blanket Purchase Order Approval	Purchasing and Receiving
TRAN_BUDGET	Set Up Budgets	Accounting
TRAN_BUILD	Build Assemblies	Assembly Items
TRAN_CARDCHRG	Credit Card	Accounting
TRAN_CASHRFND	Cash Sale Refund	Accounting
TRAN_CASHSALE	Cash Sale	Accounting
TRAN_CHARGE	Charge	Project Management
TRAN_CHARGERULE	Charge Rule	Project Management
TRAN_CHECK	Check	Accounting
TRAN_CLEARHOLD	Override Payment Hold	Order Management
TRAN_COMMISSN	Employee Commission Transaction	Employee Commissions
TRAN_COPY_BUDGET	Copy Budgets	Accounting
TRAN_CUSTAUTH	Customer Payment Authorization	Customer Payment Authorizations
TRAN_CUSTCHRG	Statement Charge	A/R
TRAN_CUSTCRED	Credit Memo	A/R
TRAN_CUSTDEP	Customer Deposit	A/R
TRAN_CUSTINVC	Invoice	A/R
TRAN_CUSTINVCAPPRV	Invoice Approval	Order Management
TRAN_CUSTPYMT	Customer Payment	A/R
TRAN_CUSTRFND	Customer Refund	A/R
TRAN_DEPAPPL	Deposit Application	A/R
TRAN_DEPOSIT	Deposit	Accounting
TRAN_EDITPROFILE	Edit Profile	Employees

Permission ID	Permission Name	Feature
TRAN_ESTIMATE	Estimate	Estimates
TRAN_EXPREPT	Expense Report	Expense Reports
TRAN_FFTREQ	Fulfillment Request	Order Management
TRAN_FINCHRG	Finance Charge	A/R
TRAN_FIND	Find Transaction	Transactions
TRAN_FINDOLBMATCH	Find Matching Online Banking Transactions	Accounting
TRAN_FORECAST	Edit Forecast	Sales Force Automation
TRAN_FXREVAL	Currency Revaluation	Multiple Currencies
TRAN_GST_REFUND	Process GST Refund	Accounting
TRAN_IMPORTOLBFILE	Import Online Banking (QIF) File	Accounting
TRAN_INTERCOADJ	Intercompany Adjustments	Accounting
TRAN_INVADJST	Adjust Inventory	Inventory
TRAN_INVSTATUSCHANGE	Inventory Status Change	Inventory Managment
TRAN_INVCOUNT	Count Inventory	Advanced Inventory Management
TRAN_INVDISTR	Distribute Inventory	Multi-Location Inventory
TRAN_INVREVAL	Revalue Inventory Cost	Inventory
TRAN_INVTRNFR	Transfer Inventory	Multi-Location Inventory
TRAN_INVWKSHT	Adjust Inventory Worksheet	Inventory
TRAN_ITEMRCPT	Receive Items	Advanced Receiving
TRAN_ITEMSHIP	Ship Items	Advanced Shipping
TRAN_JOURNAL	Make Journal Entry	Accounting
TRAN_JOURNALAPPRV	Journal Approval	Accounting
TRAN_LIABPYMT	Payroll Liability Payments	Payroll
TRAN_MGRFORECAST	Edit Manager Forecast	Sales Force Automation
TRAN_OLBSTATEMENT	Online Banking Statement	Accounting
TRAN_OPENBAL	Enter Opening Balances	Accounting
TRAN_OPPRTNTY	Opportunity	Opportunities
TRAN_PARTNERCOMMISSN	Partner Commission Transaction	Partner Commissions/ Royalties
TRAN_PAYCHECK	Individual Paycheck	Payroll
TRAN_PAYMENTAUDIT	Access Payment Audit Log	Order Management
TRAN_PAYROLLRUN	Process Payroll	Payroll
TRAN_PCHKJRNL	Paycheck Journal	Employees
TRAN_POSTVENDORBILLVARIANCE	Post Vendor Bill Variances	Vendors

Permission ID	Permission Name	Feature
TRAN_PRICELIST	Generate Price Lists	Item Record Management
TRAN_PRINT	Print/Email/Fax	Transactions
TRAN_PRINTCHECKSFORMS	Print Checks and Forms	Accounting
TRAN_PURCHCON	Purchase Contract	Vendors
TRAN_PURCHCONAPPRV	Purchase Contract Approval	Vendors
TRAN_PURCHORD	Purchase Order	Purchase Orders
TRAN_PURCHORDBILL	Bill Purchase Orders	Advanced Receiving
TRAN_PURCHORDRECEIVE	Receive Purchase Orders	Purchase Orders
TRAN_PURCHREQ	Requisition	Purchasing and Receiving
TRAN_PURCHREQAPPRV	Requisition Approval	Purchasing and Receiving
TRAN_QUOTA	Establish Quotas	Sales Force Automation
TRAN_RECOG_GIFTCERT_INCOME	Recognize Gift Certificate Income	Accounting
TRAN_RECONCILE	Reconcile	Accounting
TRAN_REVARRNG	Revenue Arrangement	Revenue Recognition
TRAN_REVARRNGAPPRV	Revenue Arrangement Approval	Revenue Recognition
TRAN_REVCOMM	Revenue Commitment	Revenue Commitments
TRAN_REVCOMRV	Revenue Commitment Reversal	Revenue Commitments
TRAN_REVCONTR	Revenue Contracts	Revenue Recognition
TRAN_RFQ	Request For Quote	Purchasing and Receiving
TRAN_RTNAUTH	Return Authorization	Return Authorizations
TRAN_RTNAUTHAPPRV	Return Auth. Approval	Return Authorizations
TRAN_RTNAUTHCREDIT	Refund Returns	Order Management
TRAN_RTNAUTHRECEIVE	Receive Returns	Order Management
TRAN_RTNAUTHREVERSEREVCOMMIT	Generate Revenue Commitment Reversals	Revenue Commitments
TRAN_SALESORD	Sales Order	Sales Orders
TRAN_SALESORDAPPRV	Sales Order Approval	Sales Orders
TRAN_SALESORDCOMMITREVENUE	Generate Revenue Commitment	Revenue Commitments
TRAN_SALESORDFULFILL	Fulfill Orders	Order Management
TRAN_SALESORDINVOICE	Bill Sales Orders	Advanced Shipping
TRAN_SALESORDREVENUECONTRACT	Generate Single Order Revenue Contracts	Revenue Recognition
TRAN_STATEMENT	Generate Statements	A/R
TRAN_STATUSDD	Direct Deposit Status	Direct Deposit

Permission ID	Permission Name	Feature
TRAN_STATUSEFT	EFT Status	Electronic Funds Transfer
TRAN_STATUSVP	Vendor Payment Status	ACH Vendor Payments
TRAN_STPICKUP	Store Pickup Fulfillment	SuiteCommerce InStore
TRAN_TAXLIAB	Pay Tax Liability	Accounting
TRAN_TAXPYMT	Pay Sales Tax	Accounting
TRAN_TEGPYBL	Tegata Payable	Accounting
TRAN_TEGRCVBL	Tegata Receivable	Accounting
TRAN_TIMEBILL	Track Time	Time Tracking
TRAN_TIMEPOST	Post Time	Project Management
TRAN_TRANSFER	Transfer Funds	Accounting
TRAN_TRNFRORD	Transfer Order	Inventory Management
TRAN_TRNFRORDAPPRV	Transfer Order Approval	Inventory Management
TRAN_UNBUILD	Unbuild Assemblies	Assembly Items
TRAN_VENDAUTH	Vendor Return Authorization	Vendor Return Authorizations
TRAN_VENDAUTHAPPRV	Vendor Return Auth. Approval	Vendor Return Authorizations
TRAN_VENDAUTHCREDIT	Credit Returns	Vendor Returns
TRAN_VENDAUTHRETURN	Vendor Returns	Vendor Return Authorizations
TRAN_VENDBILL	Bills	A/P
TRAN_VENDBILLAPPRV	Vendor Bill Approval	Vendor Bills
TRAN_VENDCRED	Enter Vendor Credits	A/P
TRAN_VENDPYMT	Pay Bills	A/P
TRAN_VENDRFQ	Vendor Request For Quote	Purchasing and Receiving
TRAN_WOCLOSE	Work Order Close	Inventory Management
TRAN_WOCOMPL	Work Order Completion	Inventory Management
TRAN_WOISSUE	Work Order Issue	Inventory Management
TRAN_WORKORD	Work Order	Work Orders
TRAN_WORKORDBUILD	Build Work Orders	Work Orders
TRAN_WORKORDCLOSE	Close Work Orders	Inventory Management
TRAN_WORKORDCOMPLETE	Enter Completions	Inventory Management
TRAN_WORKORDISSUE	Issue Components	Inventory Management
TRAN_WORKORDMARKBUILT	Mark Work Orders Built	Inventory Management
TRAN_WORKORDMARKFIRMED	Mark Work Orders Firmed	Inventory Management
TRAN_WORKORDMARKRELEASED	Mark Work Orders Released	Inventory Management
TRAN_YTDADJST	Enter Year-To-Date Payroll Adjustments	Payroll

Feature Names and IDs

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The following table provides the internal IDs and feature names for all NetSuite features. You can use the feature ID with nlobjContext.getSetting(type, name) to see if a particular feature is enabled in your account.

Feature Internal ID	Feature Name
ACCOUNTING	Accounting
ACCOUNTINGPERIODS	Accounting Periods
ACHVEND	ACH Vendor Payments
ADVANCEDJOBS	Project Management
ADVANCEDPRINTING	Advanced PDF/HTML Templates
ADVANCEDPROCUREMENTAPPROVALS	Advanced Procurement Approvals
ADVANCEDPROMOTIONS	Advanced Promotions
ADVANCEDREVENUERECOGNITION	Advanced Revenue Management
ADVANCEDREVENUERECOGNITIONAPP	Advanced Revenue Recognition SuiteApp
ADVANCEDSITECUST	Advanced Site Customization
ADVANCEDSITEMANAGEMENT	Site Management Tools
ADVBILLING	Advanced Billing
ADVBINSERIALLOTMGMT	Advanced Bin/Numbered Inventory Management
ADVFORECASTING	Advanced Forecasting
ADVINVENTORYMGMT	Advanced Inventory Management
ADVPARTNERACCESS	Advanced Partner Access
ADVRECEIVING	Advanced Receiving
ADVSHIPPING	Advanced Shipping
ADVTAXENGINE	Advanced Taxes
ADVWEBREPORTS	Advanced Web Reports
ADVWEBSEARCH	Advanced Web Search
ALTERNATIVEPAYMENTS	Alternative Payments
ALTSALESADVFORECAST	ALT_SALES Advanced Forecasting
ALTSALESAMOUNT	Alternate Sales Amount
AMORTIZATION	Amortization
APPROVALROUTING	Approval Routing
ASSEMBLIES	Assembly Items
ASYNCCUSTOMER	Asynchronous AfterSubmit Customer Processing
ASYNCSALESORDER	Asynchronous AfterSubmit Sales Order Processing

Feature Internal ID	Feature Name
AUTOAPPLYPROMOTIONS	Auto-Apply Promotions
AUTOLOCATIONASSIGNMENT	Automatic Location Assignment
AVAILABLETOPROMISE	Available To Promise
BARCODES	Bar Coding and Item Labels
BILLINGACCOUNTS	Billing Accounts
BILLINGCLASSES	Per-Employee Billing Rates
BILLINGRATECARDS	Billing Rate Cards
BILLINGWORKCENTER	Billing Operations
BILLSCOSTS	Bill Costs To Customers
BINMANAGEMENT	Bin Management
BLANKETPURCHASEORDERS	Blanket Purchase Orders
BOXNET	Box Document Management
CAMPAIGNASSISTANT	Campaign Assistant
CAMPAIGNSUBSCRIPTIONS	Subscription Categories
CCTRACKING	Credit Card Payments
CHARGEBASEDBILLING	Charge-Based Billing
CHECKOUTSUBDOMAIN	Customizable Checkout Subdomains
CLASSES	Classes
COMMACCTAPP	Accounting Approval of Commissions
COMMERCECATEGORIES	Commerce Categories
COMMISSIONONCUSTOMFIELDS	Commission on Custom Fields
COMMISSIONS	Employee Commissions
COMPENSATIONTRACKING	Compensation Tracking
CONSOLPAYMENTS	Consolidated Payments
CREATESUITEBUNDLES	Create bundles with SuiteBundler
CRM	Customer Relationship Management
CRMTIME	Time Tracking for CRM
CRM_TEMPLATE_CATEGORIES	CRM Template Categories
CUSTOMCODE	Client SuiteScript
CUSTOMERACCESS	Customer Access
CUSTOMGLLINES	Custom GL Lines
CUSTOMRECORDS	Custom Records
CUSTOMSEGMENTS	Custom Segments
CUSTOMTRANSACTIONS	Custom Transactions

Feature Internal ID	Feature Name
DEPARTMENTS	Departments
DIRECTDEPOSIT	Direct Deposit
DISTRIBUTIONRESOURCEPLANNING	Distribution Resource Planning
DOCUMENTPUBLISHING	Document Publishing
DOCUMENTS	File Cabinet
DOCUMENTSANTIVIRUS	Antivirus Scanning on File Cabinet Files
DOWNLOADITEMS	Sell Downloadable Files
DROPSHIPMENTS	Drop Shipments & Special Orders
DUPLICATES	Duplicate Detection & Merge
DYNALLOCATION	Dynamic Allocation
EBAY	eBay Integration
EFFECTIVEDATING	Effective Dating
EFT	Electronic Funds Transfer
EMAILINTEGRATION	Capture Email Replies
EMPPERMS	Global Permissions
ESCALATIONRULES	Automated Case Escalation
ESTIMATES	Estimates
EXPENSEALLOCATION	Expense Allocation
EXPREPORTS	Expense Reports
EXTCRM	Online Forms
EXTREMELIST	Inline Editing
EXTSTORE	External Catalog Site (WSDK)
FULFILLMENTREQUEST	Fulfillment Request
FXRATEUPDATES	Currency Exchange Rate Integration
GAINLOSSACCTMAPPING	Foreign currency variance mapping
GIFTCERTIFICATES	Gift Certificates
GLAUDITNUMBERING	GL Audit Numbering
GRIDORDERMANAGEMENT	Grid Order Management
GROSSPROFIT	Gross Profit
GROUPAVERAGECOSTING	Group Average Costing
HELPDESK	Help Desk
HISTORICALMETRICS	Historical Metrics
I18NTAXREPORTS	International Tax Reports
INBOUNDCASEEMAIL	Email Case Capture
INTERCOMPANYAUTODROPSHIP	Automated Intercompany Drop Ship

Feature Internal ID	Feature Name
INTERCOMPANYAUTOELIMINATION	Automated Intercompany Management
INTERCOMPANYTIMEEXPENSE	Intercompany Time and Expense
INTRANET	Intranet
INVENTORY	Inventory
INVENTORYCOUNT	Inventory Count
IPADDRESSRULES	IP Address Rules
ISSUEDB	Issue Management
ITEMDEMANDPLANNING	Demand Planning
ITEMOPTIONS	Item Options
JOBCOSTING	Job Costing and Project Budgeting
JOBREQUISITION	Job Requisitions
JOBS	Projects
KNOWLEDGEBASE	Knowledge Base
KPIREPORTS	KPI Scorecards
KUDOS	Kudos
LANDEDCOST	Landed Cost
LEADMANAGEMENT	Lead Conversion
LOCATIONS	Locations
LOTNUMBEREDINVENTORY	Lot Tracking
MAILMERGE	Mail Merge
MARKETING	Marketing Automation
MATRIXITEMS	Matrix Items
MFGROUTING	Manufacturing Routing and Work Center
MFGWORKINPROCESS	Manufacturing Work In Process
MOBILEPUSHNTF	Mobile Push Notification
MOSS	EU Mini One Stop Shop
MULTIBOOKMULTICURR	Multi-Book Accounting and Multiple Currencies
MULTICURRENCY	Multiple Currencies
MULTICURRENCYCUSTOMER	Multi-Currency Customers
MULTICURRENCYVENDOR	Multi-Currency Vendors
MULTILANGUAGE	Multi-Language
MULTILOCINVT	Multi-Location Inventory
MULTIPARTNER	Multi-Partner Management
MULTIPLEBUDGETS	Multiple Budgets
MULTIPLECALENDARS	Multiple Calendars

Feature Internal ID	Feature Name
MULTISHIPTO	Multiple Shipping Routes
MULTISITE	Multiple Web Sites
MULTIVENDOR	Multiple Vendors
MULTPRICE	Multiple Prices
NETSUITEAPPROVALSWORKFLOW	NetSuite Approvals Workflow
NOTALTSALESAMOUNT	Not ALT_SALES Amount
NOTMULTIPARTNER	Not Multipartner
NOTTEAMSELLING	Not Team Selling
ONLINEORDERING	Online Ordering
OPENIDSSO	OpenID Single Sign-on
OPPORTUNITIES	Opportunities
OUTLOOKINTEGRATION_V3	Outlook Integration
PARTNERACCESS	Partner Access
PARTNERCOMMISSIONS	Partner Commissions/Royalties
PAYABLES	A/P
PAYCHECKJOURNAL	Paycheck Journal
PAYPALINTEGRATION	PayPal Integration
PAYROLL	Payroll
PICKPACKSHIP	Pick, Pack and Ship
PRM	Partner Relationship Management
PROJECTTASKMANAGER	Project Task Manager
PROMOCODES	Promotion Codes
PURCHASECARDDATA	Send Purchase Card Data
PURCHASECONTRACTS	Purchase Contracts
PURCHASEORDERS	Purchase Orders
PURCHASEREQS	Purchase Requests
QUANTITYPRICING	Quantity Pricing
RECEIVABLES	A/R
REQUISITIONS	Requisitions
RESOURCEALLOCATIONAPPROVAL	Resource Allocation Approval Workflow
RESOURCEALLOCATIONCHART	Resource Allocation Chart
RESOURCEALLOCATIONS	Resource Allocations
RESOURCESKILLSETS	Resource Skill Sets
RETURNAUTHS	Return Authorizations
REVENUECOMMITMENTS	Revenue Commitments

Feature Internal ID	Feature Name
REVENUECONTRACTS	Revenue Contracts
REVENUEEXPENSEMANAGEMENT	Revenue and Expense Management
REVENUERECOGNITION	Revenue Recognition
REVRECSALESORDERFORECASTING	Sales Order Revenue Forecasting
REVRECVSOE	VSOE
RFQ	Request For Quote
RUM	Real User Monitoring
SALESCAMPAIGNS	Sales Campaigns
SALESORDERS	Sales Orders
SAMLSSO	SAML Single Sign-on
SERIALIZEDINVENTORY	Serialized Inventory
SERVERSIDESCRIPTING	Server SuiteScript
SERVICEPRINTEDCHECKS	Service Printed Checks and Stubs
SFA	Sales Force Automation
SFA_AND_NOTASA	SFA And Not ALT_SALES Amount
SHIPPINGLABELS	Shipping Label Integration
SITELOCATIONALIASES	Descriptive URLs
SOFTDESCRIPTORS	Credit Card Soft Descriptors
STACKABLEPROMOTIONS	SuitePromotions
STANDARDCOSTING	Standard Costing
STATACCOUNTING	Statistical Accounts
STOREPICKUP	Store Pickup
SUBSCRIPTIONBILLING	Subscription Billing
SUITEANALYTICSCONNECT	SuiteAnalytics Connect
SUITECOMMERCEENTERPRISE	SuiteCommerce Advanced
SUITESIGNON	SuiteSignOn
SUITESOCIAL	SuiteSocial
SUPPLTAXCALC	Supplementary Tax Calculation
SUPPLYCHAINMANAGEMENT	Supply Chain Management
SUPPORT	Customer Support and Service
TABLEAU	Tableau Workbook Export
TAXAUDITFILES	Tax Audit Files
TBA	Token-based Authentication
TEAMSELLING	Team Selling
TELEPHONY	Telephony Integration

Feature Internal ID	Feature Name
TIMETRACKING	Time Tracking
TRANDELETIONREASONCODE	Use Deletion Reason
UNITSOFMEASURE	Multiple Units of Measure
UPSELL	Upsell Manager
URLCOMPONENTALIASES	URL Component Aliases
VENDORACCESS	Vendor Access
VENDORRETURNAUTHS	Vendor Return Authorizations
WARRANTYANDREPAIRSMANAGEMENT	Warranty and Repairs Management
WEBAPPLICATIONS	SuiteScript Server Pages
WEBAPPLICATIONVERSIONING	SuiteScript Server Pages Versioning
WEBHOSTING	Host HTML Files
WEBSERVICESEXTERNAL	Web Services
WEBSITE	Web Site
WEBSTORE	Web Store
WEEKLYTIMESHEETS	Weekly Timesheets
WITHHOLDINGTAX	Withholding Tax
WORKFLOW	SuiteFlow
WORKORDERS	Work Orders

Preference Names and IDs

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The following tables list the internal IDs for all NetSuite preference configuration pages that support SuiteScript.

The way you interact with this page varies depending on your SuiteScript version:

- In SuiteScript 1.0, load the page using nlapiLoadConfiguration(type). After the page loads, you can then get or set configuration values using the methods on the nlobjConfiguration object.
- In SuiteScript 2.0, load the page using config.load(options). After the page loads, you can get or set configuration values using the returned record.Record object. Additionally, you can use User.getPreference(options) to get the values for **General Preferences** and **Accounting Preferences**.

To set values on any of these pages in SuiteScript 1.0, you must first load the page using nlapiLoadConfiguration(type). After the page loads, you can then get or set all configuration values using the methods on the nlobjConfiguration object.

To set values on any of these pages in SuiteScript 2.0, you must first load the page using config.load(options). After the page loads, you can get or set all configuration values using the returned record.Record object. Additionally, you can use User.getPreference(options) to get the values for **General Preferences** and **Accounting Preferences**.

NetSuite configuration preference IDs are grouped into the following categories:

- General Preferences
- Company Information
- User Preferences
- Accounting Preferences
- Accounting Periods
- Tax Setup
- Tax Periods

General Preferences

These are the account preferences that can be found by going to Setup > Company > General Preferences.

The internal ID for the General Preferences page is **companypreferences**. All preference internal IDs are case-insensitive.

Preference UI Label	Preference Internal ID
Date Format	DATEFORMAT
Long Date Format	LONGDATEFORMAT
Time Format	TIMEFORMAT
Number Format	NUMBERFORMAT
Negative Number Format	NEGATIVE_NUMBER_FORMAT
Phone Number Format	PHONEFORMAT
First Day of Week	FIRSTDAYOFWEEK
Search Sorting	SEARCHSORTING
Add Primary Contact to Bill To Address	CONTACTONBILLTO
Use Last Name First for Employees	LASTNAMEFIRST
Use Last Name First for Entities	LASTNAMEFIRSTENTITIES
Pre-Populate Contact Address	PREPOPULATECONTACTADDRESS
Show Employees as Contacts	SHOWEMPLOYEESASCONTACTS
Show Display Name with Item Codes	ITEMNUMBERING
Password Policy	PASSWORD_POLICY
Minimum Password Length	MINPASSWORDLENGTH
Password Expiration in Days	PASSWORDEXPIREDAYS
Idle Session Timeout in Minutes	IDLE_SESSION_TIMEOUT
Internal Web Site	INTERNALWEBSITE
Allow Free-Form States in Addresses	FREEFORMSTATES
Use State Abbreviations in Addresses	ABBREVIATESTATES
Company Logo Folder	COMPANYLOGOFOLDER
Default Role for New Customers	CUSTOMERROLE

Preference UI Label	Preference Internal ID
Assign Tasks to Partners	ASSIGNTASKSTOPARTNERS
Auto Name Customers	AUTONAMECUSTOMERS
Calendar System	CALENDARSYSTEM
Default Customer Type	CUSTOMERTYPE
Customer Center Welcome Message	CUSTOMERWELCOMEMESSAGE
Show Help Link In Customer Center	CUSTOMERSHOWHELPLINK
Horizontal Labels	HORIZONTALLABEL
Screen Font	FONT
Landing Page	CUSTOMLANDINGPAGE
Number of rows in List segments	LISTSEGMENTSIZE
Maximum entries in Dropdown	MAXDROPDOWNSIZE
Log System Notes on Update Only	LOGSYSTEMNOTESONUPDATEONLY
Show Quick Add Row on Lists	SHOWQUICKADD
Show List When Only One Results	SHOWLISTONRESULT
Default Customer Type	CUSTOMERTYPE
Default Lead Type	DEFAULTLEADTYPE
Default Vendor Type	DFLTVENDORTYPE
Default Partner Type	DFLTPARTNERTYPE
Auto Name Customers	AUTONAMECUSTOMERS
Auto Inactivate Contacts with Customers	INACTIVATE_CONTACTS
Show Individuals as Contacts	SHOWINDIVIDUALSASCONTACTS
Hide Attachment Folders	HIDEATTACHMENTFOLDERS
Assign Tasks to Partners	ASSIGNTASKSTOPARTNERS
Email Employee on Approvals	EMAILEMPLOYEEONAPPROVAL
Maintenance Complete Email Notification	EMAILMAINTENANCECOMPLETE
Show Reports in Grid	REPORTGRID
Collapse VSOE Column By Default On Sales Transactions	COLLAPSEVSOEFIELDSET
Time Selectors Use Fiscal Calendars Based on First Month	TIMESELECTORSUSEFIRSTFISCALMONTH
Web Site Hosting Files Always Available	HOSTING_FILES_PUBLIC
Show Transaction Numbering Setup	SHOW_TRAN_NUMBERING_SETUP
Asynchronous Job Plan Recalculation	SHOW_TRAN_NUMBERING_SETUP
Preferred Subcustomer Form	SUBCUSTOMERFORM

Company Information

The Company Information preferences are available at Setup > Company > Company Information.

To interact with these preferences in SuiteScript 2.0, use the COMPANY_INFORMATION value from the config.Type enum. In SuiteScript 1.0, use the internal ID **companyinformation**. All Company Information preference IDs are case-sensitive.

For details about working with the Company Information page in the UI, see the help topic Configuring Company Information.

Note: The Company Information page includes several preferences that are displayed as body fields. These fields are described in the following table.

Preference Label	Preference Internal ID
Company Name	companyname
Legal Name	legalname
Company Logo (Forms)	formlogo
Company Logo (Pages)	pagelogo
Display Logo Internally	displaylogointernally
Web Site	url
County/State/Province	state
Country	country
Return Email Address	email
Fax	fax
Currency	basecurrency
Employer Identification Number (EIN)	employerid
SSN or TIN (Social Security Number, Tax ID Number)	taxid
First Fiscal Month	fiscalmonth
Time Zone	timezone
Account ID	companyid
Customer Center Login	customersurl
These three IDs represent text summaries of the data that exists on the three address subrecords.	mainaddress _text
	shippingaddress _text
	returnaddress _text

The Company Information page also includes gray boxes with the headings Address, Shipping Address, and Return Address. In the UI, you interact with these components by clicking the Edit link next to each box. Clicking any of these links displays a popup window that includes more fields. The way you interact with these fields programmatically varies depending on whether you are using SuiteScript 1.0 or 2.0, as follows.

SuiteScript 2.0

For a script written in SuiteScript 2.0, you must use subrecord methods to access the fields associated with the Address, Shipping Address, and Return Address blocks. You must first instantiate the

appropriate subrecord by using Record.getSubrecord(options). This method requires one argument: the ID of the subrecord. For the address subrecords on the Company Information page, use the following internal IDs:

- mainaddress
- shipaddress
- returnaddress

After you instantiate the subrecord, you can interact with any of the preferences shown in the following table. These preferences are handled as fields on each subrecord instance.

Preference Label	Preference Internal ID
Address1	addr1
Address2	addr2
Attention	attention
Addressee	addressee
Phone	addrphone
City	city
County/State/Province	state
Zip	zip
Country	country

For a 2.0 script sample, see the help topic Example: Retrieving a Body Field Address Subrecord. For general details on subrecords, see the help topic SuiteScript 2.0 Scripting Subrecords.

SuiteScript 1.0

For a script written in SuiteScript 1.0, use the following preference IDs for the individual details displayed in the Address, Shipping Address, and Return Address blocks. You can script with these fields as if they were additional body fields on the page.

Preference Label	Preference Internal ID
For the main address:	
Address1	address1
Address2	address2
Ship to Attention	attention
City	city
County/State/Province	state
Zip	zip
Country	country
Address	addresstext
For the shipping address:	
Address1	shippingaddress1
Address2	shippingaddress2

Preference Label	Preference Internal ID
City	shippingcity
County/State/Province	shippingstate
Zip	shippingzip
Country	shippingcountry
Shipping address	shipaddresstext
For the return address:	
Address1	returnaddress1
Address2	returnaddress2
City	returncity
County/State/Province	returnstate
Zip	returnzip
Country	returncountry
Return address	returnaddresstext

User Preferences

These are the user preferences that can be found by going to Home > Set Preferences.

The internal ID for the user preferences page (which appears as the Set Preferences page in the UI) is **userpreferences**. All preference internal IDs are case-insensitive.

Be aware that the API for setting user preferences works the same as the UI in terms of setting a preference permanently or just for a user's session. In the UI if a user sets a preference, and the preference reverts back to a default setting on the user's next login, the same behavior is supported in SuiteScript.

Preference UI Label	Preference Internal ID
On the General tab	
Nickname	MESSAGE_NICKNAME
Signature	MESSAGE_SIGNATURE
Add Signature to Messages	MESSAGE_AUTOSIGNATURE
From Email Address	MESSAGE_EMAIL
Language	LANGUAGE
Search Sorting	SEARCHSORTING
Language of the Help Center	HELP_LANGUAGE
PDF Language	PDFLANGUAGE
Time Zone	TIMEZONE
First Day of the Week	FIRSTDAYOFWEEK
Date Format	DATEFORMAT
Long Date Format	LONGDATEFORMAT

Preference UI Label	Preference Internal ID
Time Format	TIMEFORMAT
Number Format	NUMBERFORMAT
Negative Number Format	NEGATIVE_NUMBER_FORMAT
Phone Number Format	PHONEFORMAT
Auto Place Decimal	AUTOPLACE
CSV Column Delimiter	CSV_COLUMN_DELIMITER
CSV Decimal Delimiter	CSV_DECIMAL_DELIMITER
Use Multicurrency Expense Reports	USE_MC_ON_EXPREPT
Download PDF Files	DOWNLOADPDFS
Address Mapping Type	MAPTYPE
Show Internal IDs	EXPOSEIDS
Only Show Last Subaccount	ONLYSHOWLASTSUBACCT
Only Show Last Subentity	ONLYSHOWLASTSUBENT
Only Show Last Subitem	ONLYSHOWLASTSUBITEM
Submit Warnings	SUBMITWARNINGS
Limit CC Field to Contacts & Employees	EMAILLIMITCC
Default Issue Email Notification	ISSUE_EMAIL_ME_WHEN
Notify Me Upon Issue Assignment	ISSUE_NOTIFY_UPON_ASSIGNMENT
Delay Loading of Sublists	DELAYLOADINGSUBLISTS
Number of Rows in List Segments	LISTSEGMENTSIZE
Maximum Entries in Dropdowns	MAXDROPDOWNSIZE
Type-Ahead on List Fields	TYPEAHEADSELECTS
Require Exact Match on Item Type-Ahead	ITEMEXACTMATCH
Show Quick Add Row on Lists	SHOWQUICKADD
Display Bounce Warning on Campaigns	CAMPAIGN_BOUNCE_WARNING
Prefer Native Select Fields Over NS Dropdowns in Internet Explorer	NATIVE_DROPDOWNS
Show App ID Field	SHOW_APPID_FIELD
Show ID Field on Sublists	SHOW_ID_FIELD
On the Appearance tab	
Color Theme	COLORTHEME
Screen Font	FONT
Compensate for Large Fonts	SYSTEMLARGEFONTS
Density Setting for Internet Explorer	MSIE_ZOOM_FACTOR
Register Look on Lists	REGISTERSTYLE

Preference UI Label	Preference Internal ID
Only Show Field Boarders on Hover	SHOWFIELDBORDERONHOVER
Chart Theme	CHART_THEME
Chart Background	CHART_BACKGROUND
Landing Page	LANDINGPAGE
Show Portlet Hint	SHOWPORTLETHINT
Set Customer Dashboard As Default View On Customer Record	DASHBOARD_DEFAULT_VIEW_FOR_CUSTOMER
Limit Entry Forms to Two Columns	LIMITTOTWOCOLUMNS
Expand Tabs on Entry Forms	UNLAYEREDTABS
Enable Rich Text Editing	RICHTEXTEDITOR
Default Rich Text Editor Font	EDITORFONT
Default Rich Text Editor Font Size	EDITORFONTSIZE
Display Default Them With Optimal Color Contrast	ACCESSIBILITY_HIGH_CONTRAST
On the Transactions tab	
Auto Fill Transactions	AUTOFILL
Alphabetize Items Regardless of Type	ALPHABETIZE_ITEMS
Duplicate Number Warnings	DUPLICATEWARNINGS
Inventory Level Warnings	STOCKWARNINGS
Customer Credit Limit Handling	CUSTCREDLIMHANDLING
Vendor Credit Limit Warnings	VENDCREDLIMWARNINGS
Print Using HTML	HTMLPRINTING
Transaction Email Attachment Format	TRANSACTION_ATTACHMENT_FORMAT
Horizontal Print Offset	HORZPRINTOFFSET
Vertical Print Offset	VERTPRINTOFFSET
On the Analytics tab	
Report by Period	REPORTBYPERIOD
Show Reports in Grid	REPORTGRID
Customize Font on Financial Reports	ENABLE_REPORT_FONT_CUSTOMIZATION
Print Company Logo	DISPLAYLOGO
Display Report Title on Screen	DISPLAYRPTTITLE
Display Report Description	DISPLAYRPTDESC
Calculate Forecasts as Weighted	FORECASTWEIGHTED
Default Bank Account	DEFAULT_BANKREG
Show Forecasts as Weighted	FORECASTWEIGHTED
Show List When Only One Result	SHOWLISTONERESULT

Preference UI Label	Preference Internal ID
Quick Search Uses Keywords	KEYWORDSEARCH
Popup Search Uses Keywords	KEYWORDSEARCHPOPUP
Include Inactives in Global & Quick Search	SEARCHINACTIVES
Popup Auto Suggest	POPUPAUTOSUGGEST
Global Search Auto Suggest	SEARCHAUTOSUGGEST
Global Search Sort by Name/ID	GLOBALSEARCHSORTBYNAME
Global Search Customer Prefix Includes Leads and Prospects	GLOBALSEARCHCUPREFIX
PDF Page Orientation	REPORTPDFORIENTATION
PDF Font Size	REPORTPDFFONTSIZE
CSV Export Character Encoding	CSVEXPORTENCODING
KPI Export Character Encoding	KPI_PERIOD_SPECIFIC_RATES
On the Activities tab	
Edit Activities from Calendar	EVENT_EDITFROMCALENDAR
Send Invitation Emails	EVENT_EMAILNOTIFICATION
Restrict Invitees to Employees	EVENT_INTERNALINVITEESONLY
Default Event Access Setting for New Events	EVENT_DEFAULTPUBLIC
Default Reminder Type	REMINDERTYPE
Default Reminder Time	REMINDERPERIOD
Play Audio with Popup Event Reminders	REMINDERPLAYWAVE
Default Priority for Tasks	DEFAULTTASKPRIORITY
Default New Tasks Public	TASK_DEFAULTPUBLIC
Default New Phone Calls Public	CALL_DEFAULTPUBLIC
Default Sync Category	DEFAULT_CONTACT_SYNC_CATEGORY
On the Alerts tab	
First Selection	EMAILALERT_AM
Second Selection	EMAILALERT_NOON
Third Selection	EMAILALERT_PM
Include links in HTML alerts	LINKS_EMAILALERT
Respect Quick Date Portlet Settings	USE_QUICKDATE_IN_ALERTS
E-Mail	EMAILALERT_EMAIL
Send an On-Demand Alert from this Role	ALERTONDEMAND
On the Restrict View tab	
Subsidiary	SUBSIDIARY
Include Sub-Subsidiaries	SUBSIDIARYSUBS

Preference UI Label	Preference Internal ID
Department	DEPARTMENT
Include Sub-Departments	DEPARTMENTSUBS
Include Unassigned	DEPARTMENTUNASSIGNED
Location	LOCATION
Include Sub-Locations	LOCATIONSUBS
Include Unassigned	LOCATIONUNASSIGNED
Class	CLASS
Include Sub-Classes	CLASSSUBS
Include Unassigned	CLASSUNASSIGNED
On the Telephony tab	
Telephony Option	TELEPHONY_OPTION
TAPI Device	TELEPHONYDEVICE
CTI URL	CTI_URL
Prefix to Dial Out	DIALOUTPREFIX

Accounting Preferences

These are the account preferences that can be found by going to Setup > Accounting > Accounting Preferences. All preference internal IDs are case-insensitive.

The internal ID for the Accounting Preferences page is **accountingpreferences**.

Preference UI Label	Preference Internal ID
On the General tab	
Use Account Numbers	ACCOUNTNUMBERS
Use Legal Name In Account	ACCOUNTLEGALNAME
Show All Transaction Types In Reconciliation	RECONCILIATIONALLTRANTYPES
Expand Account Lists	ASSETCOGSITEMACCTS
Cash Basis Reporting	CASHBASIS
Aging Reports Use	AGEFROM
Void Transactions Using Reversing Journals	REVERSALVOIDING
Set Reversal Variance Date Equal To The Reversing Journal Date When Voided Transaction Is In A Closed Period	SETREVERSINGVARIANCEDATETOREVJES
Use Journal Entry Summarization On Intercompany Elimination	ELIMINATION_JE_SUMMARIZATION
Require Approvals On Journal Entries	JOURNALAPPROVALS

Preference UI Label	Preference Internal ID
Allow User Events On Bulk Journal Approval	BULK_JOURNAL_APPROVAL_EVENTS
Allow GL Custom Segment Deletion	CUSTOM_SEGMENT_DELETION
Enable Accounting Period Window	ALLOWFUTUREPERIODLOCK
Minimum Period Window Size	OPENCURFUTUREPERIODWINDOW
Allow Transaction Date Outside Of Posting Period	DATEPERIODMISMATCH
GL Audit Numbering Method	GLNUMBERINGBY
Default Posting Period When Transaction Date In Closed Period	DEFAULTPERIODIFCLOSED
Create and Edit Inventory Transactions Dated In Closed Periods	INVT_TRANS_CLOSED_PERIODS
Allow Quick Close Of Accounting Periods	ALLOW_PERIODS_QUICK_CLOSE
Apply Payments Through Top-Level Customers Only	PAYMENTSONLYFROMTOPPARENT
Show Only Open Transactions On Statements	OPENONLYSTMTS
Open Transactions On Statements	OPENONLYTRANSACTIONS
Customer Credit Limit Handling	CUSTCREDLIMHANDLING
Customer Credit Limit Includes Orders	CUSTCREDLIMORDERS
Days Overdue For Warning / Hold	CREDLIMDAYS
Include Tax For Term Discounts	TERMDISCOUNTSINCLUDETAX
Include Shipping For Term Discounts	TERMDISCOUNTSINCLUDESHIPPING
Default Vendor Payments To Be Printed	VENDPYMTTOPRINT
Vendor Credit Limit Warnings	VENDCREDLIMWARNINGS
Vendor Credit Limit Includes Orders	VENDCREDLIMORDERS
Make Departments Mandatory	DEPTMANDATORY
Make Classes Mandatory	CLASSMANDATORY
Make Locations Mandatory	LOCMANDATORY
Allow Per-Line Departments	DEPTSPERLINE
Allow Per-Line Locations	LOCSPERLINE
Allow Per-Line Classes	CLASSESPERLINE
Always Allow Per-Line Classifications On Journals	CDLPERLINEONJE
Allow Non-Balancing Classifications On Journals	NONBALANCINGCDLONJE
Allow Empty Classifications On Journals	NULLCDLONJE
Allow Users to Modify Revenue Recognition Schedule	MODIFYREVRECTOTALAMOUNT
Prorate Revenue Recognition Dates For Partially Billed Sales Orders	PRORATEREVRECINVFROMSO
Create Revenue Journals In GL	REVRECJOURNALENTRYSUMMARIZATION

Preference UI Label	Preference Internal ID
Default Revenue Recognition Journal Date To	REVRECJOURNALDATEDEFAULT
Allow Users To Modify VSOE Values on Transactions	MODIFYVSOEVALSONTRAN
Use System Percentage Of Completion For Schedules	USESYSCALCPCT4REVREC
Adv Billing: Use Sales Order Amount	CALCPCTCOMPFROMSALESORDERAMT
Allow Revenue Commitments In Advance Of Fulfillment	UNSHIPPEDREVENUECOMMITMENTS
Allow Revenue Commitment Reversals In Advance Of Item Receipt	UNRECEIVEDREVENUECOMMITMENTS
Default Deferred Revenue Reclassification Account	DEFAULTDEFERREDREVENUEACCOUNT
Default Foreign Currency Adjustment Revenue Account	DEFAULTFXREVENUEACCOUNT
Allow Users To Modify Amortization Schedule	MODIFYAMORTOTALAMOUNT
Default Amortization Journal Date To	AMORJOURNALDATEDEFAULT
Default Amortization Journal Entry Form	AMORTIZATIONJOURNALENTRYFORM
Intercompany Time	INTERCOMPANYTIME
Intercompany Expenses	INTERCOMPANYEXPENSE
Intercompany Expenses	INTERCOMPANYEXPENSE
Enable Budget With Elimination Subsidiaries	BUDGETINCLUDEELIMSUB
Require Approvals on Journal Entries	JOURNALAPPROVALS
Rate Provider	EXCHANGE_RATE_PROVIDER
Use Triangulation Calculation By NetSuite	EXCHANGE_RATE_PROVIDER
Maximum number of MLI locations	MAXLOCATIONS
On the Items/Transactions tab	
Purchase Discount Account	PURCHDISCACCT
External Inventory In Transit Account	EXTINTRINVTACCT
Sales Discount Account	SALESDISCACCT
Default Expense Account	EXPENSEACCOUNT
Default Income Account	INCOMEACCOUNT
Default Receivables Account	ARACCOUNT
Default Cogs Account	COGSACCOUNT
Default Asset Account	ASSETACCOUNT
Default Payment Account	PAYMENTACCOUNT
Default Gain / Loss Account	GAINLOSSACCOUNT
Default Bill Quantity Variance Account	BILLPRICEVARIANCEACCT

Preference UI Label	Preference Internal ID
Default Unbuild Variance Account	UNBUILDVARIANCEACCOUNT
Default Vendor Return Variance Account	VENDRETURNVARIANCEACCOUNT
Default Customer Return Variance Account	CUSTOMERRETURNVARIANCEACCOUNT
Default Production Quantity Variance Account	PRODQTYVARIANCEACCT
Default Production Price Variance Account	PRODPRICEVARIANCEACCT
Default Purchase Price Variance Account	PURCHASEPRICEVARIANCEACCT
Default Inventory Count Account	INVCOUNTACCOUNT
Default WIP Cost Variance Account	WIPVARIANCEACCT
Anyone Can Set Item Accounts	EDITITEMACCOUNTS
Default Dropship Expense Account	DROPSHIPEXPENSEACCOUNT
Default Scrap Account	SCRAPACCT
Default WIP Account	WIPACCT
Consolidate Jobs on Sales Transactions	CONSOLINVOICES
Maximum # of Quantity-based Price Levels	QTYPRICECOUNT
Allow Quantity Discounts per Price Level on Schedules	QTYPRICESCHEDULEMULTDISCOUNTS
Include Reimbursements in Sales and Forecast Reports	FORECASTINCLUDES_REIMB_EXP
Include Shipping in Sales and Forecast Reports	FORECASTINCLUDES_SHIPPING
Transaction Types to Exclude from Forecast Reports	FORECASTTRANTYPES
Transaction Types to Exclude from Sales Reports	SALESTRANTYPES
Default Fixed Date Charge Rule Stage	FFCRFIXEDDATE
Default Milestone Charge Rule Stage	FFCRMILESTONE
Default Project Progress Charge Rule Stage	FFCRPROGRESS
Default Time-Based Rule Stage	TIMECR
Scan Individual Items	SINGLEITEMBARCODING
Centralize Purchasing In A Single Location	CENTRALIZEDPURCHASING
Days Before Lot Expiration Warning	LOTEXPIRATIONWARNING
Require Bins on All Transactions Except Item Receipts	REQUIREBINSONTRANS
Use Preferred Bin on Item Receipts	USEPREFERREDBINONITEMRCPT
Include Landed Cost In Last Purchase Price	INCLUDELANDEDCOSTINLPP
Inventory Costing Method	INVTCOSTMETHOD
Default Cost Category	DEFAULTCOSTCATEGORY
Average Cost Completion Unit Cost	AVGCOSTCOMPLETIONCOSTINGMETHOD

Preference UI Label	Preference Internal ID
Use Intransit Value In Group Average Cost Calculations	USEINTRANSITINGACCALCULATIONS
Customers Can Pay Online	EXTERNALPAYMENTS
Get Authorization On Customer Center Sales Orders	AUTHORIZECUSTOMERCENTERORDERS
Use Card Security Code For Credit Card Transactions	CCSECURITYCODE
Allow Adjusted Expiration Date To Improve Recurring Payments	ALLOWADJUSTEDEXPIRATIONDATE
Enable "Sale" Payment Operations On A Sales Order By Automatically Creating A Customer Deposit	CREATE_DEPOSIT_ON_SALES_ORDER_SALE
Use Strict Rules For The Selection Of Payment Processing Profiles	STRICT_SELECTION_OF_PPP
Preserve Transactions When Payment Is On Hold	PRESERVE_TRAN_ON_HOLD_PAYMENT
Duplicate Number Warnings	DUPLICATEWARNINGS
Sort Reconcile By	RECONSORTCOL
Recalculate Estimated Cost on Creation of Linked Transactions	USELATESTCOSTESTIMATE
Matrix Item Name/Number Separator	SKUSEPARATOR
Gift Certificate Auth Code Generation	GIFTCERTAUTHCODEGENERATION
Enforce Minimum Quantity On Return Authorizations	ENFORCE_MIN_QUANTITY_RET_AUTH
On the Order Management tab	
Default Sales Order Status	DEFSALESORDSTATUS
Require Re-approval on Edit of Sales Order	REAPPROVESOONEDIT
Send Email Confirmation when Sales Order Canceled	EMAILCANCELORDER
Default Location for Sales Orders	DEFAULTSALESORDERLOCATION
Default Commit Option On Sales Order	DEFAULTSALESORDERCOMMITOPTION
Default Commit Option On Work Order	DEFAULTWORKORDERCOMMITOPTION
Default Commit Option On Transfer Order	DEFAULTTRANSFERORDERCOMMITOPTION
Item Commitment Transaction Ordering	ITEMCOMMITMENTTRANSACTIONORDER
Perform Item Commitment After Transaction Entry	AUTOMATICITEMCOMMITMENT
Always Print Kit Items on Picking Tickets	PICKINGTICKETKITITEMS
Show Non-Inventory Items on Picking Tickets and Packing Slips	PICKINGTICKETNONINVT
Show Uncommitted Items on Picking Tickets	PICKINGTICKETUNCOMMITTED
Name for Packed Status	NAMINGPACKED

Preference UI Label	Preference Internal ID
Name for Picked Status	NAMINGPICKED
Name for Shipped Status	NAMINGSHIPPED
Show Additional Items on Packing Slips	SHOWADDLITEMSPACKSLIP
Show Drop Ship Items on Packing Slips	PACKINGSLIPDROPSHIP
Limit Status on Packing Slip Queue	PACKINGSLIPSTATUS
Fulfill Based on Commitment	FULFILLCOMMITTED
Default Items to Zero Received/Fulfilled	DEFAULTUNFULFILLED
Allow Overage on Assembly Builds	OVERBUILDS
Filter Bulk Fulfillment Page by Location	BULKFULFILLOCFILTERING
Send Order Fulfilled Confirmation Emails	ORDFULFILLCONFEMAIL
Use Web Site Template for Fulfillment Emails	ORDFULFILLUSESTORETEMPLATES
Manufacturing Issue Based On Commitment	BUILDCOMMITTED
Update Transaction Date Upon Fulfillment Status Change	UPDATEITEMSHIPDATEONSTATUSCHANGE
Show Unfulfilled Items on Invoices	SHOWUNSHIPPEDITEMS
Invoice In Advance Of Fulfillment	UNSHIPPEDINVOICES
Convert Absolute Discounts to Percentage When Billing	CONVERTABSOLUTEDISCOUNTS
Base Invoice Date on Billing Schedule Date	INCOICEUSESCHEDULEDATE
Drop Ship P.O. Form	DROPSHIPTEMPLATE
Automatically Email Drop Ship P.O.s	EMAILDROPSHIPPOS
Queue Drop Ship P.O.s for Printing	PRINTDROPSHIPPOS
Automatically Fax Drop Ship P.O.s	FAXDROPSHIPPOS
Include Committed Quantities	DROPSHIPINCLUDECOMMITTED
Limit Vendor List on Items	LIMITITEMVENDORS
Update Drop Ship Order Quantities Automatically Prior To Shipment	KEEPDROPSHIPQUANTITIESINSYNC
Drop Ship FulFillment Quantity Validation	FULFILLDROPSHIPORDERFROMINVENTORY
Allow Both Mark Shipped Fulfillments And Receipts On A Drop Shipment Line	DROPSHIPANDRECEIVEINTOINVDROPSHIPPO
Update Special Order Quantities Automatically Prior To Shipment	KEEPSPECIALORDERQUANTITIESINSYNC
Allow Expenses On Purchases	POEXPENSES
Default Location for Purchase Orders	DEFAULTPURCHASEORDERLOCATION
Maximum {Purchase} Lines To Consolidate	MAXPURCHASES
Allow Default Email On Purchase Orders Using SuiteFlow Approval Routing	ALLOW_DEFAULT_EMAIL_ON_PO

Preference UI Label	Preference Internal ID
Default Number Of Requests For Quote Pricing Tiers	DEFAULT_RFQ_TIERS
Bill in Advance of Receipt	UNCRECEIVEDBILLS
Allow Overage on Item Receipts	OVERRECEIPTS
Default Receiving Exchange Rate	DEFAULTRECEIVINGEXCHANGERATE
Use Purchase Order Rate On Bills	USEPORATEONBILLS
Landed Cost Allocation Per Line	LANDEDCOSTPERLINEDEFAULT
Default Return Auth. Status	DEFRTNAUTHSTATUS
Refund in Advance of Return	UNCRECEIVEDRTNAUTHS
Restock Returned Items	RESTOCKRETURNS
Write-Off Account for Returns	WRITEOFFACCOUNT
Default Vendor Return Auth. Status	DEFVENDAUTHSTATUS
Credit in Advance of Vendor Return	UNRETURNEDVENDAUTHS
Default Transfer Order Status	DEFTRNFRORDSTATUS
Generate Transfer Orders In Supply Planning	GENERATETRNFRORDINPLANNING
Use Item Cost As Transfer Cost	ITEMCOSTASTRNFRORDCOST
Default Transfer Order Incoterms	DEFAULT_TRNFRORD_INCOTERM
Default Vendor Bill Status	DEFVENDBILLSTATUS
Allow Purchase Of Assembly Items	ALLOWASSEMBLYPURCHASE
Allow Overage On Work Order Transactions	OVERBUILDS
Default Work Order Status	DEFAULTWORKORDERSTATUS
Create Work Orders In Supply Planning	CREATEWORKORDERINPLANNING
Check Completed Quantity In Prior Operations During Operation Completion	VALIDATEPREDECESSORCOMPLETEDQTY
Show Planned Capacity On Work Orders	SHOWPLANNEDCAPACITYONWORKORDERS
Default Scheduling Method	DEFAULTSCHEDULINGMETHOD
On the Time & Expenses tab	
Show Jobs Only for Time and Expense Entry	TIMEEXPENSEJOBONLY
Automatically Notify Supervisor	AUTONOTIFYSUPV
Expenses Billable by Default	DEFAULTEXPENSEBILLABLE
Items Billable by Default	DEFAULTITEMSBILLABLE
Combine Detail Items on Expense Reports	COMBINEEXPENSEITEMS
Copy Expense Memos to Invoices	COPYEXPENSEMEMOS
Default Advance To Apply Account For Expense Reports	DEFAULT_ADVANCE_ACCT_FOR_EXPREPT
Foreign Amount Change	EXP_REPORT_FOREIGN_AMOUNT_CHANGE

Preference UI Label	Preference Internal ID
On the Approval Routing tab	
Expense Reports	CUSTOMAPPROVALEXPENSE
Purchase Orders	CUSTOMAPPROVALPURCHORD
Vendor Bills	CUSTOMAPPROVALVENDORBILL
Requisitions	CUSTOMAPPROVALPURCHREQ
Invoices	CUSTOMAPPROVALCUSTINVC
Purchase Contracts	CUSTOMAPPROVALPURCHCON
Blanket Purchase Orders	CUSTOMAPPROVALBLANKORD
Journal Entries	CUSTOMAPPROVALJOURNAL

Accounting Periods

These are the account preferences that can be found by going to Setup > Accounting > Manage G/L > Manage Accounting Periods.

The internal ID for the Accounting Periods page is **accountingperiods**. All preference internal IDs are case-insensitive.

Preference UI Label	Preference Internal ID
First Fiscal Month	fiscalmonth
Fiscal Year End	fiscalyear
Period Format	periodstyle
Year in Period Name	periodnameyear
One-Day Year-End Adjustment Period	lastday

Tax Setup

The internal ID for the Set Up Taxes page is **taxpreferences**. All preference internal IDs are case-insensitive.

Preference IDs for fields on this page vary according to the country of the nexus. Field internal IDs are suffixed with the nexus country code. The format for these preferences is <field internal id><nexus country code>. Be aware that different fields are available for different nexuses.

The table below shows some scriptable tax preference fields for a US nexus. Fields are suffixed with **us**, for the US nexus. This table is provided for example purposes.

Preference UI Label	Preference Internal ID
	defaulttaxableus
	type
	chargoutofdistrictus
	perlinetaxesus
	storeordertaxationus

Preference UI Label	Preference Internal ID
	enabletaxlookupus

Field IDs in your account may be suffixed with a different nexus country code. And different fields may be available. You may be able to look up field IDs in the user interface, by going to Setup > Accounting > Set Up Taxes, and clicking on a nexus.

- To make field IDs available, go to Home > Set Preferences and ensure that the Show Internal IDs box is checked on the General subtab, Defaults area.
- Find the field in the NetSuite user interface and click the field label to display the field level help text. The field ID is displayed in the popup.

Tax Periods

These are the tax preferences that can be found by going to Setup > Accounting > Taxes > Manage Tax Periods.

The internal ID for the Tax Periods page is **taxperiods**. All preference internal IDs are case-insensitive.

Preference UI Label	Preference Internal ID
First Fiscal Month	fiscalmonth
Fiscal Year End	fiscalyear
Period Format	periodstyle
Year in Period Name	periodnameyear

Task IDs

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The following table lists NetSuite tasklinks that can be referenced when using nlapiResolveURL(type, identifier, id, displayMode) or nlapiSetRedirectURL(type, identifier, id, editmode, parameters).

Tasklinks are only available if the related feature is enabled in your account. Also, the following table lists only standard tasklinks. This list does not include tasklinks related to custom objects that have been created in your account. This list also does not include tasklinks related to objects that have been added to your account as part of SuiteApp installations.

Task ID	Page Label in NetSuite	URL
ADMI_ACCOUNTMOBILITY	Account Mobility Tasks	/app/setup/accountmobilitytasks.nl
ADMI_ACCTSETUP	Accounting Preferences	/app/setup/acctsetup.nl
ADMI_ACHSETUP	Set Up ACH Processing	/app/setup/achsetup.nl
ADMI_ACTIVATEBANKACCOUNT	Activate Bank Account	/app/payroll/verifyacct.nl
ADMI_ADVINVENTORYSETUP	Inventory Management Preferences	/app/setup/advinventorysetup.nl
ADMI_ALLOW_LOGIN	NetSuite Support Login	/app/crm/support/allowsupportlogin.nl
ADMI_ALTERNATIVEPAYMENTS	Alternative Payments	/app/setup/altpaymentaccounts.nl

Task ID	Page Label in NetSuite	URL
ADMI_BACKUP	Full CSV Export	/app/external/export/backup/backup.nl?csv=T
ADMI_BASSETUP	Set Up Business Activity Statement	/app/setup/bassetup.nl
ADMI_BILLINGEVENTS	View Billing Information	/app/billing/billingevents.nl
ADMI_BILLINGTERMS	View Billing Terms	/app/billing/billingterms.nl
ADMI_BOUNCEDADDRESS	Bounced Email Addresses	/app/common/entity/bouncedaddresses.nl
ADMI_BUDGET	Import Set Up Budget	/app/setup/assistants/nsimport/simpleimport.nl?rectype=BUDGETIMPORT
ADMI_BUNDLEDETAILS	Bundle Details	/app/bundler/bundledetails.nl
ADMI_CLASSESTOLOCS	Convert Classes to Locations	/app/setup/classestolocs.nl
ADMI_CLEARACCOUNT	Delete All Data	/pages/setup/clearaccount.jsp
ADMI_CLOSEPERIOD	Lock Transactions	/app/setup/closeperiod.nl
ADMI_COMMERCECATEGORY	Admin Commerce Categories	/app/platform/services/commercecategory/commercecategories.nl
ADMI_COMMISSIONSETUP	Set Up Commissions	/app/setup/commissionsetup.nl
ADMI_COMPANY	Company Information	/app/common/otherlists/subsidiarytype.nl?forcompany=T
ADMI_CONVERTCLASSES	Convert Classes to Departments	/app/setup/convertclasses.nl
ADMI_COPYCOA	Copy Chart of Accounts to New Company	/app/signup/consolidation.nl?coa=
ADMI_CREDCARD	Set Up Credit Card Processing	/app/setup/ccprocessorlist.nl
ADMI_CSVIMPORTPREF	CSV Import Preferences	/app/setup/upload/csv/csvpreferences.nl
ADMI_CUSTAPPTEXT	Customize Fulfillment Email	/app/setup/customizetext.nl?scope=APP
ADMI_DUPLICATESETUP	Set Up Duplicate Detection	/app/setup/duplicatedetectsetup.nl
ADMI_EDITNEXUSES	New Nexus	/app/setup/nexus.nl
ADMI_EDITTAXACCTS	New Tax Control Account	/app/setup/taxacct.nl
ADMI_EDITTAXTYPES	New Tax Types	/app/setup/taxtype.nl
ADMI_EDITTEGATA	New Tegata Account	/app/setup/tegataaccount.nl
ADMI_EMAIL	Email Preferences	/app/setup/emailpreferences.nl
ADMI_EXPORTIIF	Export as IIF	/pages/setup/exportiifform.jsp
ADMI_EXPORTPRODUCTFEEDS	Product Feeds	/app/site/setup/exportproductfeeds.nl
ADMI_EXPORTPRODUCTFEEDSADV	Product Feeds	/app/site/setup/exportproductfeeds.nl
ADMI_FEATURES	Enable Features	/app/setup/features.nl

Task ID	Page Label in NetSuite	URL
ADMI_FEDEXREG	Fedex Registration Wizard	/app/common/shipping/fedex/fedexregistration.nl
ADMI_FINCHARGEPREF	Finance Charge Preferences	/app/setup/finchargepref.nl
ADMI_FISCALPERIODS	Accountng Periods	/app/setup/period/generatefiscalperiods.nl
ADMI_GENERAL	General Preferences	/app/setup/general.nl
ADMI_HEADING	Heading	#
ADMI_IMAGERESIZE	Image Resizing	/app/site/setup/imageresizeadmin.nl
ADMI_IMPORTCSV	Import CSV Records	/app/setup/assistants/nsimport/importassistant.nl?new=T
ADMI_IMPORTCSV_LOG	View CSV Import Status	/app/setup/upload/csv/csvstatus.nl
ADMI_IMPORTOLB	Import Online Banking Data	/pages/setup/importolbform.jsp
ADMI_IMPORTQIF	Import Quicken® QIF File	/pages/setup/importqifform.jsp
ADMI_IMPORT_COUPONCODE_STATUS	Import Coupon Codes Status	/app/setup/upload/csv/csvstatus.nl?importtype=COUPONCODE
ADMI_INTEGR_APP	Manage Integrations	/app/common/integration/integrapplist.nl
ADMI_INVOICESETUP	Invoicing Preferences	/app/setup/invoicesetup.nl
ADMI_ISSUEBUILD	Product Builds	/app/crm/support/issuedb/issuebuilds.nl
ADMI_ISSUEEXTSTATUS	Issue External Statuses	/app/crm/support/issuedb/issueextstatus.nl
ADMI_ISSUEIMPORT	Import Issue Records	/app/setup/assistants/nsimport/importassistant.nl
ADMI_ISSUEMODULE	Product Modules	/app/crm/support/issuedb/issuemodule.nl
ADMI_ISSUEPRIORITY	Issue Priorities	/app/crm/support/issuedb/issuepriority.nl
ADMI_ISSUEPRODUCT	Products	/app/crm/support/issuedb/issueproducts.nl
ADMI_ISSUERELEASEUPDATE	Manage Released Issues	/app/crm/support/issuedb/issuereleaseupdate.nl
ADMI_ISSUEREPRODUCE	Issue Reproducibility	/app/crm/support/issuedb/issuereproduce.nl
ADMI_ISSUESETUP	Issue Preferences	/app/setup/issuesetup.nl
ADMI_ISSUESEVERITY	Issue Severities	/app/crm/support/issuedb/issueseverity.nl
ADMI_ISSUESOURCE	Issue Sources	/app/crm/support/issuedb/issuesource.nl
ADMI_ISSUESTATUS	Issue Statuses	/app/crm/support/issuedb/issuestatus.nl
ADMI_ISSUESTATUSFLOW	Manage Status Transitions	/app/crm/support/issuedb/issuestatusflow.nl
ADMI_ISSUETAGS	Issue Tags	/app/crm/support/issuedb/issuetags.nl
ADMI_ISSUETYPE	Issue Types	/app/crm/support/issuedb/issuetype.nl
ADMI_ISSUEUSERTYPE	Issue Roles	/app/crm/support/issuedb/issueusertypes.nl
ADMI_ISSUEVERSION	Product Versions	/app/crm/support/issuedb/issueversion.nl

Task ID	Page Label in NetSuite	URL
ADMI_ITEMATTRGROUP	Field Set	/app/site/setup/fieldset.nl
ADMI_LEADCUSTOMFIELDMAPPING	Lead Conversion Mapping	/app/common/custom/custleadfieldmapping.nl
ADMI_LOGINAUDIT	View Login Audit Trail	/app/common/search/search.nl?searchtype=LoginAudit
ADMI_LOGINRESTRICT	View Login Restrictions	/app/setup/userloginrestrictions.nl
ADMI_MAINTENANCEDOMAIN	Site Maintenance Domain	/app/site/setup/maintenancedomain.nl
ADMI_MAINTENANCEDOMAINADV	Site Maintenance Domain	/app/site/setup/maintenancedomain.nl
ADMI_MAINTENANCEDOMAINLIST	Site Maintenance Domains	/app/setup/maintenancedomains.nl
ADMI_MAINTENANCEDOMAINLISTADV	Site Maintenance Domains	/app/setup/maintenancedomains.nl
ADMI_MANAGEPAYROLL	Update Payroll Information	/app/payroll/managepayroll.nl
ADMI_MANAGE_PLUGINS	Manage Plug-ins	/app/common/scripting/manageplugins.nl
ADMI_NAMING	Rename Records/Transactions	/app/setup/naming.nl
ADMI_NETANSWERS	NetAnswers Knowledge Base	javascript:window.open("http://shopping.netsuite.com/netledger","NetAnswers","resizable,scrollbars=yes");void(0)
ADMI_NEXUSES	Nexuses	/app/setup/nexuses.nl
ADMI_NOTIFICATIONS	Administrative Notifications	/app/setup/notifications.nl?id=0
ADMI_NUMBERING	Set Up Auto-Generated Numbers	/app/setup/numbering.nl
ADMI_OAUTH_TOKENS	Access Tokens	/app/setup/accesstokens.nl
ADMI_OPENIDSSO	OpenID Single Sign-on	/app/setup/openidsetup.nl
ADMI_OTHERSTUB	Other Section Task	#
ADMI_OUTLOOKINTEGRATION	Outlook Integration	/app/external/xml/outlook/outlookDownload.nl
ADMI_OUTLOOKINTEGRATION_V3	Outlook Integration 3.0	/app/external/xml/outlook/outlookv3download.nl
ADMI_PAYPAL	Set Up PayPal Processing	/app/common/otherlists/accountingotherlist.nl?paymentmethodtype=paypal&system=F&e=T
ADMI_PAYROLL	Set Up Payroll	/app/setup/payrollsetup.nl
ADMI_PAYROLLMAP	Map Payroll Items	/app/payroll/mappayrollitem.nl
ADMI_PAYROLLREP	Report Sections	/app/setup/payrollreportsections.nl
ADMI_PRINTING	Printing & Fax	/app/setup/printing.nl
ADMI_REDIRECT	Import Redirect	/app/setup/assistants/nsimport/simpleimport.nl?rectype=REDIRECT

Task ID	Page Label in NetSuite	URL
ADMI_REDIRECTADV	Import Redirect	/app/setup/assistants/nsimport/simpleimport.nl?rectype=REDIRECT
ADMI_REDIRECTS	Web Site Redirects	/app/site/setup/redirects.nl
ADMI_REDIRECTSADV	Web Site Redirects	/app/site/setup/redirects.nl
ADMI_SAMLSSO	SAML Single Sign-on	/app/setup/samlsetup.nl
ADMI_SANDBOXACCOUNTS	Sandbox Accounts	/app/setup/sandboxaccounts.nl
ADMI_SAVEDASH	Publish Dashboard	/app/center/setup/savedashboard.nl
ADMI_SAVEDIMPORTS	Saved CSV Imports	/app/setup/assistants/nsimport/savedimports.nl
ADMI_SCRIPTDEBUGGER	Script Debugger	/app/common/scripting/scriptdebugger.nl
ADMI_SETUPMANAGER	Setup Manager	/app/setup/mainsetup.nl
ADMI_SETUPURLS	Bulk Set URL Components	/app/site/setup/setupurlcomponents.nl
ADMI_SETUPURLSADV	Bulk Set URL Components	/app/site/setup/setupurlcomponents.nl
ADMI_SETUPURLSADV_LOG	Bulk Set URL Components Status	/app/external/xml/upload/uploadlog.nl?displayType=SITESETUP
ADMI_SETUPURLS_LOG	Bulk Set URL Components Status	/app/external/xml/upload/uploadlog.nl?displayType=SITESETUP
ADMI_SETUPYEARSTATUS	Set Up Year Status	/app/setup/period/generateperiodsstatus.nl
ADMI_SFASETUP	Sales Preferences	/app/setup/sfasetup.nl
ADMI_SHIPPING	Set Up Shipping	/app/setup/shipping.nl
ADMI_SITEMAPGENERATOR	Sitemap Generator	/app/site/setup/sitemap/sitemapgenerator.nl
ADMI_SITEMAPGENERATORADV	Sitemap Generator	/app/site/setup/sitemap/sitemapgenerator.nl?sitetype=ADVANCED
ADMI_SITEMAP_MANAGER	Content Manager	/app/site/setup/sitemanager.nl
ADMI_SOFTDESCRIPTORS	Set Up Credit Card Soft Descriptors	/app/setup/softdescriptors.nl
ADMI_STATETAXIMPORT	Use State Sales Tax Tables	/app/setup/statetaximport.nl
ADMI_STOREADMIN	Set Up Web Site	/app/site/setup/siteadmin.nl?sitetype=STANDARD
ADMI_STOREADMINADV	Set Up Web Site	/app/site/setup/siteadmin.nl?sitetype=ADVANCED
ADMI_STOREASSISTANT	Web Site Assistant	/app/setup/assistants/sitesetup.nl
ADMI_STORELIST	Set Up Web Site	/app/site/setup/sitelist.nl?sitetype=STANDARD
ADMI_STORELISTADV	Set Up Web Site	/app/site/setup/sitelist.nl?sitetype=ADVANCED
ADMI_STOREPREVIEW	Preview Web Site	/app/site/setup/sitepreview.nl
ADMI_STOREPREVIEWADV	Preview Web Site	/app/site/setup/sitepreview.nl

Task ID	Page Label in NetSuite	URL
ADMI_SUITESIGNON	SuiteSignOn	/app/setup/ssoapplist.nl
ADMI_SUPPORTSETUP	Support Preferences	/app/setup/supportsetup.nl
ADMI_SWAPPRICES	Swap Prices Between Price Levels	/app/common/item/swapprices.nl
ADMI_SYNCHRONIZATIONSETUP	Synchronization	/app/external/xml/pumatech/syncDownload.nl
ADMI_TAX	Set Up Taxes	/app/setup/taxpreferences.nl
ADMI_TAXACCTS	Tax Control Accounts	/app/setup/taxaccts.nl
ADMI_TAXTYPES	Tax Types	/app/setup/taxtypes.nl
ADMI_TEGATA	Tegata Accounts	/app/setup/tegataaccounts.nl
ADMI_TEXTCUST	Customize Web Site Text	/app/setup/customizetextbundles.nl
ADMI_TEXTCUSTADV	Customize Web Site Text	/app/setup/customizetextbundles.nl
ADMI_TEXTCUSTGROUP	Customize Text Group	/app/setup/customizetextbundle.nl
ADMI_TEXTCUSTGROUPADV	Customize Text Group	/app/setup/customizetextbundle.nl
ADMI_TRANSETUP	Set Up Transactions	/app/setup/acctsetup.nl
ADMI_TRANSITEMTXT	Bulk Update Translation	/app/setup/translateitems.nl
ADMI_TWOFACTORDEVICES	Two-Factor Authentication Tokens	/app/setup/twofactordevices.nl
ADMI_TWOFACTORPHONERESET	Two-Factor Phone Number Reset	/app/setup/adminresetphone.nl
ADMI_TWOFACTORROLES	Two-Factor Authentication Roles	/app/setup/twofactorroles.nl
ADMI_UPDATEPRICES	Update Prices	/app/common/bulk/bulkop.nl?searchtype=Item&opcode=Pricing
ADMI_UPSELLSETUP	Upsell Preferences	/app/setup/upsellsetup.nl
ADMI_UPSWIZ	UPS Registration Wizard	/app/common/shipping/ups/upsregistration.nl
ADMI_URLCOMPONENTS	URL Components for Facets	/app/site/setup/faceturlcomponentmanager.nl
ADMI_WEBSERVICEPREFS	Web Services Preferences	/app/webservices/setup.nl
ADMI_WEBSERVICES_STATUS	Web Services Process Status	/app/webservices/asyncstatus.nl
ADMI_WEBSERVICES_USAGE_LOG	Web Services Usage Log	/app/webservices/syncstatus.nl
ADMI_XML_PAYTRUST_APPROVE	Approve Online Bill Payments	/app/external/xml/paytrust/approve.nl
ADMI_XML_PAYTRUST_SETUP	Set Up Online Bill Pay	/app/external/xml/paytrust/setup.nl
ADMI_XML_PAYTRUST_STATUS	View Online Bill Pay Status	/app/external/xml/paytrust/status.nl

Task ID	Page Label in NetSuite	URL
ADMI_YTDTAXLIBANDPYMTS	Set Up Year-to-Date Information	/app/payroll/ytdprocess.nl
EDIT_ACCOUNT	New Accounts	/app/accounting/account/account.nl
EDIT_ACCOUNTINGBOOK	New Accounting Book	/app/accounting/multibook/accountingbook.nl
EDIT_ACCOUNTINGOTHERLIST	New Accounting List Element	/app/common/otherlists/accountingotherlist.nl
EDIT_ACTIVITY	New Activity	/app/crm/calendar/activity.nl
EDIT_ADVANCEDORDERMANAGEMENTSETUP	Advanced Order Management	/app/setup/advancedordermanagement/advancedordermanagementsetup.nl
EDIT_ALLOCATION	Create Allocation Schedules	/app/accounting/transactions/allocation.nl
EDIT_ALLOCATIONBATCH	Create Allocation Batches	/app/accounting/transactions/allocationbatch.nl
EDIT_AMENDW4	Form W-4	/app/common/entity/amendw4.nl
EDIT_AMORTIZATIONSCHEDULE	New Amortization Template	/app/accounting/otherlists/revrecschedule.nl?type=Amortization
EDIT_BILLINGACCOUNT	Create Billing Accounts	/app/accounting/otherlists/billingaccount.nl
EDIT_BILLINGCLASS	New Billing Class	/app/accounting/otherlists/billingclass.nl
EDIT_BILLINGRULE	New Billing Rule	/app/accounting/transactions/billing/billingrule.nl
EDIT_BILLINGSCHEDULE	New Billing Schedule	/app/accounting/otherlists/billingschedule.nl
EDIT_BILLING_GROUPS	Billing Groups	/app/crm/common/crmgroup.nl?billinggroupsonly=T&grouptype_filter=CustJob
EDIT_BILLOFDISTRIBUTION	New Bill Of Distribution	/app/accounting/inventory/distributionplanning/billofdistribution.nl
EDIT_BILLRUNSCHEDULE	Schedule Billing Operations	/app/accounting/transactions/billingworkcenter/billrunschedule.nl
EDIT_BILLRUNSCHEDULES	New Billing Operation Schedule	/app/accounting/transactions/billingworkcenter/billrunschedule.nl
EDIT_BINNUMBERRECORD	New Bin	/app/accounting/transactions/inventory/binnumberrecord.nl
EDIT_BULKOP	Edit Mass Update	/app/common/bulk/bulkop.nl
EDIT_BUNDLE	Create Bundle	/app/setup/assistants/bundlebuilder.nl?new=T
EDIT_BUNDLEAUDITTRAIL	Bundle Audit Trail	/app/bundler/bundleauditrail.nl
EDIT_CALENDARPREFERENCE	Calendar Preference	/app/crm/calendar/calendarpreference.nl
EDIT_CALL	New Phone Call	/app/crm/calendar/call.nl
EDIT_CAMPAIGN	New Marketing Campaign	/app/crm/marketing/campaign.nl
EDIT_CAMPAIGNAUDIENCE	New Campaign Audience	/app/crm/marketing/campaignaudience.nl

Task ID	Page Label in NetSuite	URL
EDIT_CAMPAIGNBULK	Create Keyword Campaigns	/app/crm/marketing/campaign.nl?bulk=T
EDIT_CAMPAIGNBULKIMPORT	Import Keywords	/app/setup/assistants/nsimport/simpleimport.nl?rectype=CAMPAIGNKEYWORD
EDIT_CAMPAIGNCATEGORY	New Campaign Category	/app/crm/marketing/campaigncategory.nl
EDIT_CAMPAIGNCHANNEL	New Campaign Channel	/app/crm/marketing/campaignchannel.nl
EDIT_CAMPAIGNEMAIL	New Campaign Email Address	/app/crm/marketing/campaignemail.nl
EDIT_CAMPAIGNFAMILY	New Campaign Family	/app/crm/marketing/campaignfamily.nl
EDIT_CAMPAIGNOFFER	New Campaign Offer	/app/crm/marketing/campaignoffer.nl
EDIT_CAMPAIGNSEARCHENGINE	New Campaign Search Engine	/app/crm/marketing/campaignsearchengine.nl
EDIT_CAMPAIGNSUBSCRIPTION	New Campaign Subscription	/app/crm/marketing/campaignsubscription.nl
EDIT_CAMPAIGNVERTICAL	New Campaign Vertical	/app/crm/marketing/campaignvertical.nl
EDIT_CASEFIELDRULE	New Case Rule	/app/crm/support/casefieldrule.nl
EDIT_CASEFORM	New Online Case Forms	/app/crm/support/caseform.nl
EDIT_CASEISSUE	New Case Issue	/app/crm/support/caseissue.nl
EDIT_CASEORIGIN	New Case Origin Type	/app/crm/support/caseorigin.nl
EDIT_CASEPRIORITY	New Case Priority	/app/crm/support/casepriority.nl
EDIT_CASEPROFILE	New Case Profile	/app/crm/support/profiles/caseprofile.nl
EDIT_CASESTATUS	New Case Status	/app/crm/support/casestatus.nl
EDIT_CASETERRITORY	New Case Territory	/app/crm/support/supportterritory.nl
EDIT_CASETYPE	New Case Type	/app/crm/support/casetype.nl
EDIT_CHARGE	Create Charges	/app/accounting/transactions/billing/charge.nl
EDIT_CHARGERULE	New Charge Rule	/app/accounting/transactions/billing/chargerule.nl
EDIT_CLASS	New Class	/app/common/otherlists/classtype.nl
EDIT_CLASSSEGMENTMAPPING	New Class Mapping	/app/accounting/account/classsegmentmapping.nl
EDIT_COLORTHEME	New Color Theme	/app/setup/look/colortheme.nl
EDIT_COMMERCECATALOG	Manage Commerce Category Catalog	/app/platform/services/commercecategory/commercecatalog.nl
EDIT_COMMERCECATEGORY	Manage Commerce Category	/app/platform/services/commercecategory/commercecategory.nl
EDIT_COMMISSIONSCHEDULE	New Employee Schedule	/app/crm/sales/commissions/commissionschedule.nl
EDIT_COMPANY	New Company	/app/common/entity/company.nl
EDIT_COMPETITOR	Competitor	/app/crm/sales/competitor.nl

Task ID	Page Label in NetSuite	URL
EDIT_CONTACT	New Contacts	/app/common/entity/contact.nl
EDIT_CRMGROUP	New Groups	/app/crm/common/crmgroup.nl
EDIT_CRMMESSAGE	New Email	/app/crm/common/crmmessage.nl
EDIT_CRMOTHERLIST	New CRM List Element	/app/common/otherlists/crmotherlist.nl
EDIT_CRMTEMPLATE	New Marketing Templa tes	/app/crm/common/merge/marketingtempl ate.nl
EDIT_CURRENCY	New Currencies	/app/common/multicurrency/currency.nl
EDIT_CURRENCYRATE	New Currency Exchange Rate	/app/common/multicurrency/currencyrate. nl
EDIT_CUSTADDRESSENTRYFORM	Address Form	/app/common/custom/custaddressentryf orm.nl
EDIT_CUSTADDRESSFORM	Address Form	/app/common/custom/custaddressform.nl? e=T
EDIT_CUSTBODYFIELD	New Transaction Body Fields	/app/common/custom/bodycustfield.nl
EDIT_CUSTCATEGORY	New Center Category	/app/common/custom/custcategory.nl
EDIT_CUSTCENTER	New Center	/app/common/custom/custcenter.nl
EDIT_CUSTCOLUMNFIELD	New Transaction Col umn Fields	/app/common/custom/columncustfield.nl
EDIT_CUSTEMAILLAYOUT	New Transaction Form HTML Layouts	/app/common/custom/custemaillayout.nl
EDIT_CUSTENTITYFIELD	New Entity Fields	/app/common/custom/entitycustfield.nl
EDIT_CUSTENTRYFORM	New Entry Forms	/app/common/custom/custentryform.nl
EDIT_CUSTEVENTFIELD	New CRM Fields	/app/common/custom/eventcustfield.nl
EDIT_CUSTFORM	New Transaction Forms	/app/common/custom/custform.nl
EDIT_CUSTITEMFIELD	New Item Fields	/app/common/custom/itemcustfield.nl
EDIT_CUSTITEMNUMBERFIELD	New Item Number Field	/app/common/custom/itemnumbercustfie ld.nl
EDIT_CUSTJOB	New Customers	/app/common/entity/custjob.nl
EDIT_CUSTLAYOUT	New Transaction Form PDF Layouts	/app/common/custom/custlayout.nl
EDIT_CUSTLIST	New Lists	/app/common/custom/custlist.nl
EDIT_CUSTOMERFIELDRULE	New Set Up Sales Rules	/app/crm/sales/customerfieldrule.nl
EDIT_CUSTOMERFORM	New Online Customer Forms	/app/crm/sales/leadform.nl
EDIT_CUSTOMERSTATUS	New Customer Statuses	/app/crm/sales/customerstatus.nl
EDIT_CUSTOMSEGMENT	Edit Custom Segment	/app/common/custom/segments/segment. nl
EDIT_CUSTOTHERFIELD	New Other Field	/app/common/custom/othercustfield.nl
EDIT_CUSTPROFILE	Customer Profile	/app/common/entity/custprofile.nl?category =billing&sc=4

Task ID	Page Label in NetSuite	URL
EDIT_CUSTRECORD	New Record Types	/app/common/custom/custrecord.nl
EDIT_CUSTRECORDFIELD	Custom Record Field	/app/common/custom/custreccustfield.nl
EDIT_CUSTRECORDFORM	New Custom Record Form	/app/common/custom/custrecordform.nl
EDIT_CUSTSCRIPTFIELD	New Script Fields	/app/common/custom/scriptcustfield.nl
EDIT_CUSTSECTION	New Center Tab	/app/common/custom/custsection.nl
EDIT_CUSTTAB	New Subtab	/app/common/custom/subtab.nl
EDIT_CUSTTASKS	Center Links	/app/common/custom/custtasks.nl
EDIT_CUSTTRANSACTION	New Transaction Type	/app/common/custom/customtransaction.nl
EDIT_CUSTWFSTATEFIELD	New Workflow State Fields	/app/common/custom/wfstatecustfield.nl
EDIT_CUSTWORKFLOWFIELD	New Workflow Fields	/app/common/custom/workflowcustfield.nl
EDIT_CUST_	Custom Record Entry	/app/common/custom/custrecordentry.nl
EDIT_DEPARTMENT	New Departments	/app/common/otherlists/departmenttype.nl
EDIT_DEPLOYMENTAUDITTRAIL	Deployment Audit Trail	/app/suiteapp/devframework/deploymentauditrail.nl
EDIT_DEPTSEGMENTMAPPING	New Department Mapping	/app/accounting/account/deptsegmentmapping.nl
EDIT_DISTRIBUTIONNETWORK	New Distribution Network	/app/accounting/inventory/distributionplanning/distributionnetwork.nl
EDIT_DRIVERSLICENSE	New Driver's Liscense	/app/hcm/hris/governmentid/driverslicense.nl
EDIT_EDITPROFILE	Employee Profile	/app/common/entity/editprofile.nl
EDIT_EMAILTEMPLATE	New Email Templates	/app/crm/common/merge/emailtemplate.nl
EDIT_EMPLCATEGORY	New Employee Directory	/app/site/setup/emplcategory.nl
EDIT_EMPLOYEE	New Employees	/app/common/entity/employee.nl
EDIT_EMPLOYEESFA	New Assign Support Reps	/app/common/entity/employeesfa.nl
EDIT_EMPOTHERLIST	New Employee Related List Element	/app/common/otherlists/empotherlist.nl
EDIT_ENTITYACCOUNTMAPPING	New Entity Account Mapping	/app/accounting/account/entityaccountmapping.nl
EDIT_ESCALATIONRULE	New Escalation Rules	/app/crm/support/escalationfieldrule.nl
EDIT_ESCALATIONTERRITORY	New Manage Escalation Assignment	/app/crm/support/escalationterritory.nl
EDIT_EVENT	New Event	/app/crm/calendar/event.nl
EDIT_EXPCATEGORY	New Expense Categories	/app/accounting/otherlists/expcategory.nl
EDIT_FAIRVALUEDIMENSION	Fair Value Dimension	/app/accounting/revrec/fairvaluedimension.nl

Task ID	Page Label in NetSuite	URL
EDIT_FAIRVALUEFORMULA	New Fair Value Formula	/app/accounting/revrec/fairvalueformula.nl
EDIT_FAIRVALUEPRICE	New Fair Value Price	/app/accounting/revrec/fairvalueprice.nl
EDIT_FAXMESSAGE	New Fax Message	/app/crm/common/merge/faxmessage.nl
EDIT_FAXTEMPLATE	New Fax Templates	/app/crm/common/merge/faxtemplate.nl
EDIT_FISCALCALENDAR	New Fiscal Calendar	/app/setup/period/fiscalcalendar.nl
EDIT_FISCALPERIOD	Setup Fiscal Period	/app/setup/period/fiscalperiod.nl
EDIT_GENERICRESOURCE	New Generic Resource	/app/accounting/project/genericresource.nl
EDIT_GLOBALACCOUNTMAPPING	New Global Account Mapping	/app/accounting/account/globalaccountmapping.nl
EDIT_GOVERNMENTISSUEDIDTYPE	New Government-Issued ID Type	/app/hcm/hris/governmentid/governmentissuedidtype.nl
EDIT_HCMJOB	New Job	/app/hcm/hris/job.nl
EDIT_IC_ALLOCATION	Create Intercompany Allocation Schedules	/app/accounting/transactions/intercompanyallocation.nl
EDIT_IMPORT_COUPONCODE	Import Coupon Codes	/app/setup/assistants/nsimport/simpleimport.nl?rectype=COUPONCODE
EDIT_INFOITEM	New Information Items	/app/site/setup/infoitem.nl?Information_TYPE=TEXT
EDIT_INFOITEMFORM	New Publish Forms	/app/site/setup/infoitem.nl?Information_TYPE=FORM
EDIT_INSTALLBUNDLE	Search & Install Bundles	/app/bundler/installbundle.nl
EDIT_INTEGR_APP	Integration	/app/common/integration/integrapp.nl
EDIT_INVCOSTTEMPLATE	New Inventory Cost Template	/app/accounting/inventory/standard/invcosttemplate.nl
EDIT_ISSUE	New Issue	/app/crm/support/issuedb/issue.nl
EDIT_ISSUEPRODUCT	New Product	/app/crm/support/issuedb/issueproduct.nl
EDIT_ISSUETAG	New Issue Tag	/app/crm/support/issuedb/issuetag.nl
EDIT_ISSUEUSERTYPE	New Issue Role	/app/crm/support/issuedb/issueusertype.nl
EDIT_ITEM	New Items	/app/common/item/item.nl
EDIT_ITEMACCOUNTMAPPING	New Item Account Mapping	/app/accounting/account/itemaccountmapping.nl
EDIT_ITEMDEMANDPLAN	New Item Demand Plan	/app/accounting/inventory/demandplanning/itemdemandplan.nl
EDIT_ITEMOPTION	New Transaction Item Options	/app/common/custom/itemoption.nl
EDIT_ITEMREVENUECATEGORY	New Item Revenue Category	/app/accounting/revrec/itemrevenuecategory.nl
EDIT_ITEMSUPPLYPLAN	New Item Supply Plan	/app/accounting/inventory/demandplanning/itemsupplyplan.nl
EDIT_ITEM_REVISION	New Item Revision	/app/common/item/itemrevision.nl
EDIT_JOB	New Job	/app/accounting/project/project.nl

Task ID	Page Label in NetSuite	URL
EDIT_JOBREQUISITION	New Job Requisition	/app/hcm/hris/jobrequisition.nl
EDIT_KBCATEGORY	New Knowledge Base	/app/site/setup/kbcategory.nl
EDIT_KPIREPORT	New KPI Scorecard	/app/center/enhanced/kpireportsetup.nl
EDIT_KUDOS	New Kudos	/app/hcm/perfmgmt/kudos.nl
EDIT_LEAD	New Leads	/app/common/entity/custjob.nl?stage=lead
EDIT_LOCASSIGNCONF	Configurations	/app/setup/locationassignment/configuration/locationassignmentconfiguration.nl
EDIT_LOCASSIGNRULE	Rules Edit	/app/setup/locationassignment/configuration/rule/locationassignmentrule.nl
EDIT_LOCASSIGNRULES_AUDIT_TRAIL	Rules Audit Trail	/app/common/search/search.nl?searchtype=LocAssignRule&audittrail=T
EDIT_LOCATION	New Locations	/app/common/otherlists/locationtype.nl
EDIT_LOCATIONCOSTINGGROUP	New Location Costing Group	/app/accounting/inventory/costing/locationcostinggroup.nl
EDIT_LOCSEGMENTMAPPING	New Location Mapping	/app/accounting/account/locsegmentmapping.nl
EDIT_MAILMERGE	Bulk Merge	/app/crm/common/merge/mailmerge.nl
EDIT_MAILMESSAGE	New Word Message	/app/crm/common/merge/mailmessage.nl
EDIT_MAILTEMPLATE	New Letter Templates	/app/crm/common/merge/mailtemplate.nl
EDIT_MEDIAITEM	New Media Item	/app/common/media/mediaitem.nl
EDIT_MEDIAITEMFOLDER	New File Cabinet	/app/common/media/mediaitemfolder.nl
EDIT_MEMDOC	Enter Memorized Transactions	/app/accounting/transactions/bulkmemdoc.nl
EDIT_MFGCOSTTEMPLATE	New Manufacturing Cost Template	/app/accounting/manufacturing/mfgcosttemplate.nl
EDIT_MFGROUTING	New Manufacturing Routing	/app/accounting/manufacturing/mfgrouting.nl
EDIT_NEXUS	Nexus	/app/setup/nexus.nl
EDIT_NOTIFICATION	New Notification	/app/email/alerts/notification.nl
EDIT_OAUTH_TOKEN	New Access Token	/app/setup/accesstoken.nl
EDIT_OPENIDSSO	OpenID Single Sign-on	/app/setup/openidsetup.nl
EDIT_ORGANIZATIONVALUE	New Organization Value	/app/hcm/perfmgmt/organizationvalue.nl
EDIT_OTHERGOVERNMENTISSUEDID	New Other Government-Issued ID	/app/hcm/hris/governmentid/othergovernmentissuedid.nl
EDIT_OTHERNAME	New Other Names	/app/common/entity/othername.nl
EDIT_PARTNER	New Partners	/app/common/entity/partner.nl
EDIT_PARTNERCOMMISSIONSCHED	New Partner Schedule	/app/crm/sales/commissions/partnercommissionschedule.nl
EDIT_PARTNERPLANASSIGN	New Partner Plan	/app/crm/sales/commissions/planassignpartner.nl

Task ID	Page Label in NetSuite	URL
EDIT_PASSPORT	New Passport	/app/hcm/hris/governmentid/passport.nl
EDIT_PAYMETHODS	Create Payment Methods	/app/common/otherlists/accountingotherlist.nl?tname=paymeth&longitemtype=Payment+Method
EDIT_PAYPALACCOUNT	PayPal Accounts	/app/external/paypal/paypalaccount.nl
EDIT_PAYROLLBATCH2	New Payroll Batch	/app/payroll/payrollbatch2.nl
EDIT_PAYROLLITEM	New Payroll Items	/app/common/item/payrollitem.nl
EDIT_PAYTERMS	Create Payment Terms	/app/common/otherlists/accountingotherlist.nl?tname=terms&longitemtype=Term
EDIT_PDFMESSAGE	New PDF Message	/app/crm/common/merge/pdfmessage.nl
EDIT_PDFTEMPLATE	New PDF Templates	/app/crm/common/merge/pdftemplate.nl
EDIT_PERIOD	New Manage Accounting Periods	/app/setup/period/fiscalperiod.nl
EDIT_PLANASSIGN	New Employee Plan	/app/crm/sales/commissions/planassign.nl
EDIT_PLANNEDSTANDARDCOST	New Planned Standard Cost	/app/accounting/inventory/plannedstandardcost.nl
EDIT_PLUGIN	New Plug-in	/app/common/scripting/uploadPluginImpl.nl
EDIT_PLUGINTYPE	New Custom Plug-in Type	/app/common/scripting/createPluginType.nl
EDIT_POSITION	New Position	/app/hcm/hris/position.nl
EDIT_PRESCATEGORY	New Categories	/app/site/setup/prescategory.nl
EDIT_PROCESSHISTTXN	Historical Transaction Processing	/app/accounting/multibook/htp/submission.nl
EDIT_PROCESSHISTTXNLOG	Historical Transaction Processing Log	/app/accounting/multibook/htp/log.nl
EDIT_PROJECTEXPENSETYPE	New Project Expense Type	/app/accounting/otherlists/projectexpensetype.nl
EDIT_PROJECTTASK	New Job Tasks	/app/accounting/project/projecttask.nl
EDIT_PROJECTTEMPLATE	New Project Template	/app/accounting/project/projecttemplate.nl
EDIT_PROSPECT	New Prospects	/app/common/entity/custjob.nl?stage=prospect
EDIT_PUBLISHER	Register Publisher	/app/suiteapp/devframework/publisherregistry.nl
EDIT_PUBLISHERAPP	New Application	/app/suiteapp/devframework/appregistry.nl
EDIT_QUANTITYPRICINGSCHEDULE	New Quantity Pricing Schedule	/app/accounting/otherlists/quantitypricingschedule.nl
EDIT_RATEPLAN	New Rate Plan	/app/accounting/otherlists/rateplan.nl
EDIT_REDIRECT	New Redirect	/app/site/setup/redirect.nl
EDIT_REDIRECTADV	New Redirect	/app/site/setup/redirect.nl
EDIT_REFERRALCODE	New Promotion	/app/crm/sales/referralcode.nl
EDIT_REGION	New Region	/app/setup/region/region.nl

Task ID	Page Label in NetSuite	URL
EDIT_REGISTERPUBLISHER	Register Publisher	/app/suiteapp/devframework/publisherregistry.nl
EDIT_RELATEDITEM	New Related Items Category	/app/site/setup/relateditem.nl
EDIT_RELATEDITEMADV	New Related Items Category	/app/site/setup/relateditem.nl
EDIT_REPORT	Edit Report	/app/reporting/reporteditor.nl
EDIT_RESOLVECONFLICTS	Resolve Conflicts	/app/common/entity/conflictresolution.nl
EDIT_RESOURCE	Resource	/app/crm/calendar/resource.nl
EDIT_REVENUEALLOCATIONGROUP	New Revenue Allocation Group	/app/common/otherlists/accountingotherlist.nl?tname=revenueallocationgroup
EDIT_REVENUEELEMENT	New Revenue Element	/app/accounting/revrec/revenueelement.nl
EDIT_REVENUEPLAN	New Revenue Recognition Plan	/app/accounting/revrec/revenueplan.nl
EDIT_REVENUERECOGNITIONRULE	New Revenue Recognition Rule	/app/accounting/revrec/revenuerecognitionrule.nl
EDIT_REVRECFIELDMAPPING	Revenue Recognition Field Mapping	/app/accounting/revrec/revrecfieldmapping.nl
EDIT_REVRECSCHEDULE	New Revenue Recognition Template	/app/accounting/otherlists/revrecschedule.nl
EDIT_ROLE	New Role	/app/setup/role.nl
EDIT_RSRCALLOCATION	New Rsrc Allocation	/app/accounting/project/allocation.nl
EDIT_RSSFEED	New RSS Feed	/app/site/hosting/rssfeed.nl
EDIT_SALESCAMPAIGN	New Sales Campaign	/app/crm/marketing/salescampaign.nl
EDIT_SALESTEAM	New Sales Teams	/app/crm/common/crmgroup.nl?grouptype=SalesTeam
EDIT_SALESTERRITORY	New Manage Sales Territories	/app/crm/sales/salesterritory.nl
EDIT_SAVEDSEARCH	New Saved Search	/app/common/search/search.nl?cu=T&e=F
EDIT_SCRIPT	New Script	/app/common/scripting/uploadScriptFile.nl
EDIT_SCRIPTEDRECORD	Scripted Record	/app/common/scripting/scriptedrecord.nl
EDIT_SEARCH	New Search	/app/common/search/search.nl
EDIT_SHIPITEM	New Shipping Items	/app/common/item/shipitem.nl
EDIT_SHIPPARTREGISTRATION	New Shipping Partner Registration	/app/common/shipping/openapi/registration.nl
EDIT_SITEEMAILTEMPLATE	New Web Store Email Template	/app/site/setup/siteemailtemplate.nl
EDIT_SITEITEMTEMPLAT	Item Template	/app/site/setup/siteitemtemplate.nl
EDIT_SITEMEDIA	Site Media	/app/site/media/sitemedia.nl
EDIT_SITETAG	Web Site Tag	/app/site/setup/sitetag.nl
EDIT_SITETHEME	Web Site Theme	/app/site/setup/sitetheme.nl

Task ID	Page Label in NetSuite	URL
EDIT_SOLUTION	New Solutions	/app/crm/support/kb/solution.nl
EDIT_SSCATEGORY	New Publish Saved Sea rch	/app/site/setup/sscategory.nl
EDIT_STANDARDCOSTVERSION	New Standard Cost Ver sion	/app/accounting/inventory/standardcostvers ion.nl
EDIT_STATE	New State/Province/Cou nty	/app/setup/state.nl
EDIT_STOREITEMLISTLAYOUT	New Layouts	/app/site/setup/storeitemlistlayout.nl
EDIT_STORETAB	New Tabs	/app/site/setup/storetab.nl
EDIT_SUBSCRIPTION	Create Subscriptions	/app/accounting/subscription/subscription. nl
EDIT_SUBSCRIPTIONCHANGEORDER	Manage Subscriptions	/app/accounting/subscription/subscriptionc hangeorder.nl
EDIT_SUBSCRIPTIONLINE	New Subscription Line	/app/accounting/subscription/subscriptionl ine.nl
EDIT_SUBSCRIPTIONPLAN	New Subscription Plan	/app/common/item/item.nl?itemtype=Subsc riPlan
EDIT_SUBSIDIARY	New Subsidiaries	/app/common/otherlists/subsidiarytype.nl
EDIT_SUITEAPPLIST	Installed SuiteApp List	/app/suiteapp/devframework/appinstalllist. nl
EDIT_SUITESIGNON	SuiteSignOn	/app/setup/ssoapp.nl
EDIT_SUPPORTCASE	New Cases	/app/crm/support/supportcase.nl
EDIT_SYSALERT	System Alert	/app/common/otherlists/systemalert.nl
EDIT_TASK	New Tasks	/app/crm/calendar/task.nl
EDIT_TAXACCT	Tax Control Account	/app/setup/taxacct.nl
EDIT_TAXGROUP	New Tax Groups	/app/common/item/taxgroup.nl
EDIT_TAXITEM	New Tax Codes	/app/common/item/taxitem.nl
EDIT_TAXPERIOD	New Manage Tax Period s	/app/setup/period/taxperiod.nl
EDIT_TAXSCHEDULE	New Tax Schedule	/app/common/item/taxschedule.nl
EDIT_TAXTYPE	Tax Type	/app/setup/taxtype.nl
EDIT_TEMPLATECATEGORY	New Template Category	/app/crm/common/merge/templatecatego ry.nl
EDIT_TERMINATIONREASON	New Termination Rea son	/app/hcm/hris/terminationreason.nl
EDIT_TIMEOFFCHANGE	New Time-Off Change	/app/hcm/hris/timeoffmanagement/timeoff change.nl
EDIT_TIMEOFFPLAN	New Time-Off Plan	/app/hcm/hris/timeoffmanagement/timeoff plan.nl
EDIT_TIMEOFFREQUEST	New Time-Off Request	/app/hcm/hris/timeoffmanagement/timeoff request.nl

Task ID	Page Label in NetSuite	URL
EDIT_TIMEOFFRULE	New Time-Off Rule	/app/hcm/hris/timeoffmanagement/timeoffrule.nl
EDIT_TIMEOFFTYPE	New Time-Off Type	/app/hcm/hris/timeoffmanagement/timeofftype.nl
EDIT_TIMESHEET	Enter Time	/app/accounting/transactions/timesheet.nl
EDIT_TOPIC	New Topics	/app/crm/support/kb/topic.nl
EDIT_TRANSACTIONLIST	Paycheck History	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Paycheck
EDIT_TRAN_ADJJOURNAL	Make Currency Adjustment Journal Entries	/app/accounting/transactions/journal.nl?adjustmentjournal=T
EDIT_TRAN_BINTRNFR	Bin Transfer	/app/accounting/transactions/bintrnfr.nl
EDIT_TRAN_BINWKSHT	Bin Putaway Worksheet	/app/accounting/transactions/binwksht.nl
EDIT_TRAN_BLANKORD	Enter Blanket Purchase Order	/app/accounting/transactions/blankord.nl
EDIT_TRAN_BOOKICJOURNAL	Make Book Specific Intercompany Journal Entries	/app/accounting/transactions/journal.nl?bookje=T&icj=T
EDIT_TRAN_BOOKICJOURNALIMPORT	Import Book Specific Intercompany Journal Entries	/app/setup/assistants/nsimport/simpleimport.nl?rectype=INTERCOMPANYJOURNALENTRY
EDIT_TRAN_BOOKJOURNAL	Make Book Specific Journal Entries	/app/accounting/transactions/journal.nl?bookje=T
EDIT_TRAN_BOOKJOURNALIMPORT	Import Book Specific Journal Entries	/app/setup/assistants/nsimport/simpleimport.nl?rectype=JOURNALENTRY
EDIT_TRAN_BUILD	Build Assemblies	/app/accounting/transactions/build.nl
EDIT_TRAN_CARDCHRG	Use Credit Card	/app/accounting/transactions/cardchrg.nl
EDIT_TRAN_CASHRFND	Refund Cash Sales	/app/accounting/transactions/cashrfnd.nl
EDIT_TRAN_CASHSALE	Enter Cash Sales	/app/accounting/transactions/cashsale.nl
EDIT_TRAN_CHECK	Write Checks	/app/accounting/transactions/check.nl
EDIT_TRAN_COMMISSN	Individual Employee Commission	/app/accounting/transactions/commissn.nl
EDIT_TRAN_CUSTCHRG	Create Statement Charges	/app/accounting/transactions/custchrg.nl
EDIT_TRAN_CUSTCRED	Issue Credit Memos	/app/accounting/transactions/custcred.nl
EDIT_TRAN_CUSTDEP	Record Customer Deposits	/app/accounting/transactions/custdep.nl
EDIT_TRAN_CUSTINVC	Create Invoices	/app/accounting/transactions/custinvc.nl
EDIT_TRAN_CUSTPYMT	Accept Customer Payments	/app/accounting/transactions/custpymt.nl
EDIT_TRAN_CUSTRFND	Issue Customer Refund	/app/accounting/transactions/custrfnd.nl
EDIT_TRAN_DEPAPPL	Apply Customer Deposits	/app/accounting/transactions/depappl.nl

Task ID	Page Label in NetSuite	URL
EDIT_TRAN_DEPOSIT	Make Deposits	/app/accounting/transactions/deposit.nl
EDIT_TRAN_ESTIMATE	Prepare Estimates	/app/accounting/transactions/estimate.nl
EDIT_TRAN_EXPREPT	Enter Expense Reports	/app/accounting/transactions/exprept.nl
EDIT_TRAN_FFTREQ	Edit	/app/accounting/transactions/fftreq.nl
EDIT_TRAN_FXREVAL	Revalue Open Currency Balances	/app/accounting/transactions/fxrevalsetup.nl
EDIT_TRAN_ICJOURNAL	Make Intercompany Journal Entries	/app/accounting/transactions/journal.nl?icj=T
EDIT_TRAN_ICJOURNALIMPORT	Import Intercompany Journal Entries	/app/setup/assistants/nsimport/simpleimport.nl?rectype=INTERCOMPANYJOURNALENTRY
EDIT_TRAN_ICTRNFRORD	Enter Intercompany Transfer Orders	/app/accounting/transactions/trnfrord.nl?icto=T
EDIT_TRAN_INVADJST	Adjust Inventory	/app/accounting/transactions/invadjst.nl
EDIT_TRAN_INVCOUNT	Enter Inventory Count	/app/accounting/transactions/invcount.nl
EDIT_TRAN_INVDISTR	Distribute Inventory	/app/accounting/transactions/invdistr.nl
EDIT_TRAN_INVREVAL	Revalue Inventory Cost	/app/accounting/transactions/invreval.nl
EDIT_TRAN_INVTRNFR	Transfer Inventory	/app/accounting/transactions/invtrnfr.nl
EDIT_TRAN_INVWKSHT	Adjust Inventory Worksheet	/app/accounting/transactions/invwksht.nl
EDIT_TRAN_INVWKSHTIMPORT	Import Inventory Worksheets	/app/setup/assistants/nsimport/simpleimport.nl?rectype=INVENTORYWORKSHEET
EDIT_TRAN_ITEMRCPT	New Item Receipt	/app/accounting/transactions/itemrcpt.nl
EDIT_TRAN_ITEMSHIP	New Item Fulfillment	/app/accounting/transactions/itemship.nl
EDIT_TRAN_JOURNAL	Make Journal Entries	/app/accounting/transactions/journal.nl
EDIT_TRAN_JOURNALIMPORT	Import Journal Entries	/app/setup/assistants/nsimport/simpleimport.nl?rectype=JOURNALENTRY
EDIT_TRAN_LIAADJST	Liability Adjustment	/app/accounting/transactions/liaadjst.nl
EDIT_TRAN_LIABPYMT	Pay Payroll Liabilities	/app/accounting/transactions/liabpymt.nl
EDIT_TRAN_OPPRTNTY	Create Opportunities	/app/accounting/transactions/opprtnty.nl
EDIT_TRAN_ORDERPURCHREQ	Order Requisitions	/app/accounting/transactions/orderpurchreqs.nl
EDIT_TRAN_PARTNERCOMMISSN	Individual Partner Commission	/app/accounting/transactions/partnercommissn.nl
EDIT_TRAN_PAYCHECK	Create Individual Paycheck	/app/accounting/transactions/paycheck.nl
EDIT_TRAN_PAYCHECK2	Create Individual Paycheck	/app/accounting/transactions/paycheck2.nl
EDIT_TRAN_PCHKJRNL	Paycheck Journal	/app/accounting/transactions/pchkjrnl.nl
EDIT_TRAN_PURCHCON	Enter Purchase Contracts	/app/accounting/transactions/purchcon.nl

Task ID	Page Label in NetSuite	URL
EDIT_TRAN_PURCHORD	Enter Purchase Orders	/app/accounting/transactions/purchord.nl
EDIT_TRAN_PURCHORD_REQ	Enter Purchase Reques ts	/app/accounting/transactions/purchord.nl
EDIT_TRAN_PURCHREQ	Enter Requisitions	/app/accounting/transactions/purchreq.nl
EDIT_TRAN_REPLENISHINVENTORY	Replenish Location By Transfer Order	/app/accounting/transactions/inventory/bul ktransferorder/replenishinventory.nl
EDIT_TRAN_REPLENISHLOC	Replenish Location By Inventory Transfer	/app/accounting/transactions/replenishloc. nl
EDIT_TRAN_REVARRNG	Enter Revenue Arrang ements	/app/accounting/transactions/revarrng.nl
EDIT_TRAN_REVCOMM	RevComm	/app/accounting/transactions/revcomm.nl
EDIT_TRAN_REVCOMRV	RevComRv	/app/accounting/transactions/revcomrv.nl
EDIT_TRAN_REVCONTR	RevContr	/app/accounting/transactions/revcontr.nl
EDIT_TRAN_RFQ	Enter Requests For Quo te	/app/accounting/transactions/rfq.nl
EDIT_TRAN_RTNAUTH	Issue Return Authoriza tions	/app/accounting/transactions/rtnauth.nl
EDIT_TRAN_SALESORD	Enter Sales Orders	/app/accounting/transactions/salesord.nl
EDIT_TRAN_STATJOURNAL	Make Statistical Journal Entries	/app/accounting/transactions/statisticaljo urnal.nl
EDIT_TRAN_STATJOURNALIMPORT	Import Statistical Journa l Entries	/app/setup/assistants/nsimport/simpleim port.nl?rectype=STATISTICALJOURNALENTRY
EDIT_TRAN_STATSCHEDULE	Create Statistical Sch edule	/app/accounting/transactions/statistical/stat isticalschedule.nl
EDIT_TRAN_STPICKUP	Edit	/app/accounting/transactions/stpickup.nl
EDIT_TRAN_TAXLIAB	Write GST Liability	/app/accounting/transactions/vatliab.nl
EDIT_TRAN_TAXLIAB2	Write Tax Liability	/app/accounting/transactions/vatliab.nl
EDIT_TRAN_TAXPYMT	Pay Sales Tax	/app/accounting/transactions/taxpymt.nl
EDIT_TRAN_TAXPYMT2	Pay Sales Tax	/app/accounting/transactions/taxpymt.nl
EDIT_TRAN_TAXPYMTCA	Pay PST	/app/accounting/transactions/taxpymt.nl
EDIT_TRAN_TEGCOLLECT	Collect Tegata	/app/accounting/transactions/tegcollect.nl
EDIT_TRAN_TEGPAY	Pay Tegata	/app/accounting/transactions/tegpay.nl
EDIT_TRAN_TEGPYBL	Issue Tegata	/app/accounting/transactions/tegpybl.nl
EDIT_TRAN_TEGRCVBL	Receive Tegata	/app/accounting/transactions/tegrcvbl.nl
EDIT_TRAN_TRANSFER	Transfer Funds	/app/accounting/transactions/transfer.nl
EDIT_TRAN_TRNFRORD	Enter Transfer Orders	/app/accounting/transactions/trnfrord.nl
EDIT_TRAN_UNBUILD	Unbuild Assemblies	/app/accounting/transactions/unbuild.nl
EDIT_TRAN_VATLIABAU	Pay Tax Liability	/app/accounting/transactions/vatliab.nl
EDIT_TRAN_VENDAUTH	Enter Vendor Return Authorizations	/app/accounting/transactions/vendauth.nl

Task ID	Page Label in NetSuite	URL
EDIT_TRAN_VENDBILL	Enter Bills	/app/accounting/transactions/vendbill.nl
EDIT_TRAN_VENDCRED	Enter Vendor Credits	/app/accounting/transactions/vendcred.nl
EDIT_TRAN_VENDPYMT	Pay Single Vendor	/app/accounting/transactions/vendpymt.nl
EDIT_TRAN_VENDRFQ	Enter Vendor Requests For Quote	/app/accounting/transactions/vendrfq.nl
EDIT_TRAN_WITHDRAWINVENTORY	Withdraw By Transfer Order	/app/accounting/transactions/inventory/bulktransferorder/withdrawinventory.nl
EDIT_TRAN_WOCLOSE	Close Work Order	/app/accounting/transactions/woclose.nl
EDIT_TRAN_WOCOMPL	Complete Work Order	/app/accounting/transactions/wocompl.nl
EDIT_TRAN_WOISSUE	Issue Work Order	/app/accounting/transactions/woissue.nl
EDIT_TRAN_WORKORD	Enter Work Orders	/app/accounting/transactions/workord.nl
EDIT_TRAN_YTDADJST	Create Payroll Adjustment	/app/accounting/transactions/ytdadjst.nl
EDIT_UNITSTYPE	New Units Of Measure	/app/common/units/unitstype.nl
EDIT_UPGRADEBUNDLE	Managed Bundles	/app/bundler/bundlelist.nl?type=S&subtype=m
EDIT_URLALIAS	New Promotional URL	/app/setup/urlalias.nl
EDIT_USAGE	Create Usages	/app/accounting/otherlists/usage.nl
EDIT_VENDOR	New Vendors	/app/common/entity/vendor.nl
EDIT_WEBAPPS	New SSP Application	/app/common/scripting/webapp.nl
EDIT_WEBAPPSADV	New SSP Application	/app/common/scripting/webapp.nl
EDIT_WORKCALENDAR	New Work Calendar	/app/accounting/project/workcalendar.nl
EDIT_WORKFLOW	New Workflow	/app/common/workflow/setup/workflowmanager.nl
EDIT_WORKPLACE	New Workplaces	/app/common/otherlists/workplacetype.nl
LIST_ACCOUNT	Accounts	/app/accounting/account/accounts.nl
LIST_ACCOUNTINGBOOK	Accounting Books	/app/accounting/multibook/accountingbooklist.nl
LIST_ACCOUNTINGOTHERLIST	Accounting Lists	/app/common/otherlists/accountingotherlists.nl
LIST_ACTIVITY	Activity List	/app/crm/calendar/activitylist.nl
LIST_ADVPDFTEMPLATE	Advanced PDF/HTML Templates	/app/common/custom/pdftemplates.nl?sc=-90
LIST_ALLOCATION	Allocation Schedules	/app/accounting/transactions/allocschedulelist.nl
LIST_ALLOCATIONBATCH	Allocation Batches	/app/accounting/transactions/allocationbatchlist.nl
LIST_AMORTIZATION	Amortization Templates	/app/accounting/otherlists/amortizationtemplates.nl
LIST_AMORTIZATIONSCHEDULE	Amortization Schedules	/app/accounting/otherlists/amortizationschedules.nl

Task ID	Page Label in NetSuite	URL
LIST_APPDEF	App Definitions	/app/appdef/appdeflist.nl
LIST_APPPKG	App Packages	/app/appdef/apppkglist.nl
LIST_APPPUBLISHERS	Application Publishers	/app/setup/apppublishers.nl
LIST_APPROVEACH	Approve Direct Deposits	/app/accounting/transactions/approveach.nl
LIST_APPROVEEFT	Approve Electronic Fun ds Transfers	/app/accounting/transactions/approveeft.nl
LIST_APPROVERSRCALLOCATION	Approve Resource All ocations	/app/common/bulk/approval/recordapprova l.nl?type=resourceallocation
LIST_APPROVERTIMEENTRY	Approve Time	/app/common/bulk/approval/recordapprova l.nl?type=timeentry
LIST_APPROVEVP	Approve Vendor Pay ment Transfers	/app/accounting/transactions/approvevp.nl
LIST_BILLINGACCOUNT	Billing Accounts	/app/accounting/otherlists/billingaccounts.nl
LIST_BILLINGCLASS	Billing Classes	/app/accounting/otherlists/billingclasses.nl
LIST_BILLINGRULE	Billing Rule	/app/accounting/transactions/billing/billingr ules.nl
LIST_BILLINGRUNRESULTS	Billing Run Results	/app/accounting/transactions/billing/billingr unresults.nl
LIST_BILLINGSCHEDULE	Billing Schedules	/app/accounting/otherlists/billingschedule s.nl
LIST_BILLING_GROUPS	Billing Groups	/app/crm/common/crmgrouplist.nl?billing groupsonly=T
LIST_BILLING_WORK_CENTER	Billing Work Center	/app/accounting/transactions/billingworkce nter/billingworkcenter.nl
LIST_BILLOFDISTRIBUTION	Bill Of Distribution	/app/accounting/inventory/distributionplan ning/billofdistributionlist.nl
LIST_BILLRUN	Billing Operations	/app/accounting/transactions/billingworkce nter/billrunresults.nl
LIST_BILLRUNSCHEDULE	Billing Operation Sch edules	/app/accounting/transactions/billingworkce nter/billrunschedules.nl
LIST_BIN	Bins	/app/accounting/transactions/inventory/bin list.nl
LIST_BUDGET	Budgets	/app/accounting/transactions/budgetlist.nl
LIST_BUDGETRATES	Budget Exchange Rates	/app/accounting/otherlists/budgetrates.nl
LIST_BULKOP	Mass Updates	/app/common/bulk/bulkops.nl
LIST_BULKRESULTS	Mass Update Results	/app/common/bulk/bulkresults.nl
LIST_BUNDLE	List Bundles	/app/bundler/bundlelist.nl?type=S
LIST_CALENDAR	Calendar	/app/crm/calendar/calendar.nl
LIST_CALL	Phone Calls	/app/crm/calendar/calllist.nl
LIST_CAMPAIGN	Marketing Campaigns	/app/crm/marketing/campaignlist.nl?Camp aign_ISSALESCAMPAIGN=F
LIST_CAMPAIGNAUDIENCE	Campaign Audiences	/app/crm/marketing/campaignaudiences.nl

Task ID	Page Label in NetSuite	URL
LIST_CAMPAIGNCATEGORY	Campaign Categories	/app/crm/marketing/campaigncategories.nl
LIST_CAMPAIGNCHANNEL	Campaign Channels	/app/crm/marketing/campaignchannels.nl
LIST_CAMPAIGNEMAIL	Campaign Email Addresses	/app/crm/marketing/campaignemails.nl
LIST_CAMPAIGNFAMILY	Campaign Families	/app/crm/marketing/campaignfamilies.nl
LIST_CAMPAIGNOFFER	Campaign Offers	/app/crm/marketing/campaignoffers.nl
LIST_CAMPAIGNSEARCHENGINE	Campaign Search Engines	/app/crm/marketing/campaignsearchengines.nl
LIST_CAMPAIGNSUBSCRIPTION	Campaign Subscriptions	/app/crm/marketing/campaignsubscriptions.nl
LIST_CAMPAIGNVERTICAL	Campaign Verticals	/app/crm/marketing/campaignverticals.nl
LIST_CASEFIELDRULE	Set Up Case Rules	/app/crm/support/casefieldrules.nl
LIST_CASEFORM	Online Case Forms	/app/crm/support/caseforms.nl
LIST_CASEISSUE	Case Issues	/app/crm/support/caseissuelist.nl
LIST_CASEORIGIN	Case Origin Types	/app/crm/support/caseoriginlist.nl
LIST_CASEPRIORITY	Case Priorities	/app/crm/support/caseprioritylist.nl
LIST_CASEPROFILE	Case Profile	/app/crm/support/profiles/caseprofiles.nl
LIST_CASESTATUS	Case Statuses	/app/crm/support/casestatuslist.nl
LIST_CASETERRITORIES	Case Territory List	/app/crm/support/supportterritorylist.nl
LIST_CASETERRITORY	Manage Case Territories	/app/crm/common/automation/territorymanager.nl?case=T
LIST_CASETERRITORYASSIGN	Territory Reassignment	/app/common/bulk/bulkop.nl?searchtype=Case&opcode=ReAssign
LIST_CASETYPE	Case Types	/app/crm/support/casetypelist.nl
LIST_CCTRAN	View Credit Card Transactions	/app/accounting/transactions/cardtrans.nl
LIST_CHARGE	Charges	/app/accounting/transactions/billing/charges.nl
LIST_CHARGERULE	View Charge Rules	/app/accounting/transactions/billing/chargerules.nl
LIST_CHARGERUNRESULTS	Charge Run Results	/app/accounting/transactions/billing/chargerunresults.nl
LIST_CHART_ACCOUNT	Chart of Accounts	/app/accounting/account/accounts.nl?report=T&code=COA
LIST_CLASS	Classes	/app/common/otherlists/classlist.nl
LIST_CLASSSEGMENTMAPPING	Class Mapping	/app/accounting/account/classsegmentmappinglist.nl
LIST_COLORTHEME	Color Themes	/app/setup/look/colorthemes.nl
LIST_COMMERCECATALOG	Commerce Catalogs	/app/platform/services/commercecategory/commercecatalogs.nl

Task ID	Page Label in NetSuite	URL
LIST_COMMERCECATEGORY	Commerce Categories	/app/platform/services/commercecategory/commercecategories.nl
LIST_COMMISSIONSCHEDULE	Employee Schedules	/app/crm/sales/commissions/commissionscheds.nl
LIST_COMPANY	All Companies	/app/common/entity/companylist.nl
LIST_COMPETITOR	Competitors	/app/crm/sales/competitorlist.nl
LIST_CONSOLRATES	Consolidated Exchange Rates	/app/accounting/otherlists/consolidatedrates.nl
LIST_CONTACT	Contacts	/app/common/entity/contactlist.nl
LIST_COUNTRY	Set Up Countries	/app/setup/countries.nl
LIST_CRMGROUP	Groups	/app/crm/common/crmgrouplist.nl
LIST_CRMOTHERLIST	CRM Lists	/app/common/otherlists/crmotherlists.nl
LIST_CRMTEMPLATE	Marketing Templates	/app/crm/common/merge/marketingtemplates.nl
LIST_CSPSETUP	Setup CSP	/app/setup/CSPList.nl
LIST_CURRENCY	Currencies	/app/common/multicurrency/currencylist.nl
LIST_CURRENCYRATE	Currency Exchange Rates	/app/common/multicurrency/currencyratelist.nl
LIST_CUSTADDRESSENTRYFORM	Address Forms	/app/common/custom/custaddressentryforms.nl
LIST_CUSTBODYFIELD	Transaction Body Fields	/app/common/custom/bodycustfields.nl
LIST_CUSTCATEGORY	Center Categories	/app/common/custom/custcategories.nl
LIST_CUSTCENTER	Centers	/app/common/custom/custcenters.nl
LIST_CUSTCOLUMNFIELD	Transaction Column Fields	/app/common/custom/columncustfields.nl
LIST_CUSTEMAILLAYOUT	Transaction Form HTML Layouts	/app/common/custom/custemaillayouts.nl
LIST_CUSTENTITYFIELD	Entity Fields	/app/common/custom/entitycustfields.nl
LIST_CUSTENTRYFORM	Entry Forms	/app/common/custom/custentryforms.nl
LIST_CUSTEVENTFIELD	CRM Fields	/app/common/custom/eventcustfields.nl
LIST_CUSTFIELDTAB	Subtabs	/app/common/custom/custfieldtabs.nl
LIST_CUSTFORM	Transaction Forms	/app/common/custom/custforms.nl
LIST_CUSTITEMFIELD	Item Fields	/app/common/custom/itemcustfields.nl
LIST_CUSTITEMNUMBERFIELD	Item Number Fields	/app/common/custom/itemnumbercustfields.nl
LIST_CUSTJOB	Customers	/app/common/entity/custjoblist.nl?Customer_STAGE=CUSTOMER
LIST_CUSTLAYOUT	Transaction Form PDF Layouts	/app/common/custom/custlayouts.nl
LIST_CUSTLIST	Lists	/app/common/custom/custlists.nl

Task ID	Page Label in NetSuite	URL
LIST_CUSTOMCODEFILES	SuiteScripts	/app/common/media/mediaitemfolders.nl?folder=-15
LIST_CUSTOMERFIELDRULE	Set Up Sales Rules	/app/crm/sales/customerfieldrules.nl
LIST_CUSTOMERFORM	Online Customer Forms	/app/crm/sales/leadforms.nl
LIST_CUSTOMERSTATUS	Customer Statuses	/app/crm/sales/customerstatuslist.nl
LIST_CUSTOMSEGMENT	Custom Segments	/app/common/custom/segments/segments.nl
LIST_CUSTOMSUBLIST	Sublists	/app/common/custom/customsublists.nl
LIST_CUSTOTHERFIELD	Other Custom Fields	/app/common/custom/othercustfields.nl
LIST_CUSTRECORD	Record Types	/app/common/custom/custrecords.nl
LIST_CUSTRECORDFORM	Online Custom Record Forms	/app/common/custom/custrecordforms.nl
LIST_CUSTSECTION	Center Tabs	/app/common/custom/custsections.nl
LIST_CUSTTAB	Custom Subtabs	/app/common/custom/subtabs.nl
LIST_CUSTTRANSACTION	Transaction Types	/app/common/custom/customtransactions.nl
LIST_DEPARTMENT	Departments	/app/common/otherlists/departmentlist.nl
LIST_DEPTSEGMENTMAPPING	Department Mapping	/app/accounting/account/deptsegmentmappinglist.nl
LIST_DEVICE_ID	Device ID	/app/setup/deviceauth/devicelist.nl
LIST_DIFFROLE	Show Role Differences	/app/setup/diffroles.nl
LIST_DISTRIBUTIONNETWORK	Distribution Network	/app/accounting/inventory/distributionplanning/distributionnetworklist.nl
LIST_DRIVERSLICENSE	Driver's Licences	/app/hcm/hris/governmentid/driverslicenses.nl
LIST_DUPLICATE_RESOLUTION_STATUS	Duplicate Resolution Status	/app/common/entity/duplicatemanagement/dupejob.nl
LIST_EMAILTEMPLATE	Email Templates	/app/crm/common/merge/emailtemplates.nl
LIST_EMPLCATEGORY	Employee Directory	/app/site/setup/emplcategories.nl
LIST_EMPLOYEE_RECORD	Employees	/app/common/entity/employeelist.nl
LIST_EMPOTHERLIST	Employee Related Lists	/app/common/otherlists/empotherlists.nl
LIST_ENTITY	All Entities	/app/common/entity/entitylist.nl
LIST_ENTITYACCOUNTMAPPING	Entity Account Mappings	/app/accounting/account/entityaccountmappinglist.nl
LIST_ENTITY_DUPLICATES	Entity Duplicate Resolution	/app/common/entity/manageduplicates.nl
LIST_ESCALATIONRULE	Set Up Escalation Rules	/app/crm/support/escalationfieldrules.nl
LIST_ESCALATIONTERRITORY	Manage Escalation Assignment	/app/crm/common/automation/territorymanager.nl?escalation=T
LIST_EVENT	Events	/app/crm/calendar/eventlist.nl

Task ID	Page Label in NetSuite	URL
LIST_EXPCATEGORY	Expense Categories	/app/accounting/otherlists/expcategories.nl
LIST_FAIRVALUEDIMENSION	Fair Value Dimensions	/app/accounting/revrec/fairvaluedimension.nl
LIST_FAIRVALUEFORMULA	Fair Value Formulas	/app/accounting/revrec/fairvalueformulalist.nl
LIST_FAIRVALUEPRICE	Fair Value Price Lists	/app/accounting/revrec/fairvaluepricelist.nl
LIST_FAXTEMPLATE	Fax Templates	/app/crm/common/merge/faxtemplates.nl
LIST_FCSITEFOLDER	Web Site Hosting Files	/app/common/media/mediaitemfolders.nl?folder=-100
LIST_FCSITEFOLDERADV	Web Site Hosting Files	/app/common/media/mediaitemfolders.nl?folder=-100
LIST_FILECABINET	Media Items	/app/common/media/mediaitems.nl
LIST_FINCHRG	Assess Finance Charges	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustInvc&Transaction_FINCHRG=T
LIST_FISCALCALENDAR	Fiscal Calendars	/app/setup/period/fiscalcalendars.nl
LIST_FORECAST	List Sales Rep Forecast	/app/crm/sales/forecastlist.nl?Forecast_ISTEAM=F
LIST_GENERICRESOURCE	Generic Resources	/app/accounting/project/genericresources.nl
LIST_GIFTCERTIFICATES	Gift Certificates	/app/accounting/otherlists/giftcertificates.nl
LIST_GLNUMHISTORY	GL Audit Numbering History	/app/accounting/transactions/glnumbering/glnumhistory.nl
LIST_GLOBALACCOUNTMAPPING	Global Account Mappings	/app/accounting/account/globalaccountmappinglist.nl
LIST_GOVERNMENTISSUEDIDTYPE	Government-Issued ID Types	/app/hcm/hris/governmentid/governmentissuedidtypes.nl
LIST_HCMJOB	Jobs	/app/hcm/hris/jobs.nl
LIST_IMAGE	Images	/app/common/media/mediaitemfolders.nl?folder=-4
LIST_INFOITEM	Information Items	/app/site/setup/infoitemlist.nl?searchid=-2540
LIST_INFOITEMFORM	Publish Forms	/app/site/setup/infoitemlist.nl?searchid=-2541
LIST_INSTALLEDBUNDLE	Installed Bundles	/app/bundler/bundlelist.nl?type=I
LIST_INVCOSTTEMPLATE	Inventory Cost Template	/app/accounting/inventory/standard/invcosttemplatelist.nl
LIST_ISSUE	Issues	/app/crm/support/issuedb/issuelist.nl
LIST_ITEM	Items	/app/common/item/itemlist.nl
LIST_ITEMACCOUNTMAPPING	Item Account Mappings	/app/accounting/account/itemaccountmappinglist.nl
LIST_ITEMATTRGROUP	Field Sets	/app/site/setup/fieldsets.nl

Task ID	Page Label in NetSuite	URL
LIST_ITEMDEMANDPLAN	Item Demand Plans	/app/accounting/inventory/demandplanning/itemdemandplanlist.nl
LIST_ITEMOPTION	Transaction Item Options	/app/common/custom/itemoptions.nl
LIST_ITEMREVENUECATEGORY	Item Revenue Categories	/app/accounting/revrec/itemrevenuecategorylist.nl
LIST_ITEMSUPPLYPLAN	Item Supply Plans	/app/accounting/inventory/demandplanning/itemsupplyplanlist.nl
LIST_ITEM_REVISION	Item Revisions	/app/common/item/itemrevisionlist.nl
LIST_JOB	Jobs	/app/accounting/project/projects.nl
LIST_JOBREQUISITION	Job Requisitions	/app/hcm/hris/jobrequisitions.nl
LIST_KBCATEGORY	Knowledge Base	/app/site/setup/kbcategories.nl
LIST_KPIREPORT	KPI Scorecards	/app/center/enhanced/kpireports.nl
LIST_KUDOS	Kudos	/app/hcm/perfmgmt/kudoslist.nl
LIST_LEAD	Leads	/app/common/entity/custjoblist.nl?Customer_STAGE=LEAD
LIST_LOCASSIGNCONF	Configurations List	/app/setup/locationassignment/configuration/locationassignmentconfigurationlist.nl
LIST_LOCATION	Locations	/app/common/otherlists/locationlist.nl
LIST_LOCATIONCOSTINGGROUP	Location Costing Groups	/app/accounting/inventory/costing/locationcostinggrouplist.nl
LIST_LOCSEGMENTMAPPING	Location Mapping	/app/accounting/account/locsegmentmappinglist.nl
LIST_MAILMERGE	Merge History	/app/crm/common/merge/mailmergehistory.nl
LIST_MAILTEMPLATE	Letter Templates	/app/crm/common/merge/mailtemplates.nl
LIST_MAPREDUCESCRIPTSTATUS	Map/Reduce Script Status	/app/common/scripting/mapreducescriptstatus.nl
LIST_MEDIAITEMFOLDER	File Cabinet	/app/common/media/mediaitemfolders.nl
LIST_MEDIAITEMFOLDER_LOG	Job Status	/app/external/xml/upload/uploadlog.nl?displayType=FILECABINET
LIST_MEMDOC	Memorized Transactions	/app/accounting/otherlists/memdoclist.nl
LIST_MEMDOCRESULTS	Memorized Transactions Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=MEMORIZEDTRANSACTIONS&BulkProcSubmission_CREATEDDATE=TODAY
LIST_MFGCOSTTEMPLATE	Manufacturing Cost Templates	/app/accounting/manufacturing/mfgcosttemplatelist.nl
LIST_MFGOPERATIONTASK	Manufacturing Operation Tasks	/app/accounting/manufacturing/mfgoperationlist.nl
LIST_MFGROUTING	Manufacturing Routing	/app/accounting/manufacturing/mfgroutinglist.nl

Task ID	Page Label in NetSuite	URL
LIST_MGRFORECAST	List Sales Manager Forecast	/app/crm/sales/forecastlist.nl?Forecast_ISTEAM=T
LIST_MYROLES	My Roles	/app/center/myroles.nl
LIST_NOTIFICATION	Notifications	/app/email/alerts/notifications.nl
LIST_ORGANIZATIONVALUE	Organization Values	/app/hcm/perfmgmt/organizationvaluelist.nl
LIST_OTHERGOVERNMENTISSUEDID	Other Government-Issued IDs	/app/hcm/hris/governmentid/othergovernmentissuedids.nl
LIST_OTHERNAME	Other Names	/app/common/entity/othernames.nl
LIST_PARTNER	Partners	/app/common/entity/partnerlist.nl
LIST_PARTNERCOMMISSIONSCHED	Partner Schedules	/app/crm/sales/commissions/partnercommissionscheds.nl
LIST_PARTNERPLANASSIGN	Partner Plans	/app/crm/sales/commissions/partnercommissionplans.nl
LIST_PARTNERPLANASSIGNS	Partner Plan Assignments	/app/crm/sales/commissions/partnerplanassigns.nl
LIST_PASSPORT	Passports	/app/hcm/hris/governmentid/passports.nl
LIST_PAYMENTEVENT	View Payment Events	/app/paymentprocessing/paymentevents.nl
LIST_PAYMETHODS	Payment Methods	/app/common/otherlists/accountingotherlists.nl?tname=paymeth
LIST_PAYPALACCOUNT	PayPal Accounts	/app/external/paypal/paypalaccountlist.nl
LIST_PAYROLLBATCH	Payroll Batches	/app/payroll/payrollbatchlist2.nl
LIST_PAYROLLISSUES	Payroll Issues	/app/payroll/cdstatus.nl
LIST_PAYROLLITEM	Payroll Items	/app/common/item/payrollitems.nl
LIST_PAYTERMS	Payment Terms	/app/common/otherlists/accountingotherlists.nl?tname=terms
LIST_PDFTEMPLATE	PDF Templates	/app/crm/common/merge/pdftemplates.nl
LIST_PERIOD	Manage Accounting Periods	/app/setup/period/fiscalperiods.nl
LIST_PLANASSIGN	Employee Plans	/app/crm/sales/commissions/commissionplans.nl
LIST_PLANASSIGNS	Employee Plan Assignments	/app/crm/sales/commissions/planassigns.nl
LIST_PLANNEDSTANDARDCOST	Planned Standard Costs	/app/accounting/inventory/plannedstandardcostlist.nl
LIST_PLUGIN	Plug-in Implementations	/app/common/scripting/pluginlist.nl
LIST_PLUGINTYPE	Custom Plug-in Types	/app/common/scripting/plugintypelist.nl
LIST_POSITION	Positions	/app/hcm/hris/positions.nl
LIST_PRESCATEGORY	Categories	/app/site/setup/prescategories.nl?ctype=PRES&section=&siteid=
LIST_PROCESSHISTTXNSTATUS	Historical Transaction Processing Status	/app/accounting/multibook/htp/status.nl

Task ID	Page Label in NetSuite	URL
LIST_PROJECTEXPENSETYPE	Project Expense Types	/app/accounting/otherlists/projectexpensetypes.nl
LIST_PROJECTTASK	Project Tasks	/app/accounting/project/projecttasks.nl
LIST_PROJECTTASKIMPORT	Import Project Tasks	/app/setup/assistants/nsimport/simpleimport.nl?rectype=PROJECTTASK
LIST_PROJECTTEMPLATE	Project Templates	/app/accounting/project/projecttemplates.nl
LIST_PROSPECT	Prospects	/app/common/entity/custjoblist.nl?Customer_STAGE=PROSPECT
LIST_PUBLISHERAPP	List Applications	/app/suiteapp/devframework/appregistrylist.nl
LIST_QUANTITYPRICINGSCHEDULE	Quantity Pricing Schedules	/app/accounting/otherlists/quantitypricingschedules.nl
LIST_QUOTA	Quotas	/app/crm/sales/quotalist.nl
LIST_RATEPLAN	Rate Plans	/app/accounting/otherlists/rateplans.nl
LIST_RECENTRECORDS	Recent Records	/app/common/otherlists/recentrecords.nl
LIST_RECVDFILES	Attachments Received	/app/common/media/mediaitemfolders.nl?folder=-10
LIST_REFERRALCODE	Promotions	/app/crm/sales/referralcodelist.nl
LIST_REGION	Set Up Regions	/app/setup/region/regionlist.nl
LIST_RELATEDITEM	Related Items Categories	/app/site/setup/relateditems.nl
LIST_RELATEDITEMADV	Related Items Categories	/app/site/setup/relateditems.nl
LIST_REPORESULT	Report Results	/app/reporting/workqueue/reportresultlist.nl
LIST_REPOSCHEDULE	Report Schedules	/app/reporting/workqueue/reportschedulelist.nl
LIST_RESOURCE	Resources	/app/crm/calendar/resources.nl
LIST_REVENUEALLOCATIONGROUP	Revenue Allocation Groups	/app/common/otherlists/accountingotherlists.nl?tname=revenueallocationgroup
LIST_REVENUEELEMENT	Revenue Elements	/app/accounting/revrec/revenueelementlist.nl
LIST_REVENUEPLAN	Revenue Recognition Plans	/app/accounting/revrec/revenueplanlist.nl
LIST_REVENUERECOGNITIONRULE	Revenue Recognition Rules	/app/accounting/revrec/revenuerecognitionrulelist.nl
LIST_REVRECFIELDMAPPING	Revenue Field Mapping	/app/accounting/revrec/revrecfieldmapping.nl
LIST_REVRECSCHEDS	Revenue Recognition Schedules	/app/accounting/otherlists/revrecschedules.nl
LIST_REVRECSCHEDULE	Revenue Recognition Templates	/app/accounting/otherlists/revrectemplates.nl
LIST_ROLE	Manage Roles	/app/setup/rolelist.nl

Task ID	Page Label in NetSuite	URL
LIST_RSRCALLOCATION	Rsrc Allocations	/app/accounting/project/allocations.nl
LIST_RSSFEED	RSS Feeds	/app/site/hosting/rssfeeds.nl
LIST_SALESCAMPAIGN	Sales Campaigns	/app/crm/marketing/salescampaignlist.nl
LIST_SALESTEAM	Sales Teams	/app/crm/common/salesteamlist.nl
LIST_SALESTERRITORIES	Sales Territory List	/app/crm/sales/salesterritorylist.nl
LIST_SALESTERRITORY	Manage Sales Territori es	/app/crm/common/automation/territory manager.nl?sales=T
LIST_SALESTERRITORYASSIGN	Territory Reassignment	/app/common/bulk/bulkop.nl?searchtype=C ustomer&opcode=ReAssign
LIST_SAVEDASHBOARD	Published Dashboards	/app/center/setup/savedashboards.nl
LIST_SAVEDBULKOP	Saved Mass Updates	/app/common/bulk/savedbulkops.nl
LIST_SAVEDSEARCH	Saved Searches	/app/common/search/savedsearchlist.nl
LIST_SCRIPT	Scripts	/app/common/scripting/scriptlist.nl
LIST_SCRIPTEDRECORD	Scripted Records	/app/common/scripting/scriptedrecords.nl
LIST_SCRIPTLOGS	Script Execution Logs	/app/common/scripting/scriptnotearchive.nl
LIST_SCRIPTRECORD	Script Deployments	/app/common/scripting/scriptrecordlist.nl
LIST_SCRIPTSTATUS	Scheduled Script Status	/app/common/scripting/scriptstatus.nl
LIST_SEARCHRESULTS	Search Results	/app/common/search/searchresults.nl
LIST_SEARCHRESULTSARCHIVE	Search Results	/app/common/search/backend/searchresult sarchive.nl
LIST_SENTFILES	Attachments Sent	/app/common/media/mediaitemfolders.nl?f older=-14
LIST_SHIPITEM	Shipping Items	/app/common/item/shipitems.nl
LIST_SHIPPARTREGISTRATION	Shipping Partner Regist rations	/app/setup/shipping.nl?tab=partnerregistra tions
LIST_SHIPPINGMANIFEST	Shipping Manifest	/app/common/shipping/fedex/shippingmani fest.nl
LIST_SITEEMAILTEMPLATE	Web Store Email Templa tes	/app/site/setup/siteemailtemplates.nl
LIST_SITEITEMTEMPLAT	Item/Category Templa tes	/app/site/setup/siteitemtemplates.nl
LIST_SITETAGS	Web Site Tags	/app/site/setup/sitetags.nl
LIST_SITETHEMES	Web Site Themes	/app/site/setup/sitethemes.nl
LIST_SMBIMPORT	SMB Import	/app/external/xml/upload/upload.nl
LIST_SMTP_SERVERS	SMTP Servers	/app/setup/smtpservers.nl
LIST_SOLUTION	Solutions	/app/crm/support/kb/solutions.nl
LIST_SSCATEGORY	Publish Saved Search	/app/site/setup/sscategories.nl
LIST_STANDARDCOSTVERSION	Standard Cost Versions	/app/accounting/inventory/standardcostvers ionlist.nl

Task ID	Page Label in NetSuite	URL
LIST_STATE	Set Up States/Provinces/Counties	/app/setup/states.nl
LIST_STATUSACH	View Direct Deposit Status	/app/accounting/transactions/statusach.nl
LIST_STATUSEFT	View Electronic Funds Transfer Status	/app/accounting/transactions/statuseft.nl
LIST_STATUSVP	View Vendor Payment Status	/app/accounting/transactions/statusvp.nl
LIST_STOREITEMLISTLAYOUT	Layouts	/app/site/setup/storeitemlistlayouts.nl
LIST_STORETAB	Tabs	/app/site/setup/storetabs.nl
LIST_SUBSCRIPTION	Subscriptions	/app/accounting/subscription/subscriptionlist.nl
LIST_SUBSCRIPTIONCHANGEORDER	View Subscription Change Orders	/app/accounting/subscription/subscriptionchangeorders.nl
LIST_SUBSCRIPTIONCHANGEORDER2	Subscription Change Orders	/app/accounting/subscription/subscriptionchangeorders.nl
LIST_SUBSCRIPTIONLINE	Subscription Lines	/app/accounting/subscription/subscriptionlinelist.nl
LIST_SUBSCRIPTIONPLAN	Subscription Plans	/app/accounting/subscription/subscriptionplans.nl
LIST_SUBSCRIPTIONRENEWALHISTORY	View Subscription Renewal History	/app/accounting/subscription/autorenewal/subscriptionrenewalhistory.nl
LIST_SUBSCRIPTIONRENEWALHISTORY2	Subscription Renewal History	/app/accounting/subscription/autorenewal/subscriptionrenewalhistory.nl
LIST_SUBSIDIARY	Subsidiaries	/app/common/otherlists/subsidiarylist.nl
LIST_SUPPORTCASE	Cases	/app/crm/support/caselist.nl
LIST_SYSALERT	System Alerts	/app/common/otherlists/systemalerts.nl
LIST_SYSTEMEMAILTEMPLATE	Set Up System Email Templates	/app/crm/common/merge/systememailtemplates.nl
LIST_TASK	Tasks	/app/crm/calendar/tasklist.nl
LIST_TAXGROUP	Tax Groups	/app/common/item/taxgroups.nl
LIST_TAXITEM	Tax Codes	/app/common/item/taxitems.nl
LIST_TAXPERIOD	Manage Tax Periods	/app/setup/period/taxperiods.nl
LIST_TAXSCHEDULE	Tax Schedules	/app/common/item/taxschedules.nl
LIST_TEMPLATEFILES	Template Files	/app/common/media/mediaitemfolders.nl?folder=-9
LIST_TEMPLATE_CATEGORY	Template Categories	/app/crm/common/merge/templatecategories.nl
LIST_TERMINATIONREASON	Termination Reasons	/app/hcm/hris/terminationreasons.nl
LIST_TIMEBILL_WEEKLY	List Weekly Time Tracking	/app/accounting/transactions/weeklytimelist.nl

Task ID	Page Label in NetSuite	URL
LIST_TIMEOFFCHANGE	Time-Off Changes	/app/hcm/hris/timeoffmanagement/timeoffchanges.nl
LIST_TIMEOFFPLAN	Time-Off Plans	/app/hcm/hris/timeoffmanagement/timeoffplans.nl
LIST_TIMEOFFREQUEST	Time-Off Requests	/app/hcm/hris/timeoffmanagement/timeoffrequests.nl
LIST_TIMEOFFTYPE	Time-Off Types	/app/hcm/hris/timeoffmanagement/timeofftypes.nl
LIST_TIMESHEET	Timesheets	/app/accounting/transactions/timesheets.nl
LIST_TOPIC	Topics	/app/crm/support/kb/topics.nl
LIST_TRANNUMBERAUDITLOG	Transaction Numbering Audit Log	/app/accounting/transactions/trannumberauditlist.nl
LIST_TRANSACTION	Transactions	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=@ALL@
LIST_TRAN_BINTRNFR	Bin Transfers	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=BinTrnfr
LIST_TRAN_BINWKSHT	Bin Putaway Worksheets	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=BinWksht
LIST_TRAN_BLANKORD	Blanket Purchase Orders	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=BlankOrd
LIST_TRAN_BOOKICJOURNAL	Accounting Book Intercompany Journal Entries	/app/accounting/transactions/transactionliststwitch.nl?TR_Transaction_TYPE=Journal&icj=T
LIST_TRAN_BOOKJOURNAL	Accounting Book Journal Entries	/app/accounting/transactions/transactionliststwitch.nl?TR_Transaction_TYPE=Journal
LIST_TRAN_BUILD	Assembly Builds	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Build
LIST_TRAN_CARDCHRG	Credit Card Charges	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CardChrg
LIST_TRAN_CASHRFND	Cash Refunds	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CashRfnd
LIST_TRAN_CASHSALE	Cash Sales	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CashSale
LIST_TRAN_CHECK	Checks	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Check
LIST_TRAN_COMMISSN	Individual Employee Commissions	/app/accounting/transactions/commissns.nl
LIST_TRAN_CUSTCHRG	Statement Charges	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustChrg
LIST_TRAN_CUSTCRED	Credit Memos	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustCred
LIST_TRAN_CUSTDEP	Customer Deposits	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustDep
LIST_TRAN_CUSTINVC	Invoices	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustInvc

Task ID	Page Label in NetSuite	URL
LIST_TRAN_CUSTPYMT	Customer Payments	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustPymt
LIST_TRAN_CUSTRFND	Customer Refunds	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustRfnd
LIST_TRAN_DEPAPPL	Deposit Applications	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=DepAppl
LIST_TRAN_DEPOSIT	Deposits	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Deposit
LIST_TRAN_DOWNLOAD	My Downloads	/app/common/entity/downloads.nl
LIST_TRAN_ESTIMATE	Estimates	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Estimate
LIST_TRAN_EXPREPT	Expense Reports	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=ExpRept
LIST_TRAN_FFTREQ	Manage Fulfillment Requests	/app/accounting/transactions/fulfillmentrequest/fftreqlist.nl?Transaction_TYPE=FftReq
LIST_TRAN_FXREVAL	Currency Revaluations	/app/accounting/transactions/transactionliststwitch.nl?TR_Transaction_TYPE=FxReval
LIST_TRAN_ICJOURNAL	Intercompany Journal Entries	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Journal&icj=T
LIST_TRAN_ICTRNFRORD	Intercompany Transfer Orders	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TrnfrOrd&icto=T
LIST_TRAN_INVADJST	Inventory Adjustments	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=InvAdjst
LIST_TRAN_INVCOUNT	Inventory Counts	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=InvCount
LIST_TRAN_INVDISTR	Inventory Distributions	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=InvDistr
LIST_TRAN_INVREVAL	Inventory Cost Revaluations	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=InvReval
LIST_TRAN_INVTRNFR	Inventory Transfers	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=InvTrnfr
LIST_TRAN_INVWKSHT	Inventory Worksheets	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=InvWksht
LIST_TRAN_ITEMRCPT	Item Receipts	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=ItemRcpt
LIST_TRAN_ITEMSHIP	Item Fulfillments	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=ItemShip
LIST_TRAN_JOURNAL	Journal Entries	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Journal
LIST_TRAN_LIABPYMT	Liability Payments	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=LiabPymt
LIST_TRAN_OPPRTNTY	Opportunities	/app/accounting/transactions/opprtntylist.nl
LIST_TRAN_PARTNERCOMMISSN	Individual Partner Commissions	/app/accounting/transactions/partnercommissns.nl

Task ID	Page Label in NetSuite	URL
LIST_TRAN_PAYCHECK	Paychecks	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Paycheck
LIST_TRAN_PCHKJRNL	Paycheck Journals	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=PChkJrnl
LIST_TRAN_PURCHCON	Purchase Contracts	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=PurchCon
LIST_TRAN_PURCHORD	Purchase Orders	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=PurchOrd
LIST_TRAN_PURCHORD_REQ	View Purchase Requests/Orders	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=PurchOrd
LIST_TRAN_PURCHREQ	Requisitions	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=PurchReq
LIST_TRAN_REORDER	Items Ordered	/app/common/item/itemsordered.nl
LIST_TRAN_REVARRNG	Revenue Arrangements	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=RevArrng
LIST_TRAN_REVCOMM	Revenue Commitments	/app/accounting/transactions/transactionlistswitch.nl?TR_Transaction_TYPE=RevComm
LIST_TRAN_REVCOMRV	View Revenue Commitment Reversals	/app/accounting/transactions/transactionlistswitch.nl?TR_Transaction_TYPE=RevComRv
LIST_TRAN_REVCONTR	View Revenue Contracts	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=RevContr
LIST_TRAN_RFQ	Requests For Quote	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Rfq
LIST_TRAN_RTNAUTH	Return Authorizations	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=RtnAuth
LIST_TRAN_SALESORD	Sales Orders	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=SalesOrd
LIST_TRAN_STATJOURNAL	Statistical Journal Entries	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Journal&statistical=T
LIST_TRAN_STATSCHEDULE	Statistical Schedules	/app/accounting/transactions/statistical/statisticalschedulelist.nl
LIST_TRAN_STPICKUP	Store Pickup Fulfillments	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=StPickUp
LIST_TRAN_TAXLIAB	Tax Liabilities	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TaxLiab
LIST_TRAN_TAXPYMT	Tax Payments	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TaxPymt
LIST_TRAN_TEGPYBL	Issued Tegatas	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TegPybl
LIST_TRAN_TEGRCVBL	Received Tegatas	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TegRcvbl
LIST_TRAN_TRANSFER	Bank Transfers	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Transfer
LIST_TRAN_TRNFRORD	Transfer Orders	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TrnfrOrd

Task ID	Page Label in NetSuite	URL
LIST_TRAN_UNBUILD	Assembly Unbuilds	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Unbuild
LIST_TRAN_VATLIAB	Tax Liabilities	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TaxLiab
LIST_TRAN_VATLIABAU	GST Liabilities	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TaxLiab
LIST_TRAN_VATLIABUK	VAT Liabilities	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TaxLiab
LIST_TRAN_VENDAUTH	Vendor Return Authorizations	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=VendAuth
LIST_TRAN_VENDBILL	Bills	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=VendBill
LIST_TRAN_VENDCRED	Bill Credits	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=VendCred
LIST_TRAN_VENDPYMT	Bill Payments	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=VendPymt
LIST_TRAN_VENDRFQ	Vendor Requests For Quote	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=VendRfq
LIST_TRAN_WOCLOSE	Work Order Closes	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=WOClose
LIST_TRAN_WOCOMPL	Work Order Completions	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=WOCompl
LIST_TRAN_WOISSUE	Work Order Issues	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=WOIssue
LIST_TRAN_WORKORD	Work Orders	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=WorkOrd
LIST_TRAN_YTDADJST	Payroll Adjustment	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=YtdAdjst
LIST_UNCATSITEITEM	View Uncategorized Items	/app/setup/uncatsiteitems.nl
LIST_UNCATSITEITEMADV	View Uncategorized Items	/app/setup/uncatsiteitems.nl
LIST_UNITSTYPE	Units Of Measure	/app/common/units/unitstypelist.nl
LIST_UPSELLPOPUP	Upsell Popup	/app/crm/sales/upsell/upsellpopup.nl
LIST_UPSELLWIZARD	Upsell Manager	/app/crm/sales/upsell/upsellmanager.nl
LIST_URLALIASES	Promotional URLs	/app/setup/urlaliases.nl
LIST_USAGE	Usages	/app/accounting/otherlists/usages.nl
LIST_USER	Manage Users	/app/setup/listusers.nl
LIST_VENDOR	Vendors	/app/common/entity/vendorlist.nl
LIST_WEBAPPS	SSP Applications	/app/common/scripting/webapplist.nl
LIST_WEBAPPSADV	SSP Applications	/app/common/scripting/webapplist.nl
LIST_WORKCALENDAR	Work Calendars	/app/accounting/project/workcalendars.nl

Task ID	Page Label in NetSuite	URL
LIST_WORKFLOW	Workflows	/app/common/workflow/setup/workflowlist.nl
LIST_WORKPLACE	Workplaces	/app/common/otherlists/workplacelist.nl
REPO_10	ALT_SALES Pipeline by Sales Rep Detail	/app/reporting/reportrunner.nl?cr=10
REPO_100	Total Pipeline by Sales Rep	/app/reporting/reportrunner.nl?cr=100
REPO_101	Total Pipeline by Sales Rep Detail	/app/reporting/reportrunner.nl?cr=101
REPO_102	New Customer Sales Detail	/app/reporting/reportrunner.nl?cr=102
REPO_103	New Customer Sales	/app/reporting/reportrunner.nl?cr=103
REPO_104	Opportunities Lost	/app/reporting/reportrunner.nl?cr=104
REPO_105	Opportunities Won	/app/reporting/reportrunner.nl?cr=105
REPO_106	Sales Activity by Customer	/app/reporting/reportrunner.nl?cr=106
REPO_107	Sales Activity by Customer Detail	/app/reporting/reportrunner.nl?cr=107
REPO_108	Comparative Sales	/app/reporting/reportrunner.nl?cr=108
REPO_109	Income Statement	/app/reporting/reportrunner.nl?cr=109
REPO_11	ALT_SALES Pipeline by Sales Rep Summary	/app/reporting/reportrunner.nl?cr=11
REPO_110	Income Statement	/app/reporting/reportrunner.nl?cr=110
REPO_111	Authorized Commission	/app/reporting/reportrunner.nl?cr=111
REPO_112	Authorized Commission Detail	/app/reporting/reportrunner.nl?cr=112
REPO_113	GST on Purchases	/app/reporting/reportrunner.nl?cr=113
REPO_114	GST on Sales	/app/reporting/reportrunner.nl?cr=114
REPO_115	GST on Purchases Detail	/app/reporting/reportrunner.nl?cr=115
REPO_116	GST on Sales Detail	/app/reporting/reportrunner.nl?cr=116
REPO_118	Commissions Pending Authorization	/app/reporting/reportrunner.nl?cr=118
REPO_119	Commissions Pending Authorization Detail	/app/reporting/reportrunner.nl?cr=119
REPO_12	ALT_SALES Forecast by Sales Rep Detail	/app/reporting/reportrunner.nl?cr=12
REPO_121	Estimated Commission	/app/reporting/reportrunner.nl?cr=121
REPO_122	Estimated Commission Detail	/app/reporting/reportrunner.nl?cr=122
REPO_123	Campaign Response Detail	/app/reporting/reportrunner.nl?cr=123
REPO_126	Pipeline by Customer	/app/reporting/reportrunner.nl?cr=126

Task ID	Page Label in NetSuite	URL
REPO_127	Pipeline by Customer Detail	/app/reporting/reportrunner.nl?cr=127
REPO_128	Pipeline by Sales Rep	/app/reporting/reportrunner.nl?cr=128
REPO_129	Pipeline ROI by Manager	/app/reporting/reportrunner.nl?cr=129
REPO_13	ALT_SALES Forecast by Sales Rep	/app/reporting/reportrunner.nl?cr=13
REPO_130	Pipeline by Sales Rep Detail	/app/reporting/reportrunner.nl?cr=130
REPO_131	Pipeline by Status Summary	/app/reporting/reportrunner.nl?cr=131
REPO_132	Campaign ROI Analysis Detail	/app/reporting/reportrunner.nl?cr=132
REPO_133	Campaign Response	/app/reporting/reportrunner.nl?cr=133
REPO_134	Sales Activity by Sales Rep	/app/reporting/reportrunner.nl?cr=134
REPO_135	Sales Activity by Sales Rep Detail	/app/reporting/reportrunner.nl?cr=135
REPO_14	Billings Forecast vs. Quota	/app/reporting/reportrunner.nl?cr=14
REPO_141	Partner Activity	/app/reporting/reportrunner.nl?cr=141
REPO_142	Partner Activity Detail	/app/reporting/reportrunner.nl?cr=142
REPO_143	Sales by Partner	/app/reporting/reportrunner.nl?cr=143
REPO_144	Sales by Partner Detail	/app/reporting/reportrunner.nl?cr=144
REPO_145	Shipping Report	/app/reporting/reportrunner.nl?cr=145
REPO_146	Total Open Opportunities Detail	/app/reporting/reportrunner.nl?cr=146
REPO_147	Forecast vs. Quota	/app/reporting/reportrunner.nl?cr=147
REPO_148	Sales Orders Pending Fulfillment	/app/reporting/reportrunner.nl?cr=148
REPO_149	Forecast by Sales Rep Detail	/app/reporting/reportrunner.nl?cr=149
REPO_15	ALT_SALES and Billings Forecast vs. Quota	/app/reporting/reportrunner.nl?cr=15
REPO_150	Inventory Back Order Report	/app/reporting/reportrunner.nl?cr=150
REPO_151	Inventory Activity Detail	/app/reporting/reportrunner.nl?cr=151
REPO_152	Realized Exchange Rate Gains and Losses	/app/reporting/reportrunner.nl?cr=152
REPO_16	ALT_SALES Forecast vs. Quota	/app/reporting/reportrunner.nl?cr=16
REPO_161	Customers by Partner Detail	/app/reporting/reportrunner.nl?cr=161

Task ID	Page Label in NetSuite	URL
REPO_162	Customers by Partner	/app/reporting/reportrunner.nl?cr=162
REPO_163	Closed Cases Snapshot	/app/reporting/reportrunner.nl?cr=163
REPO_164	New Cases Snapshot	/app/reporting/reportrunner.nl?cr=164
REPO_165	Open Cases Snapshot	/app/reporting/reportrunner.nl?cr=165
REPO_169	Closed Case Analysis Detail	/app/reporting/reportrunner.nl?cr=169
REPO_17	Gross Lead Source Ana lysis Detail	/app/reporting/reportrunner.nl?cr=17
REPO_170	Closed Case Analysis	/app/reporting/reportrunner.nl?cr=170
REPO_171	Open Case Analysis Det ail	/app/reporting/reportrunner.nl?open=T&c r=171
REPO_172	Open Case Analysis	/app/reporting/reportrunner.nl?open=T&c r=172
REPO_173	Customers by Sales Rep Detail	/app/reporting/reportrunner.nl?cr=173
REPO_174	Customers by Sales Rep	/app/reporting/reportrunner.nl?cr=174
REPO_175	Customers by Territory Detail	/app/reporting/reportrunner.nl?cr=175
REPO_176	Customers by Territory	/app/reporting/reportrunner.nl?cr=176
REPO_177	Lead Source Analysis Detail	/app/reporting/reportrunner.nl?cr=177
REPO_178	Lead Source Analysis	/app/reporting/reportrunner.nl?cr=178
REPO_179	Sales Orders Register	/app/reporting/reportrunner.nl?cr=179
REPO_18	Gross Lead Source Ana lysis	/app/reporting/reportrunner.nl?cr=18
REPO_180	Open Sales Orders	/app/reporting/reportrunner.nl?cr=180
REPO_183	Shopping Cart Abando nment	/app/reporting/reportrunner.nl?cr=183
REPO_184	Shopping Activity Analys is by Category	/app/reporting/reportrunner.nl?cr=184
REPO_185	Shopping Activity Analys is	/app/reporting/reportrunner.nl?cr=185
REPO_188	Item Page Views and Sal es Summary	/app/reporting/reportrunner.nl?cr=188
REPO_189	Hosted Page Hits Detail	/app/reporting/reportrunner.nl?cr=189
REPO_19	A/P Payment History by Payment	/app/reporting/reportrunner.nl?cr=19
REPO_191	Hosted Page Hits	/app/reporting/reportrunner.nl?cr=191
REPO_192	Item Orders Detail	/app/reporting/reportrunner.nl?cr=192
REPO_193	Item Orders by Catego ry	/app/reporting/reportrunner.nl?cr=193
REPO_194	Item Orders	/app/reporting/reportrunner.nl?cr=194

Task ID	Page Label in NetSuite	URL
REPO_195	Page Hits Detail	/app/reporting/reportrunner.nl?cr=195
REPO_196	Page Hits by Category	/app/reporting/reportrunner.nl?cr=196
REPO_197	Page Hits	/app/reporting/reportrunner.nl?cr=197
REPO_198	Sales by Promotion Det ail	/app/reporting/reportrunner.nl?subtype=r&c r=198
REPO_199	Sales by Promotion	/app/reporting/reportrunner.nl?subtype=r&c r=199
REPO_2	Deferred / Capitalized Expense	/app/reporting/reportrunner.nl?cr=2
REPO_20	A/P Payment History by Bill	/app/reporting/reportrunner.nl?cr=20
REPO_203	Prospect Analysis	/app/reporting/reportrunner.nl?cr=203
REPO_204	Prospect Analysis Detail	/app/reporting/reportrunner.nl?cr=204
REPO_205	Partner Commissions Pending Authorization	/app/reporting/reportrunner.nl?cr=205
REPO_206	Partner Commissions Pending Authorization Detail	/app/reporting/reportrunner.nl?cr=206
REPO_209	Unbilled Cost by Cus tomer Detail	/app/reporting/reportrunner.nl?cr=209
REPO_21	A/R Payment History by Payment	/app/reporting/reportrunner.nl?cr=21
REPO_210	Unbilled Time by Cus tomer Detail	/app/reporting/reportrunner.nl?cr=210
REPO_211	Unbilled Time by Cus tomer	/app/reporting/reportrunner.nl?cr=211
REPO_214	Sales Tax Transaction Detail	/app/reporting/reportrunner.nl?cr=214
REPO_215	Sales Tax Liability By Tax Item	/app/reporting/reportrunner.nl?cr=215
REPO_218	Time by Item Detail	/app/reporting/reportrunner.nl?cr=218
REPO_219	Time by Customer Det ail	/app/reporting/reportrunner.nl?cr=219
REPO_22	A/R Payment History by Invoice	/app/reporting/reportrunner.nl?cr=22
REPO_220	Time by Employee Det ail	/app/reporting/reportrunner.nl?cr=220
REPO_221	Time by Item	/app/reporting/reportrunner.nl?cr=221
REPO_222	Time by Customer	/app/reporting/reportrunner.nl?cr=222
REPO_223	Time by Employee	/app/reporting/reportrunner.nl?cr=223
REPO_225	Unbilled Cost by Cus tomer	/app/reporting/reportrunner.nl?cr=225

Task ID	Page Label in NetSuite	URL
REPO_226	Inventory Revenue Detail	/app/reporting/reportrunner.nl?cr=226
REPO_227	Inventory Valuation Detail	/app/reporting/reportrunner.nl?cr=227
REPO_229	Inventory Revenue	/app/reporting/reportrunner.nl?cr=229
REPO_23	Comparative Sales (ALT_SALES)	/app/reporting/reportrunner.nl?cr=23
REPO_230	Inventory Valuation	/app/reporting/reportrunner.nl?cr=230
REPO_231	Inventory Profitability	/app/reporting/reportrunner.nl?cr=231
REPO_232	Payroll Liability	/app/reporting/reportrunner.nl?payitemtype=&cr=232
REPO_233	Payroll Journal	/app/reporting/reportrunner.nl?payitemtype=&cr=233
REPO_234	Payroll Summary by Employee	/app/reporting/reportrunner.nl?cr=-176
REPO_235	Payroll Summary	/app/reporting/reportrunner.nl?payitemtype=&cr=235
REPO_239	Item Transaction Detail	/app/reporting/reportrunner.nl?cr=239
REPO_24	Comparative Sales (Orders)	/app/reporting/reportrunner.nl?cr=24
REPO_240	Sales Back Order Report	/app/reporting/reportrunner.nl?cr=240
REPO_241	Physical Inventory Worksheet	/app/reporting/reportrunner.nl?cr=241
REPO_242	Payroll Check Register	/app/reporting/reportrunner.nl?cr=242
REPO_243	Current Inventory Snapshot	/app/reporting/reportrunner.nl?cr=243
REPO_246	Items Pending Fulfillment	/app/reporting/reportrunner.nl?cr=246
REPO_248	Cost by Customer Detail	/app/reporting/reportrunner.nl?cr=248
REPO_249	Estimated Partner Commission	/app/reporting/reportrunner.nl?cr=249
REPO_25	New Customer Sales Orders Detail	/app/reporting/reportrunner.nl?cr=25
REPO_250	Transaction Detail	/app/reporting/reportrunner.nl?cr=250
REPO_251	Account Detail	/app/reporting/reportrunner.nl?cr=251
REPO_256	Total Open Estimates	/app/reporting/reportrunner.nl?cr=256
REPO_257	Estimates Register	/app/reporting/reportrunner.nl?cr=257
REPO_258	Estimated Partner Commission Detail	/app/reporting/reportrunner.nl?cr=258
REPO_26	New Customer Sales Orders	/app/reporting/reportrunner.nl?cr=26
REPO_261	Open Invoices	/app/reporting/reportrunner.nl?cr=261

Task ID	Page Label in NetSuite	URL
REPO_262	A/R Register	/app/reporting/reportrunner.nl?cr=262
REPO_264	Customer Profitability Detail	/app/reporting/reportrunner.nl?cr=264
REPO_266	Customer Profitability	/app/reporting/reportrunner.nl?cr=266
REPO_267	Sales by Sales Rep Det ail	/app/reporting/reportrunner.nl?cr=267
REPO_268	Sales by Item Detail	/app/reporting/reportrunner.nl?cr=268
REPO_269	Sales by Customer Det ail	/app/reporting/reportrunner.nl?cr=269
REPO_27	Sales Orders by Promot ion Detail	/app/reporting/reportrunner.nl?cr=27
REPO_270	Sales by Sales Rep	/app/reporting/reportrunner.nl?cr=270
REPO_271	Sales by Item	/app/reporting/reportrunner.nl?cr=271
REPO_272	Sales by Customer	/app/reporting/reportrunner.nl?cr=272
REPO_273	A/R Aging Detail	/app/reporting/reportrunner.nl?cr=273
REPO_274	A/R Aging	/app/reporting/reportrunner.nl?cr=274
REPO_277	Open Purchase Orders	/app/reporting/reportrunner.nl?viewasrepor t=T&specacct=PurchOrd&openonly=T&acc tid=&cr=277
REPO_278	Purchase Order Regist er	/app/reporting/reportrunner.nl?viewasrepor t=T&specacct=PurchOrd&reporttype=REGIST ER&acctid=&cr=278
REPO_279	Purchase by Item Detail	/app/reporting/reportrunner.nl?subtype=i&c r=279
REPO_28	Sales Orders by Promot ion	/app/reporting/reportrunner.nl?cr=28
REPO_280	Purchase by Vendor Det ail	/app/reporting/reportrunner.nl?rtype=v&cr= 280
REPO_281	Purchase by Item	/app/reporting/reportrunner.nl?subtype=i&c r=281
REPO_282	Purchase by Vendor	/app/reporting/reportrunner.nl?subtype= v&cr=282
REPO_283	Open Bills	/app/reporting/reportrunner.nl?viewasrepor t=T&specacct=AcctPay&openonly=T&acctid= &cr=283
REPO_284	A/P Register	/app/reporting/reportrunner.nl?viewasrepor t=T&specacct=AcctPay&reporttype=REGISTE R&acctid=&cr=284
REPO_285	A/P Aging Detail	/app/reporting/reportrunner.nl?id=&ddue 1=&ddue2=&cr=285
REPO_286	A/P Aging	/app/reporting/reportrunner.nl?cr=286
REPO_29	Sales Orders by Partne r Detail	/app/reporting/reportrunner.nl?cr=29
REPO_292	Trial Balance	/app/reporting/reportrunner.nl?cr=292

Task ID	Page Label in NetSuite	URL
REPO_293	General Ledger	/app/reporting/reportrunner.nl?cr=293
REPO_3	Amortization Forecast Detail	/app/reporting/reportrunner.nl?cr=3
REPO_30	Sales Orders by Partner	/app/reporting/reportrunner.nl?cr=30
REPO_309	Sales by Customer Det ail filter for inv	/app/reporting/reportrunner.nl?cr=309&relo ad=T
REPO_31	Sales Orders by Histor ical Team Detail	/app/reporting/reportrunner.nl?cr=31
REPO_310	Back Ordered Items	/app/reporting/reportrunner.nl?cr=310&relo ad=T
REPO_318	Total Open Opportuni tes By Competitor	/app/reporting/reportrunner.nl?cr=318&relo ad=T
REPO_319	My Open Case Analysis Summary	/app/reporting/reportrunner.nl?cr=319&relo ad=T
REPO_32	Sales Orders by Histor ical Team	/app/reporting/reportrunner.nl?cr=32
REPO_325	Opportunities Lost By Competitor	/app/reporting/reportrunner.nl?cr=325&relo ad=T
REPO_326	Keyword - Customers by Keyword Family	/app/reporting/reportrunner.nl?cr=326&relo ad=T
REPO_327	Keyword - Leads by Sea rch Engine	/app/reporting/reportrunner.nl?cr=327&relo ad=T
REPO_327	0403 BS - Edit Text Row	/app/reporting/reportrunner.nl?cr=327&relo ad=T
REPO_328	Orders by Items by Cla ss	/app/reporting/reportrunner.nl?cr=328&relo ad=T
REPO_329	Orders by Customer by Class	/app/reporting/reportrunner.nl?cr=329&relo ad=T
REPO_33	Sales Orders by Item Detail	/app/reporting/reportrunner.nl?cr=33
REPO_335	Email Campaign Statis tics Summary	/app/reporting/reportrunner.nl?cr=335&relo ad=T
REPO_352	Utilization by Employee Report	/app/reporting/reportrunner.nl?cr=352&relo ad=T
REPO_353	Monthly Utilization by Consultant	/app/reporting/reportrunner.nl?cr=353&relo ad=T
REPO_354	Campaign ROI Detail	/app/reporting/reportrunner.nl?cr=354&relo ad=T
REPO_355	Campaign ROI Summar y	/app/reporting/reportrunner.nl?cr=355&relo ad=T
REPO_36	Sales Orders by Sales Rep	/app/reporting/reportrunner.nl?cr=36
REPO_38	Sales Orders by Custom er	/app/reporting/reportrunner.nl?cr=38

Task ID	Page Label in NetSuite	URL
REPO_4	Amortization Forecast	/app/reporting/reportrunner.nl?cr=4
REPO_40	Integration and Automation Usage Detail By Job	/app/reporting/reportrunner.nl?cr=40
REPO_41	Integration and Automation Usage Summary By Job	/app/reporting/reportrunner.nl?cr=41
REPO_42	Integration and Automation Usage Summary By Record Type	/app/reporting/reportrunner.nl?cr=42
REPO_43	Hosted Page Hits by Customer	/app/reporting/reportrunner.nl?cr=43
REPO_44	Page Hits by Customers	/app/reporting/reportrunner.nl?cr=44
REPO_48	Forecast by Item Detail	/app/reporting/reportrunner.nl?cr=48
REPO_49	Forecast by Item	/app/reporting/reportrunner.nl?cr=49
REPO_5	Aggregated Sales Orders by Sales Rep Summary	/app/reporting/reportrunner.nl?cr=5
REPO_50	Open Escalations	/app/reporting/reportrunner.nl?cr=50
REPO_51	Total Pipeline by Statuses Summary	/app/reporting/reportrunner.nl?cr=51
REPO_53	Calculated Forecast by Sales Rep	/app/reporting/reportrunner.nl?cr=53
REPO_54	Visitor Activity	/app/reporting/reportrunner.nl?cr=54
REPO_55	Lead Conversion	/app/reporting/reportrunner.nl?cr=55
REPO_56	Visitor Activity Detail	/app/reporting/reportrunner.nl?cr=56
REPO_57	Forecast by Sales Rep	/app/reporting/reportrunner.nl?cr=57
REPO_58	Sales Tax Liability By Tax Agency	/app/reporting/reportrunner.nl?cr=58
REPO_59	GST/HST on Purchases	/app/reporting/reportrunner.nl?cr=59
REPO_6	Authorized Partner Commission	/app/reporting/reportrunner.nl?cr=6
REPO_60	GST/HST on Sales	/app/reporting/reportrunner.nl?cr=60
REPO_61	GST/HST on Purchases Detail	/app/reporting/reportrunner.nl?cr=61
REPO_62	GST/HST on Sales Detail	/app/reporting/reportrunner.nl?cr=62
REPO_63	GST/HST Audit Summary	/app/reporting/reportrunner.nl?cr=63
REPO_64	Inventory Turnover	/app/reporting/reportrunner.nl?cr=64
REPO_65	Scheduled Deferred Revenue	/app/reporting/reportrunner.nl?cr=65
REPO_66	Revenue Recognition Forecast Detail	/app/reporting/reportrunner.nl?cr=66

Task ID	Page Label in NetSuite	URL
REPO_67	Revenue Recognition Forecast	/app/reporting/reportrunner.nl?cr=67
REPO_68	Total Open Opportunities	/app/reporting/reportrunner.nl?cr=68
REPO_69	Opportunities to Close	/app/reporting/reportrunner.nl?cr=69
REPO_7	Authorized Partner Commission Detail	/app/reporting/reportrunner.nl?cr=7
REPO_71	Sales by Historical Team Detail	/app/reporting/reportrunner.nl?cr=71
REPO_72	Sales by Historical Team	/app/reporting/reportrunner.nl?cr=72
REPO_73	Forecast (Outstanding) by Customer Detail	/app/reporting/reportrunner.nl?cr=73
REPO_74	Forecast (Outstanding) by Customer	/app/reporting/reportrunner.nl?cr=74
REPO_77	State Withholding Detail	/app/reporting/reportrunner.nl?cr=77
REPO_78	State Withholding	/app/reporting/reportrunner.nl?cr=78
REPO_79	Hours & Earnings	/app/reporting/reportrunner.nl?cr=79
REPO_8	ALT_SALES Total Pipeline by Sales Rep Detail	/app/reporting/reportrunner.nl?cr=8
REPO_80	Closed Case Escalation	/app/reporting/reportrunner.nl?cr=80
REPO_81	Open Case Escalation	/app/reporting/reportrunner.nl?cr=81
REPO_82	Closed Case Escalation Detail	/app/reporting/reportrunner.nl?cr=82
REPO_83	Open Case Escalation Detail	/app/reporting/reportrunner.nl?cr=83
REPO_84	Payroll Detail	/app/reporting/reportrunner.nl?cr=-177
REPO_85	New Visitor	/app/reporting/reportrunner.nl?cr=85
REPO_87	Keywords Detail	/app/reporting/reportrunner.nl?cr=87
REPO_88	Referrer Detail	/app/reporting/reportrunner.nl?cr=88
REPO_9	ALT_SALES Total Pipeline by Sales Rep Summary	/app/reporting/reportrunner.nl?cr=9
REPO_90	Keywords	/app/reporting/reportrunner.nl?cr=90
REPO_91	Referrer	/app/reporting/reportrunner.nl?cr=91
REPO_93	Payroll Liability Detail	/app/reporting/reportrunner.nl?cr=93
REPO_94	Total Pipeline by Customer	/app/reporting/reportrunner.nl?cr=94
REPO_95	Total Pipeline by Customer Detail	/app/reporting/reportrunner.nl?cr=95
REPO_96	Estimates to Close	/app/reporting/reportrunner.nl?cr=96
REPO_97	Opportunities to Close Detail	/app/reporting/reportrunner.nl?cr=97

Task ID	Page Label in NetSuite	URL
REPO_98	Forecast by Customer Detail	/app/reporting/reportrunner.nl?cr=98
REPO_99	Forecast by Customer	/app/reporting/reportrunner.nl?cr=99
REPO_ASSIGNFINANCIALLAYOUTS	Row Layout Assignment	/app/reporting/financiallayoutassignments.nl
REPO_BANKREGISTER	Bank Register	/app/reporting/reportrunner.nl?reporttype=REGISTER&accttype=Bank&code=CHECK_REG
REPO_CUSTOMIZATION	Generic Report Customization	/app/reporting/reportcomposer.nl
REPO_CUST_10	Customize ALT_SALES Pipeline by Sales Rep Detail	/app/reporting/reportcomposer.nl?cr=10&e=T
REPO_CUST_100	Customize Total Pipeline by Sales Rep Summary	/app/reporting/reportcomposer.nl?cr=100&e=T
REPO_CUST_101	Customize Total Pipeline by Sales Rep Detail	/app/reporting/reportcomposer.nl?cr=101&e=T
REPO_CUST_102	Customize New Customer Sales Detail	/app/reporting/reportcomposer.nl?cr=102&e=T
REPO_CUST_103	Customize New Customer Sales Summary	/app/reporting/reportcomposer.nl?cr=103&e=T
REPO_CUST_104	Customize Opportunities Lost	/app/reporting/reportcomposer.nl?cr=104&e=T
REPO_CUST_105	Customize Opportunities Won	/app/reporting/reportcomposer.nl?cr=105&e=T
REPO_CUST_106	Customize Sales Activity by Customer Summary	/app/reporting/reportcomposer.nl?cr=106&e=T
REPO_CUST_107	Customize Sales Activity by Customer Detail	/app/reporting/reportcomposer.nl?cr=107&e=T
REPO_CUST_11	Customize ALT_SALES Pipeline by Sales Rep Summary	/app/reporting/reportcomposer.nl?cr=11&e=T
REPO_CUST_111	Customize Authorized Commissions Summary	/app/reporting/reportcomposer.nl?cr=111&e=T
REPO_CUST_112	Customize Authorized Commissions Detail	/app/reporting/reportcomposer.nl?cr=112&e=T
REPO_CUST_113	Customize GST on Purchases Summary	/app/reporting/reportcomposer.nl?cr=113&e=T
REPO_CUST_114	Customize GST on Sales Summary	/app/reporting/reportcomposer.nl?cr=114&e=T
REPO_CUST_115	Customize GST on Purchases Detail	/app/reporting/reportcomposer.nl?cr=115&e=T
REPO_CUST_116	Customize GST on Sales Detail	/app/reporting/reportcomposer.nl?cr=116&e=T

Task ID	Page Label in NetSuite	URL
REPO_CUST_118	Customize Commissions Pending Authorization Summary	/app/reporting/reportcomposer.nl?cr=118&e=T
REPO_CUST_119	Customize Commissions Pending Authorization Detail	/app/reporting/reportcomposer.nl?cr=119&e=T
REPO_CUST_12	Customize ALT_SALES Forecast by Sales Rep Detail	/app/reporting/reportcomposer.nl?cr=12&e=T
REPO_CUST_121	Customize Estimated Commissions Summary	/app/reporting/reportcomposer.nl?cr=121&e=T
REPO_CUST_122	Customize Estimated Commissions Detail	/app/reporting/reportcomposer.nl?cr=122&e=T
REPO_CUST_123	Customize Campaign Response Detail	/app/reporting/reportcomposer.nl?cr=123&e=T
REPO_CUST_126	Customize Pipeline by Customer Summary	/app/reporting/reportcomposer.nl?cr=126&e=T
REPO_CUST_127	Customize Pipeline by Customer Detail	/app/reporting/reportcomposer.nl?cr=127&e=T
REPO_CUST_128	Customize Pipeline by Sales Rep Summary	/app/reporting/reportcomposer.nl?cr=128&e=T
REPO_CUST_129	Customize Pipeline ROI by Manager	/app/reporting/reportcomposer.nl?cr=129&e=T
REPO_CUST_13	Customize ALT_SALES Forecast by Sales Rep Summary	/app/reporting/reportcomposer.nl?cr=13&e=T
REPO_CUST_130	Customize Pipeline by Sales Rep Detail	/app/reporting/reportcomposer.nl?cr=130&e=T
REPO_CUST_131	Customize Pipeline by Status Summary	/app/reporting/reportcomposer.nl?cr=131&e=T
REPO_CUST_132	Customize Campaign ROI Analysis Detail	/app/reporting/reportcomposer.nl?cr=132&e=T
REPO_CUST_133	Customize Campaign Response Summary	/app/reporting/reportcomposer.nl?cr=133&e=T
REPO_CUST_134	Customize Sales Activity Summary	/app/reporting/reportcomposer.nl?cr=134&e=T
REPO_CUST_135	Customize Sales Activity Detail	/app/reporting/reportcomposer.nl?cr=135&e=T
REPO_CUST_137	Customize VAT on Purchases Summary	/app/reporting/reportcomposer.nl?cr=137&e=T
REPO_CUST_138	Customize VAT on Sales Summary	/app/reporting/reportcomposer.nl?cr=138&e=T
REPO_CUST_139	Customize VAT on Purchases Detail	/app/reporting/reportcomposer.nl?cr=139&e=T
REPO_CUST_14	Customize Billings Forecast vs. Quota Report	/app/reporting/reportcomposer.nl?cr=14&e=T

Task ID	Page Label in NetSuite	URL
REPO_CUST_140	Customize VAT on Sales Detail	/app/reporting/reportcomposer.nl?cr=140&e=T
REPO_CUST_141	Customize Partner Activity Summary	/app/reporting/reportcomposer.nl?cr=141&e=T
REPO_CUST_142	Customize Partner Activity Detail	/app/reporting/reportcomposer.nl?cr=142&e=T
REPO_CUST_143	Customize Sales by Partner Summary	/app/reporting/reportcomposer.nl?cr=143&e=T
REPO_CUST_144	Customize Sales by Partner Detail	/app/reporting/reportcomposer.nl?cr=144&e=T
REPO_CUST_145	Customize Shipping Report	/app/reporting/reportcomposer.nl?cr=145&e=T
REPO_CUST_146	Customize Total Open Opportunities Detail	/app/reporting/reportcomposer.nl?cr=146&e=T
REPO_CUST_147	Customize Forecast vs. Quota	/app/reporting/reportcomposer.nl?cr=147&e=T
REPO_CUST_148	Customize Sales Orders Pending Fulfillment	/app/reporting/reportcomposer.nl?cr=148&e=T
REPO_CUST_149	Customize Forecast Detail by Sales Rep	/app/reporting/reportcomposer.nl?cr=149&e=T
REPO_CUST_15	Customize Bookings and Billings Forecast vs. Quota Report	/app/reporting/reportcomposer.nl?cr=15&e=T
REPO_CUST_150	Customize Inventory Back Order Report	/app/reporting/reportcomposer.nl?cr=150&e=T
REPO_CUST_151	Customize Inventory Activity Detail	/app/reporting/reportcomposer.nl?cr=151&e=T
REPO_CUST_16	Customize Bookings Forecast vs. Quota Report	/app/reporting/reportcomposer.nl?cr=16&e=T
REPO_CUST_161	Customize Customers by Partner Detail	/app/reporting/reportcomposer.nl?cr=161&e=T
REPO_CUST_162	Customize Customers by Partner Summary	/app/reporting/reportcomposer.nl?cr=162&e=T
REPO_CUST_169	Customize Closed Case Analysis Detail	/app/reporting/reportcomposer.nl?cr=169&e=T
REPO_CUST_17	Customize Gross Lead Analysis Detail Report	/app/reporting/reportcomposer.nl?cr=17&e=T
REPO_CUST_170	Customize Closed Case Analysis Summary	/app/reporting/reportcomposer.nl?cr=170&e=T
REPO_CUST_171	Customize Open Case Analysis Detail	/app/reporting/reportcomposer.nl?cr=171&e=T&open=T
REPO_CUST_172	Customize Open Case Analysis Summary	/app/reporting/reportcomposer.nl?cr=172&e=T&open=T
REPO_CUST_173	Customize Customers by Sales Rep Detail	/app/reporting/reportcomposer.nl?cr=173&e=T

Task ID	Page Label in NetSuite	URL
REPO_CUST_174	Customize Customers by Sales Rep Summary	/app/reporting/reportcomposer.nl?cr=174&e=T
REPO_CUST_175	Customize Customers by Territory Detail	/app/reporting/reportcomposer.nl?cr=175&e=T
REPO_CUST_176	Customize Customers by Territory Summary	/app/reporting/reportcomposer.nl?cr=176&e=T
REPO_CUST_177	Customize Lead Source Analysis Detail	/app/reporting/reportcomposer.nl?cr=177&e=T
REPO_CUST_178	Customize Lead Source Analysis Summary	/app/reporting/reportcomposer.nl?cr=178&e=T
REPO_CUST_179	Customize Sales Orders Register	/app/reporting/reportcomposer.nl?cr=179&e=T
REPO_CUST_18	Customize Gross Lead Analysis Summary Report	/app/reporting/reportcomposer.nl?cr=18&e=T
REPO_CUST_180	Customize Open Sales Orders	/app/reporting/reportcomposer.nl?cr=180&e=T
REPO_CUST_183	Customize Shopping Activity Analysis by Visitor	/app/reporting/reportcomposer.nl?cr=183&e=T
REPO_CUST_184	Customize Shopping Activity Analysis by Category	/app/reporting/reportcomposer.nl?cr=184
REPO_CUST_185	Customize Shopping Activity Analysis	/app/reporting/reportcomposer.nl?e=T&cr=185
REPO_CUST_188	Customize Item Page Views and Sales Summary	/app/reporting/reportcomposer.nl?e=T&cr=188
REPO_CUST_189	Customize Hosted Page Hits by Visitor	/app/reporting/reportcomposer.nl?cr=189&e=T
REPO_CUST_19	A/P Payment History by Payment	/app/reporting/reportcomposer.nl?cr=19&e=T
REPO_CUST_191	Customize Hosted Page Hits	/app/reporting/reportcomposer.nl?cr=191&e=T
REPO_CUST_192	Customize Item Order History Detail	/app/reporting/reportcomposer.nl?cr=192&e=T
REPO_CUST_193	Customize Item Order History by Category	/app/reporting/reportcomposer.nl?cr=193
REPO_CUST_194	Customize Item Order History	/app/reporting/reportcomposer.nl?e=T&cr=194
REPO_CUST_195	Customize Page Hits Detail	/app/reporting/reportcomposer.nl?cr=195&e=T
REPO_CUST_196	Customize Page Hits by Category	/app/reporting/reportcomposer.nl?cr=196&e=T
REPO_CUST_197	Customize Page Hits	/app/reporting/reportcomposer.nl?e=T&cr=197

Task ID	Page Label in NetSuite	URL
REPO_CUST_198	Customize Sales by Promotion Detail	/app/reporting/reportcomposer.nl?e=T&payitemtype=r&cr=198
REPO_CUST_199	Customize Sales by Promotion Summary	/app/reporting/reportcomposer.nl?e=T&subtype=r&cr=199
REPO_CUST_2	Customize Deferred / Capitalized Expense	/app/reporting/reportcomposer.nl?cr=2&e=T
REPO_CUST_20	A/P Payment History by Bill	/app/reporting/reportcomposer.nl?cr=20&e=T
REPO_CUST_203	Customize Prospect Analysis Summary	/app/reporting/reportcomposer.nl?e=T&cr=203
REPO_CUST_204	Customize Prospect Analysis Detail	/app/reporting/reportcomposer.nl?e=T&cr=204
REPO_CUST_205	Customize Partner Commission Pending Authorization Summary	/app/reporting/reportcomposer.nl?e=T&cr=205
REPO_CUST_206	Customize Partner Commission Pending Authorization Detail	/app/reporting/reportcomposer.nl?e=T&cr=206
REPO_CUST_209	Customize Unbilled Cost by Customer Detail	/app/reporting/reportcomposer.nl?e=T&cr=209
REPO_CUST_21	A/R Payment History by Payment	/app/reporting/reportcomposer.nl?cr=21&e=T
REPO_CUST_210	Customize Unbilled Time by Customer Detail	/app/reporting/reportcomposer.nl?e=T&cr=210
REPO_CUST_211	Customize Unbilled Time by Customer Summary	/app/reporting/reportcomposer.nl?e=T&cr=211
REPO_CUST_214	Customize Sales Tax Transaction Detail	/app/reporting/reportcomposer.nl?cr=214&e=T
REPO_CUST_215	Customize Sales Tax Liability By Tax Item	/app/reporting/reportcomposer.nl?cr=215&e=T
REPO_CUST_218	Customize Time by Item Detail	/app/reporting/reportcomposer.nl?cr=218&e=T
REPO_CUST_219	Customize Time by Customer Detail	/app/reporting/reportcomposer.nl?cr=219&e=T
REPO_CUST_22	A/R Payment History by Invoice	/app/reporting/reportcomposer.nl?cr=22&e=T
REPO_CUST_220	Customize Time by Employee Detail	/app/reporting/reportcomposer.nl?cr=220&e=T
REPO_CUST_221	Customize Time by Item Summary	/app/reporting/reportcomposer.nl?cr=221&e=T
REPO_CUST_222	Customize Time by Customer Summary	/app/reporting/reportcomposer.nl?cr=222&e=T
REPO_CUST_223	Customize Time by Employee Summary	/app/reporting/reportcomposer.nl?cr=223&e=T

Task ID	Page Label in NetSuite	URL
REPO_CUST_225	Customize Unbilled Cost by Customer Summary	/app/reporting/reportcomposer.nl?e=T&cr=225
REPO_CUST_226	Customize Inventory Revenue Detail	/app/reporting/reportcomposer.nl?cr=226&e=T
REPO_CUST_227	Customize Inventory Valuation Detail	/app/reporting/reportcomposer.nl?e=T&cr=227
REPO_CUST_229	Customize Inventory Revenue Summary	/app/reporting/reportcomposer.nl?cr=229&e=T
REPO_CUST_230	Customize Inventory Valuation Summary	/app/reporting/reportcomposer.nl?cr=230&e=T
REPO_CUST_231	Customize Inventory Profitability	/app/reporting/reportcomposer.nl?cr=231&e=T
REPO_CUST_232	Customize Payroll Liabilities	/app/reporting/reportrunner.nl?payitemtype=&cr=232
REPO_CUST_233	Customize Payroll Journal	/app/reporting/reportrunner.nl?payitemtype=&cr=233
REPO_CUST_234	Customize Payroll Summary by Employee	/app/reporting/reportrunner.nl?cr=-176
REPO_CUST_235	Customize Payroll Summary by Item	/app/reporting/reportrunner.nl?payitemtype=&cr=235
REPO_CUST_239	Customize Item Transaction Detail	/app/reporting/reportcomposer.nl?cr=239&e=T
REPO_CUST_240	Customize Sales Back Order Report	/app/reporting/reportcomposer.nl?cr=240&e=T
REPO_CUST_241	Customize Physical Inventory Worksheet	/app/reporting/reportcomposer.nl?cr=241&e=T
REPO_CUST_242	Customize Payroll Check Register	/app/reporting/reportrunner.nl?cr=242
REPO_CUST_243	Customize Current Inventory Snapshot	/app/reporting/reportcomposer.nl?cr=243&e=T
REPO_CUST_246	Customize Items Pending Fulfillment	/app/reporting/reportcomposer.nl?e=T&viewasreport=T&specacct=SalesOrd&reporttype=REGISTER&code=OPEN_SALES_ORDERS_DETAIL&acctid=&cr=246
REPO_CUST_248	Customize Cost by Customer Detail	/app/reporting/reportcomposer.nl?cr=248&e=T
REPO_CUST_249	Customize Estimated Partner Commission Summary	/app/reporting/reportcomposer.nl?e=T&cr=249
REPO_CUST_25	Customize New Customer Sales Orders Detail	/app/reporting/reportcomposer.nl?cr=25&e=T
REPO_CUST_250	Customize Transaction Detail	/app/reporting/reportcomposer.nl?e=T&cr=250
REPO_CUST_251	Customize Account Detail	/app/reporting/reportcomposer.nl?cr=251&e=T

Task ID	Page Label in NetSuite	URL
REPO_CUST_256	Customize Total Open Estimates	/app/reporting/reportcomposer.nl?cr=256&e=T
REPO_CUST_257	Customize Estimates Register	/app/reporting/reportcomposer.nl?cr=257&e=T
REPO_CUST_258	Customize Estimated Partner Commission Detail	/app/reporting/reportcomposer.nl?e=T&cr=258
REPO_CUST_26	Customize New Customer Sales Orders Summary	/app/reporting/reportcomposer.nl?cr=26&e=T
REPO_CUST_261	Customize Open Invoices	/app/reporting/reportcomposer.nl?cr=261&e=T
REPO_CUST_262	Customize A/R Register	/app/reporting/reportcomposer.nl?cr=262&e=T
REPO_CUST_264	Customize Customer Profitability Detail	/app/reporting/reportcomposer.nl?e=T&cr=264
REPO_CUST_266	Customize Customer Profitability Summary	/app/reporting/reportcomposer.nl?cr=266&e=T
REPO_CUST_267	Customize Sales by Sales Rep Detail	/app/reporting/reportcomposer.nl?cr=267&e=T
REPO_CUST_268	Customize Sales by Item Detail	/app/reporting/reportcomposer.nl?cr=268&e=T
REPO_CUST_269	Customize Sales by Customer Detail	/app/reporting/reportcomposer.nl?cr=269&e=T
REPO_CUST_27	Customize Sales Orders by Promotion Detail	/app/reporting/reportcomposer.nl?cr=27&e=T
REPO_CUST_270	Customize Sales by Sales Rep Summary	/app/reporting/reportcomposer.nl?cr=270&e=T
REPO_CUST_271	Customize Sales by Item Summary	/app/reporting/reportcomposer.nl?cr=271&e=T
REPO_CUST_272	Customize Sales by Customer Summary	/app/reporting/reportcomposer.nl?cr=272&e=T
REPO_CUST_273	Customize A/R Aging Detail	/app/reporting/reportcomposer.nl?cr=273&e=T
REPO_CUST_274	Customize A/R Aging Summary	/app/reporting/reportcomposer.nl?cr=274&e=T
REPO_CUST_277	Customize Open Purchase Orders	/app/reporting/reportcomposer.nl?e=T&viewasreport=T&specacct=PurchOrd&openonly=T&reporttype=REGISTER&code=OPEN_PO&acctid=&cr=277
REPO_CUST_278	Customize Purchase Order Register	/app/reporting/reportcomposer.nl?e=T&viewasreport=T&specacct=PurchOrd&reporttype=REGISTER&code=PO_REG&acctid=&cr=278
REPO_CUST_279	Customize Purchase by Item Detail	/app/reporting/reportcomposer.nl?e=T&payitemtype=i&code=PURCH_DETAIL_BY_ITEM&cr=279

Task ID	Page Label in NetSuite	URL
REPO_CUST_28	Customize Sales Orders by Promotion Summary	/app/reporting/reportcomposer.nl?cr=28&e=T
REPO_CUST_280	Customize Purchase by Vendor Detail	/app/reporting/reportcomposer.nl?e=T&rtype=v&code=PURCH_DETAIL_BY_VENDOR&cr=280
REPO_CUST_281	Customize Purchase by Item Summary	/app/reporting/reportcomposer.nl?e=T&payitemtype=i&code=PURCH_SUM_BY_ITEM&cr=281
REPO_CUST_282	Customize Purchase by Vendor Summary	/app/reporting/reportcomposer.nl?e=T&payitemtype=v&code=PURCH_SUM_BY_VENDOR&cr=282
REPO_CUST_283	Customize Open Bills	/app/reporting/reportcomposer.nl?e=T&viewasreport=T&specacct=AcctPay&openonly=T&reporttype=REGISTER&code=OPEN_BILLS&acctid=&cr=283
REPO_CUST_284	Customize A/P Register	/app/reporting/reportcomposer.nl?e=T&viewasreport=T&specacct=AcctPay&reporttype=REGISTER&code=AP_REG&acctid=&cr=284
REPO_CUST_285	Customize A/P Aging Detail	/app/reporting/reportcomposer.nl?e=T&acctid=&ddue1=&ddue2=&code=AP_AGING_DETAIL&cr=285
REPO_CUST_286	Customize A/P Aging Summary	/app/reporting/reportcomposer.nl?e=T&cr=286
REPO_CUST_29	Customize Sales Orders by Partner Detail	/app/reporting/reportcomposer.nl?cr=29&e=T
REPO_CUST_292	Customize Trial Balance	/app/reporting/reportcomposer.nl?e=T&cr=292
REPO_CUST_293	Customize General Ledger	/app/reporting/reportcomposer.nl?e=T&cr=293
REPO_CUST_3	Customize Amortization Forecast Detail	/app/reporting/reportcomposer.nl?cr=3&e=T
REPO_CUST_30	Customize Sales Orders by Partner Summary	/app/reporting/reportcomposer.nl?cr=30&e=T
REPO_CUST_31	Customize Sales Orders by Historical Team Detail	/app/reporting/reportcomposer.nl?cr=31&e=T
REPO_CUST_310	Customize Back Ordered Items	/app/reporting/reportcomposer.nl?cr=310
REPO_CUST_33	Customize Sales Orders by Item Detail	/app/reporting/reportcomposer.nl?cr=33&e=T
REPO_CUST_34	Customize Sales Orders by Item Summary	/app/reporting/reportcomposer.nl?cr=34&e=T
REPO_CUST_355	Customize Campaign ROI Summary	/app/reporting/reportcomposer.nl?e=T&cr=355
REPO_CUST_39	Customize Forecast by Status Summary	/app/reporting/reportcomposer.nl?cr=39&e=T

Task ID	Page Label in NetSuite	URL
REPO_CUST_4	Customize Amortization Forecast Summary	/app/reporting/reportcomposer.nl?cr=4&e=T
REPO_CUST_40	Customize Integration and Automation Usage Detail By Job	/app/reporting/reportcomposer.nl?cr=40&e=T
REPO_CUST_41	Customize Integration and Automation Usage Summary By Job	/app/reporting/reportcomposer.nl?cr=41&e=T
REPO_CUST_42	Customize Integration and Automation Usage Summary By Record Type	/app/reporting/reportcomposer.nl?cr=42&e=T
REPO_CUST_43	Hosted Page Hits by Customer	/app/reporting/reportcomposer.nl?cr=43&e=T
REPO_CUST_44	Page Hits by Customer	/app/reporting/reportcomposer.nl?cr=44&e=T
REPO_CUST_48	Customize Forecast by Item Detail	/app/reporting/reportcomposer.nl?cr=48&e=T
REPO_CUST_49	Customize Forecast by Item Summary	/app/reporting/reportcomposer.nl?cr=49&e=T
REPO_CUST_5	Customize Aggregated Sales Orders by Sales Rep Summary	/app/reporting/reportcomposer.nl?cr=5&e=T
REPO_CUST_51	Customize Total Pipeline by Statuses	/app/reporting/reportcomposer.nl?cr=51&e=T
REPO_CUST_53	Customize Calculated Forecast by Sales Rep	/app/reporting/reportcomposer.nl?cr=53&e=T
REPO_CUST_54	Customize Visitor Activity Summary	/app/reporting/reportcomposer.nl?cr=54&e=T
REPO_CUST_55	Customize Lead Conversion	/app/reporting/reportcomposer.nl?cr=55&e=T
REPO_CUST_56	Customize Visitor Activity Detail	/app/reporting/reportcomposer.nl?cr=56&e=T
REPO_CUST_57	Customize Forecast by Sales Rep Summary	/app/reporting/reportcomposer.nl?cr=57&e=T
REPO_CUST_58	Customize Sales Tax Liability By Tax Agency	/app/reporting/reportcomposer.nl?cr=58&e=T
REPO_CUST_59	Customize GST/HST on Purchases Summary	/app/reporting/reportrunner.nl?cr=59&e=T
REPO_CUST_6	Customize Authorized Partner Commission Summary	/app/reporting/reportcomposer.nl?cr=6&e=T
REPO_CUST_60	Customize GST/HST on Sales Summary	/app/reporting/reportrunner.nl?cr=60&e=T
REPO_CUST_61	Customize GST/HST on Purchases Detail	/app/reporting/reportrunner.nl?cr=61&e=T

Task ID	Page Label in NetSuite	URL
REPO_CUST_62	Customize GST/HST on Sales Detail	/app/reporting/reportrunner.nl?cr=62&e=T
REPO_CUST_63	Customize GST/HST Audit Summary	/app/reporting/reportrunner.nl?cr=63&e=T
REPO_CUST_64	Customize Inventory Turnover	/app/reporting/reportcomposer.nl?cr=64&e=T
REPO_CUST_65	Customize Deferred Revenue Forecast Summary	/app/reporting/reportcomposer.nl?cr=65&e=T
REPO_CUST_66	Customize Deferred Revenue Detail	/app/reporting/reportcomposer.nl?cr=66&e=T
REPO_CUST_67	Customize Deferred Revenue by Customer Summary	/app/reporting/reportcomposer.nl?cr=67&e=T
REPO_CUST_68	Customize Total Open Opportunities Summary	/app/reporting/reportcomposer.nl?cr=68&e=T
REPO_CUST_69	Customize Opportunities to Close Summary	/app/reporting/reportcomposer.nl?cr=69&e=T
REPO_CUST_7	Customize Authorized Partner Commission Detail	/app/reporting/reportcomposer.nl?cr=7&e=T
REPO_CUST_71	Customize Sales by Historical Team Detail	/app/reporting/reportcomposer.nl?cr=71&e=T
REPO_CUST_72	Customize Sales by Historical Team Summary	/app/reporting/reportcomposer.nl?cr=72&e=T
REPO_CUST_73	Customize Forecast (Outstanding) by Customer Detail	/app/reporting/reportcomposer.nl?cr=73&e=T
REPO_CUST_74	Customize Forecast (Outstanding) by Customer Summary	/app/reporting/reportcomposer.nl?cr=74&e=T
REPO_CUST_8	Customize ALT_SALES Total Pipeline by Sales Rep Detail	/app/reporting/reportcomposer.nl?cr=8&e=T
REPO_CUST_80	Closed Case Escalation Summary Customize	/app/reporting/reportcomposer.nl?cr=80&e=T
REPO_CUST_81	Open Case Escalation Summary Customize	/app/reporting/reportcomposer.nl?cr=81&e=T
REPO_CUST_82	Closed Case Escalation Detail Customize	/app/reporting/reportcomposer.nl?cr=82&e=T
REPO_CUST_83	Open Case Escalation Detail Customize	/app/reporting/reportcomposer.nl?cr=83&e=T
REPO_CUST_84	Customize Payroll Detail	/app/reporting/reportrunner.nl?cr=-177
REPO_CUST_85	New Visitor	/app/reporting/reportcomposer.nl?cr=85&e=T
REPO_CUST_87	Keywords Detail	/app/reporting/reportcomposer.nl?cr=87&e=T

Task ID	Page Label in NetSuite	URL
REPO_CUST_88	Referrer Detail	/app/reporting/reportcomposer.nl?cr=88&e=T
REPO_CUST_9	Customize ALT_SALES Total Pipeline by Sales Rep Summary	/app/reporting/reportcomposer.nl?cr=9&e=T
REPO_CUST_90	Keywords Summary	/app/reporting/reportcomposer.nl?cr=90&e=T
REPO_CUST_91	Referrer Summary	/app/reporting/reportcomposer.nl?cr=91&e=T
REPO_CUST_94	Customize Total Pipeline by Customer Summary	/app/reporting/reportcomposer.nl?cr=94&e=T
REPO_CUST_95	Customize Total Pipeline by Customer Detail	/app/reporting/reportcomposer.nl?cr=95&e=T
REPO_CUST_96	Customize Estimates to Close	/app/reporting/reportcomposer.nl?cr=96&e=T
REPO_CUST_97	Customize Opportunities to Close Detail	/app/reporting/reportcomposer.nl?cr=97&e=T
REPO_CUST_98	Customize Forecast by Customer Detail	/app/reporting/reportcomposer.nl?cr=98&e=T
REPO_CUST_99	Customize Forecast by Customer Summary	/app/reporting/reportcomposer.nl?cr=99&e=T
REPO_CUST_NEG100	Customize Case Activity by Support Rep Summary	/app/reporting/reportcomposer.nl?cr=-100&e=T
REPO_CUST_NEG101	Customize Case Activity by Support Rep Detail	/app/reporting/reportcomposer.nl?cr=-101&e=T
REPO_CUST_NEG105	Customize Utilization by Employee Summary (deprecated)	/app/reporting/reportcomposer.nl?cr=-105&e=T
REPO_CUST_NEG106	Customize Current Backlog by Resource	/app/reporting/reportcomposer.nl?cr=-106&e=T
REPO_CUST_NEG108	Customize Sales Orders by Sales Team Summary	/app/reporting/reportcomposer.nl?cr=-108&e=T
REPO_CUST_NEG109	Customize Sales Orders by Sales Team Detail	/app/reporting/reportcomposer.nl?cr=-109&e=T
REPO_CUST_NEG110	Customize Sales by Sales Team Summary	/app/reporting/reportcomposer.nl?cr=-110&e=T
REPO_CUST_NEG111	Customize Sales by Sales Team Detail	/app/reporting/reportcomposer.nl?cr=-111&e=T
REPO_CUST_NEG112	Customize Aggregated Sales Orders by Customer Summary	/app/reporting/reportcomposer.nl?cr=-112&e=T
REPO_CUST_NEG113	Customize Aggregated Sales Orders by Item Summary	/app/reporting/reportcomposer.nl?cr=-113&e=T

Task ID	Page Label in NetSuite	URL
REPO_CUST_NEG114	Customize Internal Search Summary	/app/reporting/reportcomposer.nl?cr=-114&e=T
REPO_CUST_NEG115	Customize Internal Search Detail	/app/reporting/reportcomposer.nl?cr=-115&e=T
REPO_CUST_NEG116	Customize Customer Aging History	/app/reporting/reportcomposer.nl?cr=-116&e=T
REPO_CUST_NEG118	Customize Estimated Profitability by Job	/app/reporting/reportcomposer.nl?cr=-118&e=T
REPO_CUST_NEG120	Customize Sales Tax on Sales Summary	/app/reporting/reportcomposer.nl?cr=-120&e=T
REPO_CUST_NEG121	Customize Sales Tax on Sales Detail	/app/reporting/reportcomposer.nl?cr=-121&e=T
REPO_CUST_NEG122	Customize Campaign ROI Analysis Summary	/app/reporting/reportcomposer.nl?cr=-122&e=T
REPO_CUST_NEG123	Customize Sales by Lead Source Detail	/app/reporting/reportcomposer.nl?cr=-123&e=T
REPO_CUST_NEG124	Customize Sales by Lead Source Summary	/app/reporting/reportcomposer.nl?cr=-124&e=T
REPO_CUST_NEG125	Customize PST on Sales Detail	/app/reporting/reportrunner.nl?cr=-125&e=T
REPO_CUST_NEG126	Customize PST on Purchases Detail	/app/reporting/reportrunner.nl?cr=-126&e=T
REPO_CUST_NEG127	Customize PST on Sales Summary	/app/reporting/reportrunner.nl?cr=-127&e=T
REPO_CUST_NEG128	Customize PST on Purchases Summary	/app/reporting/reportrunner.nl?cr=-128&e=T
REPO_CUST_NEG130	Customize Bank Register	/app/reporting/reportcomposer.nl?cr=-130&e=T
REPO_CUST_NEG131	Customize Sales Tax Analysis	/app/reporting/reportcomposer.nl?cr=-131&e=T
REPO_CUST_NEG132	Customize Realized Exchange Rate Gains and Losses	/app/reporting/reportcomposer.nl?cr=-132&e=T
REPO_CUST_NEG133	Customize Unrealized Exchange Rate Gains and Losses	/app/reporting/reportcomposer.nl?cr=-133&e=T
REPO_CUST_NEG134	Customize Sales by Paid Keyword Detail	/app/reporting/reportcomposer.nl?cr=-134&e=T
REPO_CUST_NEG135	Customize Sales by Paid Keyword Summary	/app/reporting/reportcomposer.nl?cr=-135&e=T
REPO_CUST_NEG136	Customize Leads by Paid Keyword Detail	/app/reporting/reportcomposer.nl?cr=-136&e=T
REPO_CUST_NEG137	Customize Leads by Paid Keyword Summary	/app/reporting/reportcomposer.nl?cr=-137&e=T
REPO_CUST_NEG138	Customize Closed Issues Summary	/app/reporting/reportcomposer.nl?cr=-138&e=T&reload=T

Task ID	Page Label in NetSuite	URL
REPO_CUST_NEG139	Customize Closed Issues Detail	/app/reporting/reportcomposer.nl?cr=-139&e=T&reload=T
REPO_CUST_NEG140	Customize Open Issues Summary	/app/reporting/reportcomposer.nl?cr=-140&e=T&reload=T
REPO_CUST_NEG141	Customize Open Issues Detail	/app/reporting/reportcomposer.nl?cr=-141&e=T&reload=T
REPO_CUST_NEG144	Customize Paid Employee Commission Summary	/app/reporting/reportcomposer.nl?cr=-144&e=T&reload=T
REPO_CUST_NEG145	Customize Paid Employee Commission Detail	/app/reporting/reportcomposer.nl?cr=-145&e=T&reload=T
REPO_CUST_NEG146	Customize Paid Partner Commission Summary	/app/reporting/reportcomposer.nl?cr=-146&e=T&reload=T
REPO_CUST_NEG147	Customize Paid Partner Commission Detail	/app/reporting/reportcomposer.nl?cr=-147&e=T&reload=T
REPO_CUST_NEG148	Customize Sales Order Revenue Forecast Summary	/app/reporting/reportcomposer.nl?cr=-148&e=T&reload=T
REPO_CUST_NEG149	Customize Sales Order Revenue Forecast Detail	/app/reporting/reportcomposer.nl?cr=-149&e=T&reload=T
REPO_CUST_NEG150	Customize Forecast vs. Quota (including Class)	/app/reporting/reportcomposer.nl?cr=-150&e=T&reload=T
REPO_CUST_NEG151	Customize Forecast Accuracy	/app/reporting/reportcomposer.nl?cr=-151&e=T&reload=T
REPO_CUST_NEG152	Customize Forecast (ALT_SALES)	/app/reporting/reportcomposer.nl?cr=-152&e=T&reload=T
REPO_CUST_NEG153	Customize Calculated Forecast Accuracy	/app/reporting/reportcomposer.nl?cr=-153&e=T&reload=T
REPO_CUST_NEG154	Customize Calculated Forecast Accuracy (ALT_SALES)	/app/reporting/reportcomposer.nl?cr=-154&e=T&reload=T
REPO_CUST_NEG155	Customize Forecast vs. Quota (including Class)	/app/reporting/reportcomposer.nl?cr=-155&e=T&reload=T
REPO_CUST_NEG156	Customize Forecast vs. Quota (including Department)	/app/reporting/reportcomposer.nl?cr=-156&e=T&reload=T
REPO_CUST_NEG157	Customize ALT_SALES Forecast vs. Quota (including Class)	/app/reporting/reportcomposer.nl?cr=-157&e=T&reload=T
REPO_CUST_NEG158	Customize ALT_SALES Forecast vs. Quota (including Department)	/app/reporting/reportcomposer.nl?cr=-158&e=T&reload=T
REPO_CUST_NEG159	Customize ALT_SALES Forecast vs. Quota (including Location)	/app/reporting/reportcomposer.nl?cr=-159&e=T&reload=T
REPO_CUST_NEG160	Customize Forecast Accuracy	/app/reporting/reportcomposer.nl?cr=-160&e=T&reload=T

Task ID	Page Label in NetSuite	URL
REPO_CUST_NEG161	Customize Calculated Forecast Accuracy	/app/reporting/reportcomposer.nl?cr=-161&e=T&reload=T
REPO_CUST_NEG162	Customize Earned Value by Job	/app/reporting/reportcomposer.nl?cr=-162&e=T
REPO_CUST_NEG175	Customize Service Fees	/app/reporting/reportrunner.nl?cr=-175
REPO_CUST_NEG178	Customize Utilization by Employee (deprecated)	/app/reporting/reportcomposer.nl?cr=-178&e=T
REPO_CUST_NEG179	Customize Deferred Revenue By Customer	/app/reporting/reportcomposer.nl?cr=-179&e=T
REPO_CUST_NEG180	Customize Deferred Revenue By Item	/app/reporting/reportcomposer.nl?cr=-180&e=T
REPO_CUST_NEG181	Customize Deferred Revenue By State	/app/reporting/reportcomposer.nl?cr=-181&e=T
REPO_CUST_NEG182	Customize Revenue By Customer	/app/reporting/reportcomposer.nl?cr=-182&e=T
REPO_CUST_NEG183	Customize Revenue By Item	/app/reporting/reportcomposer.nl?cr=-183&e=T
REPO_CUST_NEG184	Customize Revenue By State	/app/reporting/reportcomposer.nl?cr=-184&e=T
REPO_CUST_NEG185	Customize Commission Overview	/app/reporting/reportcomposer.nl?cr=-185&e=T
REPO_CUST_NEG186	Customize Partner Commission Overview	/app/reporting/reportcomposer.nl?cr=-186&e=T
REPO_CUST_NEG187	Customize Receivables by Customer	/app/reporting/reportcomposer.nl?cr=-187&e=T
REPO_CUST_NEG188	Customize Billing and Revenue Summary	/app/reporting/reportcomposer.nl?cr=-188&e=T
REPO_CUST_NEG195	Customize Budget Income Detail	/app/reporting/financialreportcomposer.nl?e=T&cr=-195
REPO_CUST_NEG196	Customize Budget Income Statement	/app/reporting/financialreportcomposer.nl?e=T&cr=-196
REPO_CUST_NEG197	Customize Budget vs. Actual	/app/reporting/financialreportcomposer.nl?e=T&cr=-197
REPO_CUST_NEG198	Customize Comparative Balance Sheet	/app/reporting/financialreportcomposer.nl?e=T&cr=-198
REPO_CUST_NEG199	Customize Comparative Income Statement	/app/reporting/financialreportcomposer.nl?e=T&cr=-199
REPO_CUST_NEG200	Customize Income Statement	/app/reporting/financialreportcomposer.nl?e=T&cr=-200
REPO_CUST_NEG201	Customize Cash Statement	/app/reporting/financialreportcomposer.nl?e=T&cr=-201
REPO_CUST_NEG202	Customize Balance Sheet	/app/reporting/financialreportcomposer.nl?e=T&cr=-202
REPO_CUST_NEG203	Customize Cash Flow Statement	/app/reporting/financialreportcomposer.nl?e=T&cr=-203

Task ID	Page Label in NetSuite	URL
REPO_CUST_NEG204	Customize Income Statement Detail	/app/reporting/financialreportcomposer.nl?e=T&cr=-204
REPO_CUST_NEG205	Customize Cash Detail	/app/reporting/financialreportcomposer.nl?e=T&cr=-205
REPO_CUST_NEG206	Customize Balance Sheet Detail	/app/reporting/financialreportcomposer.nl?e=T&cr=-206
REPO_CUST_NEG209	Customize Open Return Authorizations	/app/reporting/reportcomposer.nl?cr=-209&e=T
REPO_CUST_NEG210	Customize Return Authorizations Pending Receipt	/app/reporting/reportcomposer.nl?cr=-210&e=T
REPO_CUST_NEG211	Customize Return Authorizations Register	/app/reporting/reportcomposer.nl?cr=-211&e=T
REPO_CUST_NEG212	Customize Time Entry Exceptions	/app/reporting/reportcomposer.nl?cr=-212&e=T
REPO_CUST_NEG213	Customize Sales Orders by Historical Team Summary (Transaction Date)	/app/reporting/reportcomposer.nl?cr=-213&e=T
REPO_CUST_NEG214	Customize Sales Orders by Historical Team Detail (Transaction Date)	/app/reporting/reportcomposer.nl?cr=-214&e=T
REPO_CUST_NEG215	Customize Sales by Historical Team Summary (Transaction Date)	/app/reporting/reportcomposer.nl?cr=-215&e=T
REPO_CUST_NEG216	Customize Sales by Historical Team Detail (Transaction Date)	/app/reporting/reportcomposer.nl?cr=-216&e=T
REPO_CUST_NEG218	Customize Deferred Revenue Waterfall	/app/reporting/reportcomposer.nl?cr=-218&e=T
REPO_CUST_NEG221	Customize Transfer Order Register	/app/reporting/reportcomposer.nl?cr=-221&e=T
REPO_CUST_NEG222	Customize Purchase Order History	/app/reporting/reportcomposer.nl?cr=-222&e=T
REPO_CUST_NEG223	Customize Item Demand Forecast vs Sales	/app/reporting/reportcomposer.nl?cr=-223&e=T
REPO_CUST_NEG224	Customize Item Demand Plan by Item	/app/reporting/reportcomposer.nl?cr=-224&e=T
REPO_CUST_NEG225	Customize Demand History by Item	/app/reporting/reportcomposer.nl?cr=-225&e=T
REPO_CUST_NEG226	Customize Production Variances by Item	/app/reporting/reportcomposer.nl?cr=-226&e=T
REPO_CUST_NEG227	Customize Purchase Price Variances by Item	/app/reporting/reportcomposer.nl?cr=-227&e=T
REPO_CUST_NEG228	Customize Planned Standard Costs by Cost Version	/app/reporting/reportcomposer.nl?cr=-228&e=T

Task ID	Page Label in NetSuite	URL
REPO_CUST_NEG231	Customize Intercompany Elimination	/app/reporting/reportcomposer.nl?cr=-231&e=T
REPO_CUST_NEG232	Customize Intercompany Reconciliation	/app/reporting/reportcomposer.nl?cr=-232&e=T
REPO_CUST_NEG233	Customize Open Estimates By Lines	/app/reporting/reportcomposer.nl?cr=-233&e=T
REPO_CUST_NEG234	Customize Open Estimates By Lines Detail	/app/reporting/reportcomposer.nl?cr=-234&e=T
REPO_CUST_NEG235	Customize Revenue Reclassification Detail	/app/reporting/reportcomposer.nl?e=T&cr=-235
REPO_CUST_NEG236	Customize Revenue Reclassification	/app/reporting/reportcomposer.nl?e=T&cr=-236
REPO_CUST_NEG239	Customize Allocated vs. Actual Hours by Resource	/app/reporting/reportcomposer.nl?cr=-239&e=T
REPO_CUST_NEG240	Customize Allocated Utilization by Resource	/app/reporting/reportcomposer.nl?cr=-240&e=T
REPO_CUST_NEG241	Customize Commission Overview Detail	/app/reporting/reportcomposer.nl?cr=-241&e=T
REPO_CUST_NEG242	Customize Partner Commission Overview Detail	/app/reporting/reportcomposer.nl?cr=-242&e=T
REPO_CUST_NEG245	Customize GL Audit Numbering	/app/reporting/reportcomposer.nl?cr=-245&e=T
REPO_CUST_NEG246	Customize Planned Utilization by Resource	/app/reporting/reportcomposer.nl?cr=-246&e=T
REPO_CUST_NEG247	Customize Actual Utilization by Resource	/app/reporting/reportcomposer.nl?cr=-247&e=T
REPO_CUST_NEG248	Customize Allocated Utilization by Project	/app/reporting/reportcomposer.nl?cr=-248&e=T
REPO_CUST_NEG249	Customize Planned Utilization by Project	/app/reporting/reportcomposer.nl?cr=-249&e=T
REPO_CUST_NEG250	Customize Actual Utilization by Project	/app/reporting/reportcomposer.nl?cr=-250&e=T
REPO_CUST_NEG253	Customize Project Billing Report	/app/reporting/reportcomposer.nl?e=T&cr=-253
REPO_CUST_NEG254	Customize Project Cost Budget vs. Actual	/app/reporting/reportcomposer.nl?cr=-254&e=T
REPO_CUST_NEG255	Customize Project Billing Budget vs. Actual	/app/reporting/reportcomposer.nl?cr=-255&e=T
REPO_CUST_NEG256	Customize Project Task Cost Budget vs. Actual	/app/reporting/reportcomposer.nl?cr=-256&e=T
REPO_CUST_NEG257	Customize Project Task Billing Budget vs. Actual	/app/reporting/reportcomposer.nl?cr=-257&e=T
REPO_CUST_NEG258	Customize Project Profitability by Month	/app/reporting/reportcomposer.nl?cr=-258&e=T

Task ID	Page Label in NetSuite	URL
REPO_CUST_NEG259	Customize Deferred Revenue Reclassification Activity	/app/reporting/reportcomposer.nl?e=T&cr=-259
REPO_CUST_NEG260	Customize Deferred Revenue Reclassification	/app/reporting/reportcomposer.nl?e=T&cr=-260
REPO_CUST_NEG265	Customize Revenue Contract Activity	/app/reporting/reportcomposer.nl?e=T&cr=-265
REPO_CUST_NEG273	Customize Project Cost Budget vs. Actual Detail	/app/reporting/reportcomposer.nl?cr=-273&e=T
REPO_CUST_NEG274	Customize Project Billing Budget vs. Actual Detail	/app/reporting/reportcomposer.nl?cr=-274&e=T
REPO_CUST_NEG275	Customize Project Profitability Detail	/app/reporting/reportcomposer.nl?cr=-275&e=T
REPO_CUST_NEG277	Customize Deferred Revenue Rollforward	/app/reporting/reportcomposer.nl?e=T&cr=-277
REPO_CUST_NEG278	Customize Deferred Revenue Rollforward Detail	/app/reporting/reportcomposer.nl?e=T&cr=-278
REPO_CUST_NEG279	Customize Deferred Revenue Rollforward Customer	/app/reporting/reportcomposer.nl?e=T&cr=-279
REPO_CUST_NEG280	Customize Campaign Clickthrough Detail	/app/reporting/reportcomposer.nl?cr=-280&e=T
REPO_CUST_NEG281	Customize Campaign Clickthrough Summary	/app/reporting/reportcomposer.nl?cr=-281&e=T
REPO_CUST_NEG282	Customize CTA Balance Audit	/app/reporting/financialreportcomposer.nl?e=T&cr=-282
REPO_CUST_NEG284	Customize Project Profitability	/app/reporting/reportcomposer.nl?cr=-284&e=T
REPO_CUST_NEG285	Customize Project Charges Forecast	/app/reporting/reportcomposer.nl?cr=-285&e=T
REPO_CUST_NEG289	Customize Deferred Revenue Reclassification	/app/reporting/reportcomposer.nl?e=T&cr=-289
REPO_CUST_NEG290	Customize Deferred Revenue Detail	/app/reporting/reportcomposer.nl?cr=-290&e=T
REPO_CUST_NEG291	Customize Deferred Revenue by Customer Summary	/app/reporting/reportcomposer.nl?cr=-291&e=T
REPO_CUST_NEG292	Customize Deferred Revenue Rollforward	/app/reporting/reportcomposer.nl?e=T&cr=-292
REPO_CUST_NEG293	Customize Deferred Revenue Rollforward Detail	/app/reporting/reportcomposer.nl?e=T&cr=-293
REPO_CUST_NEG294	Customize Deferred Revenue Rollforward Customer	/app/reporting/reportcomposer.nl?e=T&cr=-294

Task ID	Page Label in NetSuite	URL
REPO_CUST_NEG295	Customize Deferred Revenue Waterfall Summary	/app/reporting/reportcomposer.nl?cr=-295&e=T
REPO_CUST_NEG296	Customize Monthly Recurring Revenue	/app/reporting/reportcomposer.nl?cr=-296&e=T
REPO_CUST_NEG297	Customize Billings To Date	/app/reporting/reportcomposer.nl?cr=-297&e=T
REPO_CUST_NEG298	Customize Stock Ledger	/app/reporting/reportcomposer.nl?cr=-298&e=T
REPO_CUST_NEG299	Customize Total Contract Value	/app/reporting/reportcomposer.nl?cr=-299&e=T
REPO_CUST_NEG306	Customize Time-Off Balance Details	/app/reporting/reportcomposer.nl?cr=-306&e=T
REPO_CUST_NEG307	Customize Time-Off Balance Summary	/app/reporting/reportcomposer.nl?cr=-307&e=T
REPO_CUST_NEG308	Customize Available Time-Off	/app/reporting/reportcomposer.nl?cr=-308&e=T
REPO_CUST_PAGE	All Saved Reports	/app/reporting/savedreports.nl
REPO_EMAIL	Email Report	/app/accounting/reports/emailContact.nl
REPO_FINANCIALLAYOUTS	Financial Row Layouts	/app/reporting/financiallayouts.nl
REPO_GST34	GST34 Worksheet	/app/accounting/reports/intl/ca/gst34.nl
REPO_JPTAXFORM	Tax Form	/app/accounting/reports/intl/jptaxform.nl
REPO_JP_BS	Japanese Balance Sheet	/app/accounting/reports/intl/jp/jpbalancesheet.nl
REPO_JP_CF	Japanese Cash Flow Statement	/app/accounting/reports/intl/jp/jpcashflow.nl
REPO_JP_EQUITY_CHANGE	Japanese Equity Change Report	/app/accounting/reports/intl/jp/jpequitychange.nl
REPO_JP_FR_PACKAGE	Japanese Financial Report Package	/app/accounting/reports/intl/jp/jpfrpackage.nl
REPO_JP_IS	Japanese Income Statement	/app/accounting/reports/intl/jp/jpincomestatement.nl
REPO_NEG100	Case Activity by Support Rep	/app/reporting/reportrunner.nl?cr=-100
REPO_NEG101	Case Activity by Support Rep Detail	/app/reporting/reportrunner.nl?cr=-101
REPO_NEG105	Utilization by Employee Summary (deprecated)	/app/reporting/reportrunner.nl?cr=-105
REPO_NEG106	Current Backlog by Resource	/app/reporting/reportrunner.nl?cr=-106
REPO_NEG108	Sales Orders by Sales Team	/app/reporting/reportrunner.nl?cr=-108
REPO_NEG109	Sales Orders by Sales Team Detail	/app/reporting/reportrunner.nl?cr=-109

Task ID	Page Label in NetSuite	URL
REPO_NEG110	Sales by Sales Team	/app/reporting/reportrunner.nl?cr=-110
REPO_NEG111	Sales by Sales Team Detail	/app/reporting/reportrunner.nl?cr=-111
REPO_NEG112	Aggregated Sales Orders by Customer Summary	/app/reporting/reportrunner.nl?cr=-112
REPO_NEG113	Aggregated Sales Orders by Item Summary	/app/reporting/reportrunner.nl?cr=-113
REPO_NEG114	Internal Search	/app/reporting/reportrunner.nl?cr=-114
REPO_NEG115	Internal Search Detail	/app/reporting/reportrunner.nl?cr=-115
REPO_NEG116	Customer Aging History	/app/reporting/reportrunner.nl?cr=-116
REPO_NEG118	Estimated Profitability by Job	/app/reporting/reportrunner.nl?cr=-118
REPO_NEG119	Reconciliation Detail	/app/reporting/reportrunner.nl?cr=-119
REPO_NEG120	Sales Tax on Sales	/app/reporting/reportrunner.nl?cr=-120
REPO_NEG121	Sales Tax on Sales Detail	/app/reporting/reportrunner.nl?cr=-121
REPO_NEG122	Campaign ROI Analysis	/app/reporting/reportrunner.nl?cr=-122
REPO_NEG123	Sales by Lead Source Detail	/app/reporting/reportrunner.nl?cr=-123
REPO_NEG124	Sales by Lead Source	/app/reporting/reportrunner.nl?cr=-124
REPO_NEG125	PST on Sales Detail	/app/reporting/reportrunner.nl?cr=-125
REPO_NEG126	PST on Purchases Detail	/app/reporting/reportrunner.nl?cr=-126
REPO_NEG127	PST on Sales	/app/reporting/reportrunner.nl?cr=-127
REPO_NEG128	PST on Purchases	/app/reporting/reportrunner.nl?cr=-128
REPO_NEG129	Reconciliation	/app/reporting/reportrunner.nl?cr=-129
REPO_NEG130	Bank Register	/app/reporting/reportrunner.nl?cr=-130
REPO_NEG131	Sales Tax Analysis	/app/reporting/reportrunner.nl?cr=-131
REPO_NEG132	Realized Exchange Rate Gains and Losses	/app/reporting/reportrunner.nl?cr=-132
REPO_NEG133	Unrealized Exchange Rate Gains and Losses	/app/reporting/reportrunner.nl?cr=-133
REPO_NEG134	Sales by Paid Keyword Detail	/app/reporting/reportrunner.nl?cr=-134
REPO_NEG135	Sales by Paid Keyword	/app/reporting/reportrunner.nl?cr=-135
REPO_NEG136	Leads by Paid Keyword Detail	/app/reporting/reportrunner.nl?cr=-136
REPO_NEG137	Leads by Paid Keyword	/app/reporting/reportrunner.nl?cr=-137
REPO_NEG138	Closed Issues	/app/reporting/reportrunner.nl?cr=-138
REPO_NEG139	Closed Issues Detail	/app/reporting/reportrunner.nl?cr=-139
REPO_NEG140	Open Issues	/app/reporting/reportrunner.nl?cr=-140

Task ID	Page Label in NetSuite	URL
REPO_NEG141	Open Issues Detail	/app/reporting/reportrunner.nl?cr=-141
REPO_NEG144	Paid Employee Commission	/app/reporting/reportrunner.nl?cr=-144
REPO_NEG145	Paid Employee Commission Detail	/app/reporting/reportrunner.nl?cr=-145
REPO_NEG146	Paid Partner Commission	/app/reporting/reportrunner.nl?cr=-146
REPO_NEG147	Paid Partner Commission Detail	/app/reporting/reportrunner.nl?cr=-147
REPO_NEG148	Sales Order Revenue Forecast	/app/reporting/reportrunner.nl?cr=-148
REPO_NEG149	Sales Order Revenue Forecast Detail	/app/reporting/reportrunner.nl?cr=-149
REPO_NEG150	Forecast vs. Quota by Class	/app/reporting/reportrunner.nl?cr=-150
REPO_NEG151	Forecast Accuracy	/app/reporting/reportrunner.nl?cr=-151
REPO_NEG152	Forecast Accuracy (ALT_SALES)	/app/reporting/reportrunner.nl?cr=-152
REPO_NEG153	Calculated Forecast Accuracy	/app/reporting/reportrunner.nl?cr=-153
REPO_NEG154	Calculated Forecast Accuracy (ALT_SALES)	/app/reporting/reportrunner.nl?cr=-154
REPO_NEG155	Forecast vs. Quota by Department	/app/reporting/reportrunner.nl?cr=-155
REPO_NEG156	Forecast vs. Quota by Location	/app/reporting/reportrunner.nl?cr=-156
REPO_NEG157	ALT_SALES Forecast vs. Quota by Class	/app/reporting/reportrunner.nl?cr=-157
REPO_NEG158	ALT_SALES Forecast vs. Quota by Department	/app/reporting/reportrunner.nl?cr=-158
REPO_NEG159	ALT_SALES Forecast vs. Quota by Location	/app/reporting/reportrunner.nl?cr=-159
REPO_NEG160	Forecast Accuracy	/app/reporting/reportrunner.nl?cr=-160
REPO_NEG161	Calculated Forecast Accuracy	/app/reporting/reportrunner.nl?cr=-161
REPO_NEG162	Earned Value by Job	/app/reporting/reportrunner.nl?cr=-162
REPO_NEG175	Service Fees	/app/reporting/reportrunner.nl?cr=-175
REPO_NEG178	Utilization by Employee (deprecated)	/app/reporting/reportrunner.nl?cr=-178
REPO_NEG179	Deferred Revenue By Customer	/app/reporting/reportrunner.nl?cr=-179
REPO_NEG180	Deferred Revenue By Item	/app/reporting/reportrunner.nl?cr=-180

Task ID	Page Label in NetSuite	URL
REPO_NEG181	Deferred Revenue By State	/app/reporting/reportrunner.nl?cr=-181
REPO_NEG182	Revenue By Customer	/app/reporting/reportrunner.nl?cr=-182
REPO_NEG183	Revenue By Item	/app/reporting/reportrunner.nl?cr=-183
REPO_NEG184	Revenue By State	/app/reporting/reportrunner.nl?cr=-184
REPO_NEG185	Commission Overview	/app/reporting/reportrunner.nl?cr=-185
REPO_NEG186	Partner Commission Overview	/app/reporting/reportrunner.nl?cr=-186
REPO_NEG187	Receivables by Custom er	/app/reporting/reportrunner.nl?cr=-187
REPO_NEG188	Billing and Revenue Summary	/app/reporting/reportrunner.nl?cr=-188
REPO_NEG195	Budget Income Statem ent Detail	/app/reporting/reportrunner.nl?cr=-195
REPO_NEG196	Budget Income Statem ent	/app/reporting/reportrunner.nl?cr=-196
REPO_NEG197	Budget vs. Actual	/app/reporting/reportrunner.nl?cr=-197
REPO_NEG198	Comparative Balance Sheet	/app/reporting/reportrunner.nl?cr=-198
REPO_NEG199	Comparative Income Statement	/app/reporting/reportrunner.nl?cr=-199
REPO_NEG200	Income Statement	/app/reporting/reportrunner.nl?cr=-200
REPO_NEG201	Cash Statement	/app/reporting/reportrunner.nl?cr=-201
REPO_NEG202	Balance Sheet	/app/reporting/reportrunner.nl?cr=-202
REPO_NEG203	Cash Flow Statement	/app/reporting/reportrunner.nl?cr=-203
REPO_NEG204	Income Statement Det ail	/app/reporting/reportrunner.nl?cr=-204
REPO_NEG205	Cash Statement Detail	/app/reporting/reportrunner.nl?cr=-205
REPO_NEG206	Balance Sheet Detail	/app/reporting/reportrunner.nl?cr=-206
REPO_NEG209	Open Return Authoriza tions	/app/reporting/reportrunner.nl?cr=-209
REPO_NEG210	Return Authorizations Pending Receipt	/app/reporting/reportrunner.nl?cr=-210
REPO_NEG211	Return Authorizations Register	/app/reporting/reportrunner.nl?cr=-211
REPO_NEG212	Time Entry Exceptions	/app/reporting/reportrunner.nl?cr=-212
REPO_NEG213	Sales Orders by Histor ical Team (Transaction Date)	/app/reporting/reportrunner.nl?cr=-213
REPO_NEG214	Sales Orders by Item Detail (Transaction Dat e)	/app/reporting/reportrunner.nl?cr=-214

Task ID	Page Label in NetSuite	URL
REPO_NEG215	Sales by Historical Team (Transaction Date)	/app/reporting/reportrunner.nl?cr=-215
REPO_NEG216	Sales by Item Detail (Transaction Date)	/app/reporting/reportrunner.nl?cr=-216
REPO_NEG218	Deferred Revenue Waterfall Detail	/app/reporting/reportrunner.nl?cr=-218
REPO_NEG221	Transfer Order Register	/app/reporting/reportrunner.nl?cr=-221
REPO_NEG222	Purchase Order History	/app/reporting/reportrunner.nl?cr=-222
REPO_NEG223	Item Demand Forecast vs Actual	/app/reporting/reportrunner.nl?cr=-223
REPO_NEG224	Item Demand Plan by Item	/app/reporting/reportrunner.nl?cr=-224
REPO_NEG225	Demand History by Item	/app/reporting/reportrunner.nl?cr=-225
REPO_NEG226	Production Variances by Item	/app/reporting/reportrunner.nl?cr=-226
REPO_NEG227	Purchase Price Variances by Item	/app/reporting/reportrunner.nl?cr=-227
REPO_NEG228	Planned Standard Costs by Cost Version	/app/reporting/reportrunner.nl?cr=-228
REPO_NEG231	Intercompany Elimination	/app/reporting/reportrunner.nl?cr=-231
REPO_NEG232	Intercompany Reconciliation	/app/reporting/reportrunner.nl?cr=-232
REPO_NEG233	Open Estimates By Lines	/app/reporting/reportrunner.nl?cr=-233
REPO_NEG234	Open Estimates By Lines Detail	/app/reporting/reportrunner.nl?cr=-234
REPO_NEG235	Revenue Reclassification Detail	/app/reporting/reportrunner.nl?cr=-235
REPO_NEG236	Revenue Reclassification	/app/reporting/reportrunner.nl?cr=-236
REPO_NEG237	Employee Change History	/app/reporting/reportrunner.nl?cr=-237
REPO_NEG239	Allocated vs. Actual Hours by Resource	/app/reporting/reportrunner.nl?cr=-239
REPO_NEG240	Allocated Utilization by Resource	/app/reporting/reportrunner.nl?cr=-240
REPO_NEG241	Commission Overview Detail	/app/reporting/reportrunner.nl?cr=-241
REPO_NEG242	Partner Commission Overview Detail	/app/reporting/reportrunner.nl?cr=-242
REPO_NEG245	GL Audit Numbering	/app/reporting/reportrunner.nl?cr=-245

Task ID	Page Label in NetSuite	URL
REPO_NEG246	Planned Utilization by Resource	/app/reporting/reportrunner.nl?cr=-246
REPO_NEG247	Actual Utilization by Res ource	/app/reporting/reportrunner.nl?cr=-247
REPO_NEG248	Allocated Utilization by Project	/app/reporting/reportrunner.nl?cr=-248
REPO_NEG249	Planned Utilization by Project	/app/reporting/reportrunner.nl?cr=-249
REPO_NEG250	Actual Utilization by Pro ject	/app/reporting/reportrunner.nl?cr=-250
REPO_NEG253	Project Billing Report	/app/reporting/reportrunner.nl?cr=-253
REPO_NEG254	Project Cost Budget vs. Actual	/app/reporting/reportrunner.nl?cr=-254
REPO_NEG255	Project Billing Budget vs. Actual	/app/reporting/reportrunner.nl?cr=-255
REPO_NEG256	Project Task Cost Bud get vs. Actual	/app/reporting/reportrunner.nl?cr=-256
REPO_NEG257	Project Task Billing Bud get vs. Actual	/app/reporting/reportrunner.nl?cr=-257
REPO_NEG258	Project Profitability by Month	/app/reporting/reportrunner.nl?cr=-258
REPO_NEG259	Deferred Revenue Rec lassification Activity	/app/reporting/reportrunner.nl?cr=-259
REPO_NEG260	Deferred Revenue Rec lassification	/app/reporting/reportrunner.nl?cr=-260
REPO_NEG265	Revenue Contract Activi ty	/app/reporting/reportrunner.nl?cr=-265
REPO_NEG271	Employee Payroll Item History	/app/reporting/reportrunner.nl?cr=-271
REPO_NEG273	Project Cost Budget vs. Actual Detail	/app/reporting/reportrunner.nl?cr=-273
REPO_NEG274	Project Billing Budget vs. Actual Detail	/app/reporting/reportrunner.nl?cr=-274
REPO_NEG275	Project Profitability Det ail	/app/reporting/reportrunner.nl?cr=-275
REPO_NEG277	Deferred Revenue Rol lforward	/app/reporting/reportrunner.nl?cr=-277
REPO_NEG278	Deferred Revenue Rol lforward Detail	/app/reporting/reportrunner.nl?cr=-278
REPO_NEG279	Deferred Revenue Rol lforward Customer	/app/reporting/reportrunner.nl?cr=-279
REPO_NEG280	Campaign Clickthrough Detail	/app/reporting/reportrunner.nl?cr=-280
REPO_NEG281	Campaign Clickthrough Summary	/app/reporting/reportrunner.nl?cr=-281

Task ID	Page Label in NetSuite	URL
REPO_NEG282	CTA Balance Audit	/app/reporting/reportrunner.nl?cr=-282
REPO_NEG284	Project Profitability	/app/reporting/reportrunner.nl?cr=-284
REPO_NEG285	Project Charges Forecast	/app/reporting/reportrunner.nl?cr=-285
REPO_NEG289	Deferred Revenue Reclassification	/app/reporting/reportrunner.nl?cr=-289
REPO_NEG290	Revenue Recognition Forecast Detail	/app/reporting/reportrunner.nl?cr=-290
REPO_NEG291	Revenue Recognition Forecast	/app/reporting/reportrunner.nl?cr=-291
REPO_NEG292	Deferred Revenue Rollforward	/app/reporting/reportrunner.nl?cr=-292
REPO_NEG293	Deferred Revenue Rollforward Detail	/app/reporting/reportrunner.nl?cr=-293
REPO_NEG294	Deferred Revenue Rollforward Customer	/app/reporting/reportrunner.nl?cr=-294
REPO_NEG295	Deferred Revenue Waterfall	/app/reporting/reportrunner.nl?cr=-295
REPO_NEG296	Monthly Recurring Revenue	/app/reporting/reportrunner.nl?cr=-296
REPO_NEG297	Billings To Date	/app/reporting/reportrunner.nl?cr=-297
REPO_NEG298	Stock Ledger	/app/reporting/reportrunner.nl?cr=-298
REPO_NEG299	Total Contract Value	/app/reporting/reportrunner.nl?cr=-299
REPO_NEG306	Time-Off Balance Details	/app/reporting/reportrunner.nl?cr=-306
REPO_NEG307	Time-Off Balance Summary	/app/reporting/reportrunner.nl?cr=-307
REPO_NEG308	Available Time-Off	/app/reporting/reportrunner.nl?cr=-308
REPO_NEWFINANCIALREPORT	New Financial Report	/app/reporting/newfinancialreport.nl
REPO_QUICKREPORT	New Report	/app/reporting/quickreports.nl
REPO_RECENT	Recent Reports	/app/common/otherlists/recentrecords.nl
REPO_REGISTER_ACCTPAY	Accounts Payable Register	/app/reporting/reportrunner.nl?reporttype=REGISTER&accttype=AcctPay
REPO_REGISTER_ACCTREC	Accounts Receivable Register	/app/reporting/reportrunner.nl?reporttype=REGISTER&accttype=AcctRec
REPO_REGISTER_BANK	Bank Register	/app/reporting/reportrunner.nl?reporttype=REGISTER&accttype=Bank
REPO_REGISTER_COGS	COGS Register	/app/reporting/reportrunner.nl?reporttype=REGISTER&accttype=COGS
REPO_REGISTER_CREDCARD	Credit Card Register	/app/reporting/reportrunner.nl?reporttype=REGISTER&accttype=CredCard
REPO_REGISTER_DEFEREXPENSE	Deferred Expense Register	/app/reporting/reportrunner.nl?reporttype=REGISTER&accttype=DeferExpense

Task ID	Page Label in NetSuite	URL
REPO_REGISTER_DEFERREVENUE	Deferred Revenue Register	/app/reporting/reportrunner.nl?reporttype=REGISTER&accttype=DeferRevenue
REPO_REGISTER_EQUITY	Equity Register	/app/reporting/reportrunner.nl?reporttype=REGISTER&accttype=Equity
REPO_REGISTER_EXPENSE	Expense Register	/app/reporting/reportrunner.nl?reporttype=REGISTER&accttype=Expense
REPO_REGISTER_FIXEDASSET	Fixed Asset Register	/app/reporting/reportrunner.nl?reporttype=REGISTER&accttype=FixedAsset
REPO_REGISTER_INCOME	Income Register	/app/reporting/reportrunner.nl?reporttype=REGISTER&accttype=Income
REPO_REGISTER_LONGTERMLIAB	Long Team Liability Register	/app/reporting/reportrunner.nl?reporttype=REGISTER&accttype=LongTermLiab
REPO_REGISTER_NONPOSTING	Non-Posting Register	/app/reporting/reportrunner.nl?reporttype=REGISTER&accttype=NonPosting
REPO_REGISTER_OTHASSET	Other Asset Register	/app/reporting/reportrunner.nl?reporttype=REGISTER&accttype=OthAsset
REPO_REGISTER_OTHCURRASSET	Other Current Asset Register	/app/reporting/reportrunner.nl?reporttype=REGISTER&accttype=OthCurrAsset
REPO_REGISTER_OTHCURRLIAB	Other Current Liability Register	/app/reporting/reportrunner.nl?reporttype=REGISTER&accttype=OthCurrLiab
REPO_REGISTER_OTHEXPENSE	Register - Other Expense	/app/reporting/reportrunner.nl?acctid=&showStartBalances=F
REPO_REGISTER_OTHINCOME	Other Income Register	/app/reporting/reportrunner.nl?reporttype=REGISTER&accttype=OthIncome
REPO_REGISTER_UNBILLEDREC	Unbilled Receivable Register	/app/reporting/reportrunner.nl?reporttype=REGISTER&accttype=UnbilledRec
SRCH_ACCOUNT	Search Account	/app/common/search/search.nl?searchtype=Account
SRCH_ACCOUNTINGBOOK	Search Accounting Book	/app/common/search/search.nl?searchtype=AccountingBook
SRCH_ACCOUNTINGPERIOD	Search Accounting Period	/app/common/search/search.nl?searchtype=AccountingPeriod
SRCH_ACCOUNTINGTRANSACTION	Search Accounting Transactions	/app/common/search/search.nl?searchtype=AccountingTransaction
SRCH_ACTIVITY	Search Activities	/app/common/search/search.nl?searchtype=Activity
SRCH_AMORTIZATIONSCHEDULE	Search Amortization Schedules	/app/common/search/search.nl?searchtype=AmortizationSchedule
SRCH_APPDEF	Search App Definitions	/app/common/search/search.nl?searchtype=AppDefinition
SRCH_APPPKG	Search App Packages	/app/common/search/search.nl?searchtype=AppPackage
SRCH_AUDITTRAIL	Search Audit Trail	/app/common/search/search.nl?searchtype=AuditTrail

Task ID	Page Label in NetSuite	URL
SRCH_BILLINGACCOUNT	Search Billing Account	/app/common/search/search.nl?searchtype=BillingAccount
SRCH_BILLINGCLASS	Search Billing Class	/app/common/search/search.nl?searchtype=BillingClass
SRCH_BILLINGRULE	Search Billing Rule	/app/common/search/search.nl?searchtype=BillingRule
SRCH_BILLINGSCHEDULE	Search Billing Schedule	/app/common/search/search.nl?searchtype=BillingSchedule
SRCH_BILLING_GROUPS	Billing Groups	/app/common/search/search.nl?searchtype=CRMGroup&adv=T&CRMGroup_GROUPTYPE=CustJob&CRMGroup_ISBILLINGGROUP=T
SRCH_BILLOFDISTRIBUTION	Search Bill Of Distribution	/app/common/search/search.nl?searchtype=BillOfDistribution
SRCH_BILLRUN	Search Billing Operations	/app/common/search/search.nl?searchtype=BillRun
SRCH_BILLRUNSCHEDULE	Search Billing Operation Schedules	/app/common/search/search.nl?searchtype=BillRunSchedule
SRCH_BIN	Search Bin	/app/common/search/search.nl?searchtype=BinNumber
SRCH_BINNUMBER	Search Bin	/app/common/search/search.nl?searchtype=BinNumber
SRCH_BUDGET	Search Set Up Budget	/app/common/search/search.nl?searchtype=Budget
SRCH_BUDGETRATES	Search Budget Rates	/app/common/search/search.nl?searchtype=BudgetExchangeRate
SRCH_CALENDAR	Search Calendar/Event	/app/common/search/search.nl?searchtype=Calendar
SRCH_CALL	Search Calls	/app/common/search/search.nl?searchtype=Call
SRCH_CAMPAIGN	Search Campaigns	/app/common/search/search.nl?searchtype=Campaign
SRCH_CAMPAIGNTEMPLATE	Search All Campaign Templates	/app/common/search/search.nl?searchtype=CampaignTemplate
SRCH_CASE	Search Case	/app/common/search/search.nl?searchtype=Case
SRCH_CHARGE	Search Charge	/app/common/search/search.nl?searchtype=Charge
SRCH_CHARGERULE	Search Charge Rule	/app/common/search/search.nl?searchtype=ChargeRule
SRCH_CLASS	Search Class	/app/common/search/search.nl?searchtype=Class
SRCH_CLASSSEGMENTMAPPING	Search Class Mapping	/app/common/search/search.nl?searchtype=ClassSegmentMapping
SRCH_COMMERCECATALOG	Search Commerce Category Catalogs	/app/common/search/search.nl?searchtype=commercecatalog

Task ID	Page Label in NetSuite	URL
SRCH_COMMERCECATEGORY	Search Commerce Categories	/app/common/search/search.nl?searchtype=commercecategory
SRCH_COMMISSIONABLEITEM	Search Commissionable Items	/app/common/search/search.nl?searchtype=Transaction
SRCH_COMMISSIONOVERVIEW	Search Commission Overview	/app/common/search/search.nl?searchtype=Transaction
SRCH_COMPANY	Search All Companies	/app/common/search/search.nl?searchtype=Company
SRCH_COMPETITOR	Search Competitors	/app/common/search/search.nl?searchtype=Competitor
SRCH_CONSOLEXCHANGERATE	Consolidated Exchange Rates	/app/common/search/search.nl?searchtype=ConsolExchangeRate
SRCH_CONSOLRATES	Search Consolidated Rates	/app/common/search/search.nl?searchtype=ConsolExchangeRate
SRCH_CONTACT	Search Contact	/app/common/search/search.nl?searchtype=Contact
SRCH_COSTCATEGORY	Cost Category	/app/common/search/search.nl?searchtype=CostCategory
SRCH_COUPONCODE	Search Coupon Code	/app/common/search/search.nl?searchtype=CouponCode
SRCH_CRMGROUP	Search Group	/app/common/search/search.nl?searchtype=CRMGroup
SRCH_CUSTBAL	Customers by Open A/R Balance	/app/common/search/searchresults.nl?searchid=-72
SRCH_CUSTOMER	Search Customer	/app/common/search/search.nl?searchtype=Customer
SRCH_CUSTOMSEGMENT	Search Custom Segments	/app/common/search/search.nl?searchtype=CustomSegment
SRCH_CUSTOVDBAL	Customers by Outstanding Balance	/app/common/search/searchresults.nl?searchid=-73
SRCH_DELETEDRECORD	Search Deleted Records	/app/common/search/search.nl?searchtype=DeletedRecord
SRCH_DEPARTMENT	Search Department	/app/common/search/search.nl?searchtype=Department
SRCH_DEPTSEGMENTMAPPING	Search Department Mapping	/app/common/search/search.nl?searchtype=DeptSegmentMapping
SRCH_DISTRIBUTIONNETWORK	Search Distribution Network	/app/common/search/search.nl?searchtype=DistributionNetwork
SRCH_DOCUMENT	Search File Cabinet	/app/common/search/search.nl?searchtype=Document
SRCH_DRIVERSLICENSE	Search Driver's Licenses	/app/common/search/search.nl?searchtype=DriversLicense
SRCH_EMAILTEMPLATE	Search All Email Templates	/app/common/search/search.nl?searchtype=EmailTemplate

Task ID	Page Label in NetSuite	URL
SRCH_EMPLOYEE	Search Employee	/app/common/search/search.nl?searchtype=Employee
SRCH_EMPLOYEEPAYROLLITEM	Search Employee Payroll Items	/app/common/search/search.nl?searchtype=EmployeePayrollItem&rectype
SRCH_ENTITY	Search All Entities	/app/common/search/search.nl?searchtype=Entity
SRCH_ENTITYACCOUNTMAPPING	Search Entity Account Mapping	/app/common/search/search.nl?searchtype=EntityAccountMapping
SRCH_FAIRVALUEFORMULA	Search Fair Value Formulas	/app/common/search/search.nl?searchtype=FairValueFormula
SRCH_FAIRVALUEPRICE	Search Fair Value Price Form	/app/common/search/search.nl?searchtype=FairValuePrice
SRCH_FAXTEMPLATE	Search All Fax Templates	/app/common/search/search.nl?searchtype=FaxTemplate
SRCH_FIRSTVISIT	Search First Site Visit	/app/common/search/search.nl?searchtype=FirstVisit
SRCH_FISCALCALENDAR	Search Fiscal Calendars	/app/common/search/search.nl?searchtype=FiscalCalendar
SRCH_FOLDER	Search Folder	/app/common/search/search.nl?searchtype=Folder
SRCH_GENERICRESOURCE	Search Generic Resources	/app/common/search/search.nl?searchtype=GenericResource
SRCH_GIFTCERTIFICATE	Search Gift Certificates	/app/common/search/search.nl?searchtype=GiftCertificate
SRCH_GLLINESAUDITLOG	Search Custom GL Lines Plug-in Audit Log	/app/common/search/search.nl?searchtype=GlLinesAuditLog
SRCH_GLOBALACCOUNTMAPPING	Search Global Account Mapping	/app/common/search/search.nl?searchtype=GlobalAccountMapping
SRCH_GOVERNMENTISSUEDIDTYPE	Search Government-Issued ID Types	/app/common/search/search.nl?searchtype=GovernmentIssuedIdType
SRCH_HCMJOB	Search HCM Job	/app/common/search/search.nl?searchtype=HCMJob
SRCH_INFOITEM	Search Information Items	/app/common/search/search.nl?id=-2540&e=T
SRCH_INFOITEMFORM	Search Published Forms	/app/common/search/search.nl?id=-2541&e=T
SRCH_INVCOSTTEMPLATE	Search Inventory Cost Template	/app/common/search/search.nl?searchtype=InvCostTemplate
SRCH_INVENTORYDETAIL	Search Inventory Detail	/app/common/search/search.nl?searchtype=InventoryDetail
SRCH_INVENTORYNUMBER	Search Inventory Numbers	/app/common/search/search.nl?searchtype=InventoryNumber
SRCH_INVENTORYNUMBERBIN	Search Inventory and Bin Numbers	/app/common/search/search.nl?searchtype=InventoryNumberBin

Task ID	Page Label in NetSuite	URL
SRCH_IPRESTRICTIONS	Login Restrictions	/app/common/search/search.nl?searchtype=IPRestrictions
SRCH_ISSUE	Search Issue	/app/common/search/search.nl?searchtype=Issue
SRCH_ITEM	Search Item	/app/common/search/search.nl?searchtype=Item
SRCH_ITEMACCOUNTMAPPING	Search Item Account Mapping	/app/common/search/search.nl?searchtype=ItemAccountMapping
SRCH_ITEMDEMANDPLAN	Search Item Demand Plan	/app/common/search/search.nl?searchtype=ItemDemandPlan
SRCH_ITEMREVENUECATEGORY	Search Item Revenue Category Form	/app/common/search/search.nl?searchtype=ItemRevenueCategory
SRCH_ITEMSUPPLYPLAN	Search Item Supply Plan	/app/common/search/search.nl?searchtype=ItemSupplyPlan
SRCH_ITEM_REVISION	Search Item Revision	/app/common/search/search.nl?searchtype=ItemRevision
SRCH_JOB	Search Job	/app/common/search/search.nl?searchtype=Job
SRCH_JOBREQUISITION	Search Job Requisition	/app/common/search/search.nl?searchtype=JobRequisition
SRCH_KUDOS	Search Kudos	/app/common/search/search.nl?searchtype=Kudos
SRCH_LEAD	Search Leads	/app/common/search/search.nl?searchtype=Customer&Customer_STAGE=LEAD
SRCH_LOCASSIGNCONF	Configurations Search	/app/common/search/search.nl?searchtype=LocAssignConf
SRCH_LOCATION	Search Location	/app/common/search/search.nl?searchtype=Location
SRCH_LOCATIONCOSTINGGROUP	Search Location Costing Group	/app/common/search/search.nl?searchtype=LocationCostingGroup
SRCH_LOCSEGMENTMAPPING	Search Location Mapping	/app/common/search/search.nl?searchtype=LocSegmentMapping
SRCH_LOGINAUDIT	Search Login Audit Trail	/app/common/search/search.nl?searchtype=LoginAudit
SRCH_MEDIAITEM	Search Media Items	/app/common/search/search.nl?searchtype=Document
SRCH_MEMDOC	Search Memorized Transactions	/app/common/search/search.nl?searchtype=MemDoc
SRCH_MESSAGE	Search Messages	/app/common/search/search.nl?searchtype=Message
SRCH_MFGCOSTTEMPLATE	Search Manufacturing Cost Template	/app/common/search/search.nl?searchtype=MfgCostTemplate
SRCH_MFGOPERATIONTASK	Search Manufacturing Operation Task	/app/common/search/search.nl?searchtype=MfgOperationTask

Task ID	Page Label in NetSuite	URL
SRCH_MFGPLANNEDTIME	Search Manufacturing Planned Time	/app/common/search/search.nl?searchtype=MfgPlannedTime
SRCH_MFGROUTING	Search Manufacturing Routing	/app/common/search/search.nl?searchtype=MfgRouting
SRCH_NEXUS	Search Nexus	/app/common/search/search.nl?searchtype=Nexus
SRCH_NOTIFICATION	Search Notification	/app/common/search/search.nl?searchtype=Notification
SRCH_OAUTH_TOKENS	Search Access Tokens	/app/common/search/search.nl?searchtype=OAuthToken
SRCH_ONLINECASEFORM	Search Online Case Forms	/app/common/search/search.nl?searchtype=OnlineCaseForm
SRCH_ONLINELEADFORM	Search Online Lead Forms	/app/common/search/search.nl?searchtype=OnlineLeadForm
SRCH_ORGANIZATIONVALUE	Search Organization Values	/app/common/search/search.nl?searchtype=OrganizationValue
SRCH_OTHERGOVERNMENTISSUEDID	Search Other Government-Issued IDs	/app/common/search/search.nl?searchtype=OtherGovernmentIssuedId
SRCH_PARTNER	Search Partner	/app/common/search/search.nl?searchtype=Partner
SRCH_PASSPORT	Search Passports	/app/common/search/search.nl?searchtype=Passport
SRCH_PAYMENTEVENT	Search Payment Events	/app/common/search/search.nl?searchtype=PaymentEvent
SRCH_PAYMETHODS	Search Payment Methods	/app/common/search/search.nl?searchtype=PaymentMethod
SRCH_PAYROLLITEM	Search Payroll Item	/app/common/search/search.nl?searchtype=PayrollItem
SRCH_PAYTERMS	Search Payment Terms	/app/common/search/search.nl?searchtype=Term
SRCH_PLANNEDSTANDARDCOST	Search Planned Standard Cost	/app/common/search/search.nl?searchtype=PlannedStandardCost
SRCH_POSITION	Search Position	/app/common/search/search.nl?searchtype=Position
SRCH_PRESCATEGORY	Search Categories	/app/common/search/search.nl?searchtype=SiteCategory
SRCH_PRICING	Search Pricing	/app/common/search/search.nl?searchtype=Pricing
SRCH_PROJECTBUDGET	Search Project Budget	/app/common/search/search.nl?searchtype=ProjectBudget
SRCH_PROJECTEXPENSETYPE	Search Project Expense Types	/app/common/search/search.nl?searchtype=ProjectExpenseType
SRCH_PROJECTRESOURCE	Search Project Resources	/app/common/search/search.nl?searchtype=ProjectResource

Task ID	Page Label in NetSuite	URL
SRCH_PROJECTTASK	Search Project Tasks	/app/common/search/search.nl?searchtype=ProjectTask
SRCH_PROJECTTASKANDCRMTASK	Search Project Tasks and CRM Tasks	/app/common/search/search.nl?searchtype=ProjectTaskAndCrmTask
SRCH_PROJECTTEMPLATE	Search Project Template	/app/common/search/search.nl?searchtype=ProjectTemplate
SRCH_PROMOTION	Search Promotion	/app/common/search/search.nl?searchtype=Promotion
SRCH_PROSPECT	Search Prospects	/app/common/search/search.nl?searchtype=Customer&Customer_STAGE=PROSPECT
SRCH_QUOTA	Search Establish Quota	/app/common/search/search.nl?searchtype=Quota
SRCH_RATEPLAN	Search Rate Plan	/app/common/search/search.nl?searchtype=RatePlan
SRCH_REGION	Search Region	/app/common/search/search.nl?searchtype=Region
SRCH_REPORT	Search Reports	/app/common/search/search.nl?searchtype=Report
SRCH_REVARRNGMESSAGE	Search Revenue Arrangement Message	/app/common/search/search.nl?searchtype=RevArrngMessage
SRCH_REVENUEALLOCATIONGROUP	Search Revenue Allocation Group	/app/common/search/search.nl?searchtype=RevAllocationGroup
SRCH_REVENUEELEMENT	Search Revenue Element	/app/common/search/search.nl?searchtype=RevenueElement
SRCH_REVENUEPLAN	Search Revenue Recognition Plan Form	/app/common/search/search.nl?searchtype=RevenuePlan
SRCH_REVENUERECOGNITIONRULE	Search Revenue Recognition Rule Form	/app/common/search/search.nl?searchtype=RevRecRule
SRCH_REVREC	Search Revenue Recgonition Schedules	/app/common/search/search.nl?searchtype=RevRecTemplate
SRCH_REVRECFIELDMAPPING	Search Revenue Field Mappings	/app/common/search/search.nl?searchtype=RevRecFieldMapping
SRCH_REVRECOGNITIONSCHED	Search Revenue Recgonition Schedules	/app/common/search/search.nl?searchtype=RevRecognitionSched
SRCH_ROLE	Search Role	/app/common/search/search.nl?searchtype=Role
SRCH_RSRCALLOCATION	Search Rsrc Allocations	/app/common/search/search.nl?searchtype=RsrcAllocation
SRCH_SALESCAMPAIGN	Search Sales Campaigns	/app/common/search/search.nl?searchtype=SalesCampaign
SRCH_SALESTERRITORIES	Sales Territory Search	/app/common/search/search.nl?searchtype=SalesTerritory
SRCH_SAVEDSEARCH	All Saved Searches	/app/common/search/savedsearchlist.nl
SRCH_SCHEDULEDSCRIPTINSTANCE	Search Scheduled Script Instance	/app/common/search/search.nl?searchtype=ScheduledScriptInstance

Task ID	Page Label in NetSuite	URL
SRCH_SCRIPT	Search Script	/app/common/search/search.nl?searchtype=Script
SRCH_SCRIPTDEPLOYMENT	Search Script Deployment	/app/common/search/search.nl?searchtype=ScriptDeployment
SRCH_SCRIPTNOTE	Search Script Execution Log	/app/common/search/search.nl?searchtype=ScriptNote
SRCH_SHIPITEM	Search Ship Item	/app/common/search/search.nl?searchtype=ShipItem
SRCH_SHIPMENTPACKAGE	Search Shipping Packages	/app/common/search/search.nl?searchtype=ShipmentPackage
SRCH_SHIPPARTPACKAGE	Search Shipping Partner Package	/app/common/search/search.nl?searchtype=ShipPartPackage
SRCH_SHIPPARTREGISTRATION	Search Shipping Partner Registration	/app/common/search/search.nl?searchtype=ShipPartRegistration
SRCH_SHIPPARTSHIPMENT	Search Shipping Partner Shipment	/app/common/search/search.nl?searchtype=ShipPartShipment
SRCH_SHOPPINGCART	Search Shopping Cart	/app/common/search/search.nl?searchtype=ShoppingCart
SRCH_SOLUTION	Search Solution	/app/common/search/search.nl?searchtype=Solution
SRCH_STANDARDCOSTVERSION	Search Standard Cost Version	/app/common/search/search.nl?searchtype=StandardCostVersion
SRCH_SUBSCRIPTION	Search Subscriptions	/app/common/search/search.nl?searchtype=Subscription
SRCH_SUBSCRIPTIONCHANGEORDER	Search Change Orders	/app/common/search/search.nl?searchtype=SubscriptionChangeOrder
SRCH_SUBSCRIPTIONLINE	Search Subscription Lines	/app/common/search/search.nl?searchtype=SubscriptionLine
SRCH_SUBSCRIPTIONPLAN	Search Subscription Plan	/app/common/search/search.nl?searchtype=Item&Item_TYPE=SubscriPlan&adv=T
SRCH_SUBSIDIARY	Search Subsidiary	/app/common/search/search.nl?searchtype=Subsidiary
SRCH_SYSTEMEMAILTEMPLATE	Search All System Email Templates	/app/common/search/search.nl?searchtype=SystemEmailTemplate
SRCH_SYSTEMNOTE	Search System Notes	/app/common/search/search.nl?searchtype=SystemNote
SRCH_TASK	Search Tasks	/app/common/search/search.nl?searchtype=Task
SRCH_TAXDETAIL	Search Tax Detail	/app/common/search/search.nl?searchtype=TaxDetail
SRCH_TAXGROUP	Search Tax Group	/app/common/search/search.nl?searchtype=TaxGroup
SRCH_TAXITEM	Search Tax Code	/app/common/search/search.nl?searchtype=TaxItem

Task ID	Page Label in NetSuite	URL
SRCH_TAXTYPE	Search Tax Type	/app/common/search/search.nl?searchtype=TaxType
SRCH_TERMINATIONREASON	Search Termination Reason	/app/common/search/search.nl?searchtype=TerminationReason
SRCH_TIME	Search Track Time	/app/common/search/search.nl?searchtype=Time
SRCH_TIMEENTRY	Search Time Entry	/app/common/search/search.nl?searchtype=Timeentry
SRCH_TIMEOFFCHANGE	Search Time-Off Changes	/app/common/search/search.nl?searchtype=TimeOffChange
SRCH_TIMEOFFPLAN	Search Time-Off Plans	/app/common/search/search.nl?searchtype=TimeOffPlan
SRCH_TIMEOFFREQUEST	Time-Off Requests	/app/common/search/search.nl?searchtype=TimeOffRequest
SRCH_TIMEOFFTYPE	Search Time-Off Types	/app/common/search/search.nl?searchtype=TimeOffType
SRCH_TIMESHEET	Search Timesheet	/app/common/search/search.nl?searchtype=Timesheet
SRCH_TIMESHEETAPPROVAL	Search Timesheet Approval	/app/common/search/search.nl?searchtype=TimesheetApproval
SRCH_TOPIC	Search Topic	/app/common/search/search.nl?searchtype=Topic
SRCH_TRANNUMBERAUDITLOG	Search Transaction Numbering Audit Log	/app/common/search/search.nl?searchtype=TranNumberAuditLog
SRCH_TRANSACTION	Search Transactions	/app/common/search/search.nl?searchtype=Transaction
SRCH_TRAN_BINTRNFR	Search Bin Transfers	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=BinTrnfr
SRCH_TRAN_BINWKSHT	Search Bin Putaway Worksheets	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=BinWksht
SRCH_TRAN_BLANKORD	Search Blanket Purchase Orders	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=BlankOrd
SRCH_TRAN_BOOKJOURNAL	Search Accounting Book Journal Entries	/app/common/search/search.nl?searchtype=AccountingTransaction&AccountingTransaction_TYPE=Journal&AccountingTransaction_BOOKSPECIFICTRANSACTION=T
SRCH_TRAN_BUILD	Search Assembly Builds	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Build
SRCH_TRAN_CARDCHRG	Search Credit Card Charges	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=CardChrg
SRCH_TRAN_CASHRFND	Search Cash Refunds	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=CashRfnd
SRCH_TRAN_CASHSALE	Search Cash Sales	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=CashSale
SRCH_TRAN_CHECK	Search Checks	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Check

Task ID	Page Label in NetSuite	URL
SRCH_TRAN_COMMISSN	Search Individual Employee Commissions	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Commissn
SRCH_TRAN_CUSTCHRG	Search Statement Charges	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=CustChrg
SRCH_TRAN_CUSTCRED	Search Credit Memos	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=CustCred
SRCH_TRAN_CUSTDEP	Search Customer Deposits	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=CustDep
SRCH_TRAN_CUSTINVC	Search Invoices	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=CustInvc
SRCH_TRAN_CUSTPYMT	Search Customer Payments	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=CustPymt
SRCH_TRAN_CUSTRFND	Search Customer Refunds	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=CustRfnd
SRCH_TRAN_DEPAPPL	Search Deposit Applications	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=DepAppl
SRCH_TRAN_DEPOSIT	Search Deposits	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Deposit
SRCH_TRAN_ESTIMATE	Search Estimates	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Estimate
SRCH_TRAN_EXPREPT	Search Expense Reports	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=ExpRept
SRCH_TRAN_FFTREQ	Search Fulfillment Requests	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=FftReq
SRCH_TRAN_FXREVAL	Search Currency Revaluations	/app/accounting/transactions/transactionsearchswitch.nl?searchtype=AccountingTransaction&AccountingTransaction_TYPE=FxReval
SRCH_TRAN_INVADJST	Search Inventory Adjustments	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=InvAdjst
SRCH_TRAN_INVCOUNT	Search Inventory Counts	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=InvCount
SRCH_TRAN_INVDISTR	Search Inventory Distributions	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=InvDistr
SRCH_TRAN_INVREVAL	Search Inventory Cost Revaluations	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=InvReval
SRCH_TRAN_INVTRNFR	Search Inventory Transfers	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=InvTrnfr
SRCH_TRAN_INVWKSHT	Search Inventory Worksheets	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=InvWksht
SRCH_TRAN_ITEMRCPT	Search Item Receipts	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=ItemRcpt
SRCH_TRAN_ITEMSHIP	Search Item Fulfillments	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=ItemShip
SRCH_TRAN_JOURNAL	Search Journal Entries	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Journal

Task ID	Page Label in NetSuite	URL
SRCH_TRAN_LIABPYMT	Search Liability Payments	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=LiabPymt
SRCH_TRAN_OPPRTNTY	Search Opportunities	/app/common/search/search.nl?searchtype=Opprtnty
SRCH_TRAN_PAYCHECK	Search Paychecks	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Paycheck
SRCH_TRAN_PCHKJRNL	Search Paycheck Journals	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=PChkJrnl
SRCH_TRAN_PURCHCON	Search Purchase Contracts	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=PurchCon
SRCH_TRAN_PURCHORD	Search Purchase Orders	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=PurchOrd
SRCH_TRAN_PURCHREQ	Search Requisitions	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=PurchReq
SRCH_TRAN_REVARRNG	Search Revenue Arrangements	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=RevArrng
SRCH_TRAN_REVCOMM	Search Revenue Commitments	/app/accounting/transactions/transactionsearchswitch.nl?searchtype=AccountingTransaction&AccountingTransaction_TYPE=RevComm
SRCH_TRAN_REVCOMRV	Search View Revenue Commitment Reversals	/app/accounting/transactions/transactionsearchswitch.nl?searchtype=AccountingTransaction&AccountingTransaction_TYPE=RevComRv
SRCH_TRAN_REVCONTR	Search View Revenue Contracts	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=RevContr
SRCH_TRAN_RFQ	Search Requests For Quote	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Rfq
SRCH_TRAN_RTNAUTH	Search Return Authorizations	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=RtnAuth
SRCH_TRAN_SALESORD	Search Sales Orders	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=SalesOrd
SRCH_TRAN_STATJOURNAL	Search Statistical Journals	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Journal&Transaction_STATISTICAL=T
SRCH_TRAN_STPICKUP	Search Store Pickup Fulfillments	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=StPickUp
SRCH_TRAN_TAXLIAB	Search Tax Liabilities	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=TaxLiab
SRCH_TRAN_TAXPYMT	Search Tax Payments	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=TaxPymt
SRCH_TRAN_TEGPYBL	Search Issued Tegatas	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=TegPybl
SRCH_TRAN_TEGRCVBL	Search Received Tegatas	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=TegRcvbl

Task ID	Page Label in NetSuite	URL
SRCH_TRAN_TRANSFER	Search Bank Transfers	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Transfer
SRCH_TRAN_TRNFRORD	Search Transfer Orders	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=TrnfrOrd
SRCH_TRAN_UNBUILD	Search Assembly Unbuilds	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Unbuild
SRCH_TRAN_VENDAUTH	Search Vendor Return Authorizations	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=VendAuth
SRCH_TRAN_VENDBILL	Search Bills	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=VendBill
SRCH_TRAN_VENDCRED	Search Bill Credits	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=VendCred
SRCH_TRAN_VENDPYMT	Search Bill Payments	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=VendPymt
SRCH_TRAN_VENDRFQ	Search Vendor Requests For Quote	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=VendRfq
SRCH_TRAN_WOCLOSE	Search Work Order Closes	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=WOClose
SRCH_TRAN_WOCOMPL	Search Work Order Completions	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=WOCompl
SRCH_TRAN_WOISSUE	Search Work Order Issues	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=WOIssue
SRCH_TRAN_WORKORD	Search Work Orders	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=WorkOrd
SRCH_TRAN_YTDADJST	Search Payroll Adjustment	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=YtdAdjst
SRCH_TWOFACTORDEVICE	Two-Factor Tokens	/app/common/search/search.nl?searchtype=TwoFactorDevice
SRCH_UNBILLORDERS	Customers by Unbilled Orders	/app/common/search/searchresults.nl?searchid=-74
SRCH_UNITSTYPE	Search Unit of Measure	/app/common/search/search.nl?searchtype=UnitsType
SRCH_USAGE	Search Usage	/app/common/search/search.nl?searchtype=Usage
SRCH_USERNOTE	Search User Notes	/app/common/search/search.nl?searchtype=UserNote
SRCH_VENDOR	Search Vendor	/app/common/search/search.nl?searchtype=Vendor
SRCH_WEEKLYTIMESHEET	Search Weekly Time Tracking	/app/common/search/search.nl?searchtype=WeeklyTimeSheet
SRCH_WORKCALENDAR	Search Work Calendar	/app/common/search/search.nl?searchtype=WorkCalendar
SRCH_WORKFLOW	Search Workflow	/app/common/search/search.nl?searchtype=Workflow

Task ID	Page Label in NetSuite	URL
SRCH_WORKFLOWINSTANCE	Search Workflow Instances	/app/common/search/search.nl?searchtype=WorkflowInstance
SRCH_WORKPLACE	Search Workplace	/app/common/search/search.nl?searchtype=Workplace
SUPT_ALLOW_LOGIN	NetSuite Support Login	/app/crm/support/allowsupportlogin.nl
SUPT_CENTER_ROLE	NetSuite Support Center	/app/login/changerole.nl?id=
TRAN_ACTIVITY	Activity	/app/crm/calendar/activity.nl
TRAN_ADDCONTENT	Add Content	/app/center/setup/addcontent.nl
TRAN_ADDSHORTCUT	Short Cuts	/core/pages/addShortcut.nl
TRAN_ALLOCATEPAYCHECKSTOJOBS	Allocate Paycheck Expenses to Jobs	/app/accounting/transactions/allocatepaycheckstojobs.nl
TRAN_ALLOCATEREVENUEARRANGEMENTS	Allocate Revenue Arrangements	/app/accounting/revrec/allocaterevenuearrangements.nl
TRAN_ALLOCATEREVENUEARRANGEMENTS_STATUS	Allocate Revenue Arrangements Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=ALLOCATEREVARRANGEMENT&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_ALLOCATIONBATCH	Allocation Batch Status	/app/accounting/transactions/allocationbatchstatus.nl
TRAN_APPROVAL_EXPREPT	Approve Expense Reports	/app/accounting/transactions/approveexpensereport.nl
TRAN_APPROVAL_PURCHORD	Approve Purchase Requests	/app/accounting/transactions/approval.nl?type=purchord&label=Purchase%20Request
TRAN_APPROVECOMMISSN	Approve Employee Commissions	/app/accounting/transactions/approvecommissn.nl
TRAN_APPROVEPARTNERCOMMISSN	Approve Partner Commissions	/app/accounting/transactions/approvepartnercommissn.nl
TRAN_AUDIT	View Audit Trail	/app/accounting/transactions/audit.nl
TRAN_BALANCELCGACCOUNTS	Balance Location Costing Group Accounts	/app/accounting/inventory/balancelcgaccounts.nl
TRAN_BALANCELCGACCOUNTS_STATUS	Balance Location Costing Group Accounts Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=BALANCELCGACCOUNTS&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_BANKRECON	Find Matching Transactions	/app/accounting/transactions/bankrecon.nl
TRAN_BANKVIEW	Online Banking Statement	/app/accounting/transactions/bankview.nl
TRAN_BAS	Business Activity Statement	/app/accounting/reports/intl/bas.nl
TRAN_BATCHCHECK	Batch Check	/app/payroll/batchcheck.nl
TRAN_BILLOFMATERIALSINQUIRY	Bill of Materials Inquiry	/app/accounting/transactions/inventory/billofmaterialsinquiry.nl

Task ID	Page Label in NetSuite	URL
TRAN_BILLPAY_LOG	Job Status	/app/external/xml/upload/uploadlog.nl?displayType=BILLPAY
TRAN_BILLRUN	Process Billing Operations	/app/accounting/transactions/billingworkcenter/billrun.nl
TRAN_BILLRUNRESULT	View Billing Operations	/app/accounting/transactions/billingworkcenter/billrunresults.nl
TRAN_BLANKORDAPPRV	Approve Blanket Purchase Orders	/app/accounting/transactions/transactionapproval.nl?type=BlankOrd
TRAN_BUDGET	Set Up Budgets	/app/accounting/transactions/budgets.nl
TRAN_BULKAUTHCOMMISSN_LOG	Job Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=BULKAUTHCOMMISSN&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_BULKAUTHPARTNERCOMMISSN_LOG	Job Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=BULKAUTHPARTNERCOMMISSN&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_BULKBILL_LOG	Job Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=BILLSALESORDERS&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_BULKCOMMITREVENUE_LOG	Job Status	/app/external/xml/upload/uploadlog.nl?displayType=BULKCOMMITREVENUE
TRAN_BULKFULFILL_LOG	Job Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=FULFILLSALESORDERS&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_BULKINVOICE_LOG	Job Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=INVOICEBILLABLECUSTOMERS&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_BULKITEMSHIPSTATUSPACK_LOG	Mark Orders Packed Job Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=BULKITEMSHIPSTATUSPACK&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_BULKITEMSHIPSTATUSSHIP_LOG	Mark Orders Shipped Job Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=BULKITEMSHIPSTATUSSHIP&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_BULKRECEIVE_LOG	Process Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=BULKRECEIVEORDER&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_BULKREVENUECONTRACT_LOG	Job Status	/app/external/xml/upload/uploadlog.nl?displayType=BULKREVENUECONTRACT
TRAN_CALCULATEITEMDEMANDPLAN	Calculate Item Demand Plan	/app/accounting/inventory/demandplanning/calculateitemdemandplan.nl
TRAN_CALCULATEITEMDEMANDPLAN_STATUS	Calculate Item Demand Plan Status	/app/accounting/inventory/demandplanning/calculateitemdemandplanstatus.nl

Task ID	Page Label in NetSuite	URL
TRAN_CALENDARPREFERENCE	Calendar Preferences	/app/crm/calendar/calendarpreference.nl
TRAN_CAMPAIGNSETUP	Marketing Preferences	/app/setup/campaignsetup.nl
TRAN_CHARGEMANAGER	Manage Charge Stages	/app/accounting/transactions/billing/chargemanager.nl
TRAN_CHECKITEMAVAILABILITY	Check Item Availability	/app/accounting/inventory/atp/checkitemavailability.nl
TRAN_CLEARHOLD	Manage Payment Holds	/app/accounting/transactions/salesordermanager.nl?type=clearholds
TRAN_CLOSEWORKORDERS_LOG	Close Work Orders Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=CLOSEWORKORDERS&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_COMPLETEWORKORDERS_LOG	Enter Completions Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=COMPLETEWORKORDERS&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_COMPONENTWHEREUSEDINQUIRY	Component Where Used Inquiry	/app/accounting/transactions/manufacturing/componentwhereusedinquiry.nl
TRAN_COPY_BUDGET	Copy Budgets	/app/accounting/transactions/copybudget.nl
TRAN_COSTEDBOMINQUIRY	Costed Bill of Materials Inquiry	/app/accounting/transactions/manufacturing/costedbillofmaterialsinquiry.nl
TRAN_CREATEAMORTIZATIONJE	Create Amortization Journal Entries	/app/accounting/transactions/createamortizationje.nl
TRAN_CREATEAMORTIZATIONJE_LOG	Create Journal Entries Status	/app/accounting/transactions/createrecognitionjestatus.nl?type=AMORTIZATION
TRAN_CREATEDEGROSSJE	Create Deferred Revenue Reclassification Journal Entries	/app/accounting/transactions/createdegrossje.nl
TRAN_CREATEDEGROSSJE_LOG	Create Reclassification Journal Entries Status	/app/accounting/transactions/createdegrossjestatus.nl
TRAN_CREATEINTERCOADJJE	Create Intercompany Adjustment Journal Entries	/app/accounting/transactions/intercoadj/createintercoadjje.nl
TRAN_CREATEINTERCOADJJE_LOG	Create Journal Entries Status	/app/accounting/transactions/intercoadj/createintercoadjjestatus.nl
TRAN_CREATENEXTGENRECLASSJE	Create Deferred Revenue Reclassification Journal Entries	/app/accounting/transactions/createdegrossje.nl
TRAN_CREATENEXTGENRECLASSJE_LOG	Create Reclassification Journal Entries Status	/app/accounting/transactions/createreclassjournalstatus.nl?bulkproctype=DEFERREDREVENUERECLASS&ReclassificationStatus_CREATEDDATE=TODAY
TRAN_CREATEREVRECJE	Create Revenue Recognition Journal Entries	/app/accounting/transactions/createrevrecje.nl
TRAN_CREATEREVRECJE_LOG	Create Journal Entries Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=BULKREVREC&bulkproctype=RECOGNIZEREVENUE&bulkp

Task ID	Page Label in NetSuite	URL
		roctype=RECOGNIZEPLANNEDREVENUE&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_CREATE_INVCOUNT	Create Inventory Count	/app/accounting/transactions/createinvcount.nl
TRAN_CREATE_JOBS_FROM_ORDERS	Create Jobs From Sales Orders	/app/accounting/transactions/jobcreationmanager.nl
TRAN_CREATE_WORK_ORDERS_FOR_STOCK	Mass Create Work Orders	/app/accounting/transactions/createworkordersforstock.nl
TRAN_CREATE_WORK_ORDERS_FOR_STOCK_DEMAND_PLANNING	Mass Create Work Orders	/app/accounting/transactions/createworkordersforstock.nl
TRAN_CREATE_WORK_ORDERS_FOR_STOCK_LOG	Process Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=CREATEWORKORDERSFORSTOCK&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_CUSTCATEGORY	Translate Category	/app/common/custom/custcategory.nl
TRAN_CUSTINVCAPPRV	Approve Invoices	/app/accounting/transactions/transactionapproval.nl?type=CustInvc
TRAN_CUSTOMIZEITEMLIST	Customize Item List	/app/accounting/transactions/customizeitemlist.nl
TRAN_CUSTOMIZETRANLIST	Customize Transaction List	/app/accounting/transactions/customizetranlist.nl
TRAN_DEPOSITSUMMARY	Deposit Summary	/app/accounting/transactions/depositsummary.nl
TRAN_DOMAINS	Set Up Domains	/app/setup/domains.nl
TRAN_DOMAINSADV	Set Up Domains	/app/setup/domains.nl
TRAN_EMAILPWD	Change Email Password	/app/center/emailpwd.nl
TRAN_EMPLOYEESFA	Assign Reps	/app/common/entity/employeesfa.nl
TRAN_FINCHRG	Assess Finance Charges	/app/accounting/transactions/finchrg.nl
TRAN_FORECAST	Edit Sales Rep Forecast	/app/crm/sales/forecast.nl
TRAN_FXREVAL_STATUS	Revalue Open Currency Balances Status	/app/accounting/transactions/fxrevalstatus.nl
TRAN_GENERATEFISCALPERIODS	Generate Fiscal Periods	/app/setup/period/generatefiscalperiods.nl
TRAN_GENERATEITEMSUPPLYPLAN	Generate Item Supply Plan	/app/accounting/inventory/demandplanning/generateitemsupplyplan.nl
TRAN_GENERATEITEMSUPPLYPLAN_STATUS	Generate Item Supply Plan Status	/app/accounting/inventory/demandplanning/generateitemsupplyplanstatus.nl
TRAN_GENERATETAXPERIODS	Generate Tax Periods	/app/setup/period/generatetaxperiods.nl
TRAN_GIFTCERTCREATEJE	Recognize Gift Certificate Income	/app/accounting/transactions/giftcertcreateje.nl
TRAN_GLNUMSTATUS	GL Audit Numbering Status	/app/accounting/transactions/glnumbering/glnumstatus.nl
TRAN_GSTREFUND	Process GST/HST Refund	/app/accounting/transactions/gstrefund.nl

Task ID	Page Label in NetSuite	URL
TRAN_HISTORY	History	/app/accounting/transactions/history.nl
TRAN_IMPACT	GL Impact	/app/accounting/transactions/impact.nl
TRAN_INVOICECUSTOMERS	Invoice Billable Customers	/app/accounting/transactions/invoicecustomers.nl
TRAN_ISSUEWORKORDERS_LOG	Issue Components Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=ISSUEWORKORDERS&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_ITEMGROSSREQUIREMENTS	Gross Requirements Inquiry	/app/accounting/inventory/demandplanning/itemgrossrequirements.nl
TRAN_ITEMSHIPPACK	Mark Orders Packed	/app/accounting/transactions/itemshipmanager.nl?type=pack
TRAN_ITEMSHIPSHIP	Mark Orders Shipped	/app/accounting/transactions/itemshipmanager.nl?type=ship
TRAN_JOURNALAPPROVAL	Approve Journal Entries	/app/accounting/transactions/journalapproval.nl
TRAN_LASTLOGIN	My Login Audit Portlet	/app/setup/lastlogin.nl
TRAN_LOGINAUDIT	Login Audit	/app/setup/loginAudit.nl
TRAN_MANAGEARRANGEMENTS	Update Revenue Arrangements	/app/accounting/revrec/managearrangementsandplans.nl?type=arrangement
TRAN_MANAGEREVENUEARRANGEMENTS	Update Revenue Arrangements and Revenue Recognition Plans	/app/accounting/bulkprocessing/managearrangementsandplansstatus.nl?bulkproctype=MANAGEREVENUEARRANGEMENT&bulkproctype=MANAGEREVENUEELEMENTS&bulkproctype=DELETEREVENUEELEMENTS&bulkproctype=MANAGEACTUALREVENUEPLANS&bulkproctype=MANAGEFORECASTREVPLANS&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_MANAGEREVENUEARRANGEMENTS_STATUS	Update Revenue Arrangements and Revenue Plans Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=MANAGEREVENUEARRANGEMENT&bulkproctype=MANAGEREVENUEELEMENTS&bulkproctype=DELETEREVENUEELEMENTS&bulkproctype=MANAGEACTUALREVENUEPLANS&bulkproctype=MANAGEFORECASTREVPLANS&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_MANAGEREVENUEPLANS	Update Revenue Recognition Plans	/app/accounting/revrec/managearrangementsandplans.nl?type=plan
TRAN_MARKBUILTWORKORDERS_LOG	Mark Work Orders Built Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=MARKBUILTWORKORDERS&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_MARKVSOEDELIVERED	Mark VSOE Delivered Status	/app/accounting/transactions/revrec/markvsoedelivered.nl
TRAN_MARKVSOEDELIVERED_LOG	Mark VSOE Delivered Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=MARKVSOEDELIVERED&BulkProcSubmission_CREATEDDATE=TODAY

Task ID	Page Label in NetSuite	URL
TRAN_MERGEREVENUEARRANGEMENTS	Merge Revenue Arrangements for Linked Sources	/app/accounting/revrec/mergerevenuearrangements.nl
TRAN_MERGEREVENUEARRANGEMENTS_STATUS	Merge Revenue Arrangements for Linked Sources Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=MERGEREVARRANGEMENT&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_MGRFORECAST	Edit Sales Manager Forecast	/app/crm/sales/mgrforecast.nl
TRAN_NLVENDOR	Vendor Center	/app/center/nlvendor.nl.nl
TRAN_OPENBAL	Enter Opening Balances	/app/accounting/transactions/openbal.nl
TRAN_ORDERDEMANDPLANITEMS	Order Items	/app/accounting/transactions/orderitems.nl
TRAN_ORDERITEMS	Order Items	/app/accounting/transactions/orderitems.nl
TRAN_ORDERITEMS_LOG	Order Items Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=BULKORDERITEMS&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_ORDERPURCHREQ_LOG	Order Requisitions Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=BULKORDERREQUISITIONS&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_ORDER_REALLOCATION	Commit Orders	/app/accounting/transactions/orderreallocation.nl
TRAN_ORDER_REALLOCATION_LOG	Commit Orders Job Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=ORDERREALLOCATION&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_PAYMENTS	Payments	/app/accounting/transactions/payments.nl
TRAN_PAYROLLBATCH	Payroll Batch	/app/payroll/payrollbatch.nl
TRAN_PAYROLLRUN	Create Payroll	/app/accounting/transactions/payroll/payrollrun.nl
TRAN_PAYROLLSTATUS	View Payroll Status	/app/payroll/cdstatus.nl
TRAN_PDF_F940	Annual Federal Unemployment (940)	/app/accounting/reports/taxformfederal.nl?formtype=f940
TRAN_PDF_F941	Quarterly Federal Tax Return (941)	/app/accounting/reports/taxformfederal.nl?formtype=f941
TRAN_PLANNEDSTANDARDCOSTROLLUP	Planned Standard Cost Rollup	/app/accounting/inventory/plannedstandardcostrollup.nl
TRAN_PLANNEDSTANDARDCOSTROLLUP_STATUS	Planned Standard Cost Rollup Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=ROLLUPITEMCOST&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_POSTVENDORBILLVARIANCES	Post Vendor Bill Variances	/app/accounting/transactions/vendorbillvariance/postvendorbillvariances.nl
TRAN_POSTVENDORBILLVARIANCES_STATUS	Post Vendor Bill Variances Status	/app/accounting/transactions/vendorbillvariance/postvendorbillvariancesstatus.nl

Task ID	Page Label in NetSuite	URL
TRAN_PREVIEWW2	Form W-2 Preview	/app/accounting/reports/w2preview.nl
TRAN_PRINT	Print Checks and Forms	/app/accounting/print/print.nl
TRAN_PRINT1096	Form 1096	/app/accounting/reports/NLPrint1096s.nl?mode=frame
TRAN_PRINT1099	Form 1099-MISC	/app/accounting/reports/NLPrint1099s.nl?mode=frame
TRAN_PRINTBARCODES	Generate Barcodes	/app/accounting/print/printbarcodes.nl?printtype=null&method=print
TRAN_PRINTMAILINGLABELS	Print Mailing Labels	/app/accounting/print/printmailinglabels.nl?printtype=null&method=print
TRAN_PRINTPRICELIST	Individual Price List	/app/accounting/print/printpricelist.nl
TRAN_PRINTSTATEMENT	Individual Statement	/app/accounting/print/printstatement.nl
TRAN_PRINTW2	Form W-2	/app/accounting/reports/NLPrintW2s.nl?mode=frame
TRAN_PRINTW2AUDIT	W-2 and 1099 Audit Information	/app/accounting/reports/printw2audit.nl?mode=frame
TRAN_PRINTW3	Form W-3	/app/accounting/reports/NLPrintW3s.nl?mode=frame
TRAN_PRINT_BOM	Print Bills of Materials	/app/accounting/print/printframe.nl?tranype=workord&printtype=bom&method=print
TRAN_PRINT_CASHSALE	Print Receipts	/app/accounting/print/printframe.nl?tranype=cashsale&printtype=transaction&method=print
TRAN_PRINT_CHECK	Print Checks	/app/accounting/print/printframe.nl?tranype=check&printtype=transaction&method=print
TRAN_PRINT_COMMERCIALINVOICE	Print Commerical Invoice	/app/accounting/print/commericalinvoice.nl
TRAN_PRINT_CUSTCRED	Print Credits Memos	/app/accounting/print/printframe.nl?tranype=custcred&printtype=transaction&method=print
TRAN_PRINT_CUSTINVC	Print Invoices	/app/accounting/print/printframe.nl?tranype=custinvc&printtype=transaction&method=print
TRAN_PRINT_ESTIMATE	Print Estimates	/app/accounting/print/printframe.nl?tranype=estimate&printtype=transaction&method=print
TRAN_PRINT_INTEGRATEDSHIPPINGLABEL	Print Intetgrated Shipping Labels	/app/accounting/print/printlabels.nl?printtype=integratedshippinglabel&method=print&title=Integrated Shipping Labels
TRAN_PRINT_ITEM_DETAIL_STATEMENT	Generate Item Detail Statements	/app/accounting/print/printframe.nl?tranype=&printtype=itemdetailstatement&method=print
TRAN_PRINT_ONE_ITEM_DETAIL_STATEMENT	Individual Item Detail Statement	/app/accounting/print/printitemdetailstatement.nl

Task ID	Page Label in NetSuite	URL
TRAN_PRINT_PACKINGSLIP	Print Packing Slips	/app/accounting/print/printframe.nl?tranype=&printtype=packingslip&method=print
TRAN_PRINT_PAYCHECK	Print Paychecks	/app/accounting/print/printframe.nl?tranype=paycheck&printtype=transaction&method=print
TRAN_PRINT_PAYMENTVOUCHER	Print Payment Vouchers	/app/accounting/print/printframe.nl?tranype=vendpymt&printtype=paymentvoucher&method=print
TRAN_PRINT_PICKINGTICKET	Print Picking Tickets	/app/accounting/print/printframe.nl?tranype=salesord&printtype=pickingticket&method=print
TRAN_PRINT_PRICELIST	Generate Price Lists	/app/accounting/print/printframe.nl?tranype=&printtype=pricelist&method=print
TRAN_PRINT_PURCHORD	Print Purchase Orders	/app/accounting/print/printframe.nl?tranype=purchord&printtype=transaction&method=print
TRAN_PRINT_RTNAUTH	Return Authorizations	/app/accounting/print/printform.nl?printtype=transaction&trantype=rtnauth&method=print&title=Return%20Authorizations
TRAN_PRINT_SALESORD	Print Sales Orders	/app/accounting/print/printframe.nl?tranype=salesord&printtype=transaction&method=print
TRAN_PRINT_SHIPPINGLABEL	Print Shipping Labels	/app/accounting/print/printframe.nl?tranype=&printtype=shippinglabel&method=print
TRAN_PRINT_STATEMENT	Generate Statements	/app/accounting/print/printframe.nl?tranype=&printtype=statement&method=print
TRAN_PROCESSCOMMISSN	Authorize Employee Commissions	/app/accounting/transactions/processcommissn.nl
TRAN_PROCESSICRTNAUTHS	Manage Intercompany Return Authorizations	/app/accounting/transactions/interco/rtnauthqueue.nl
TRAN_PROCESSICSALESORDERS	Manage Intercompany Sales Orders	/app/accounting/transactions/interco/salesordqueue.nl
TRAN_PROCESSORDER	Process Individual Order	/app/accounting/transactions/processorder.nl
TRAN_PROCESSPARTNERCOMMISSN	Authorize Partner Commissions	/app/accounting/transactions/processpartnercommissn.nl
TRAN_PURCHCONAPPRV	Approve Purchase Contracts	/app/accounting/transactions/transactionapproval.nl?type=PurchCon
TRAN_PURCHORDPROC	Bill Purchase Orders	/app/accounting/transactions/purchordermanager.nl?type=proc
TRAN_PURCHORDRECEIVE	Receive Orders	/app/accounting/transactions/purchordermanager.nl?type=receive
TRAN_PURCHREQAPPRV	Approve Requisitions	/app/accounting/transactions/transactionapproval.nl?type=PurchReq
TRAN_QUOTA	Establish Quotas	/app/crm/sales/quota.nl

Task ID	Page Label in NetSuite	URL
TRAN_REALLOCITEMS	Reallocate Items	/app/accounting/transactions/reallocitems.nl
TRAN_RECALCULATEREVENUEFORECASTPLANS	Recalculate Revenue Forecast Plans	/app/accounting/revrec/recalculaterevenueforecastplans.nl
TRAN_RECALCULATEREVENUEFORECASTPLANS_LOG	Recalculate Revenue Forecast Plans Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=RECALCREVFORECASTPLANS&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_RECONCILE	Reconcile Bank Statement	/app/accounting/transactions/reconcile.nl
TRAN_RECONCILE_CC	Reconcile Credit Card Statement	/app/accounting/transactions/reconcile.nl?page_type=cc
TRAN_REIMBURSEMENTS	Reimbursements	/app/accounting/transactions/reimbursements.nl
TRAN_REMINDERS	Setup Reminders	/app/center/setup/reminders.nl
TRAN_REPLENISHINVENTORY_LOG	Replenish Location By Transfer Order Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=REPLENISHINVENTORY&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_REVALUESTANDARDCOSTINVENTORY	Revalue Standard Cost Inventory	/app/accounting/inventory/revaluestandardcostinventory.nl
TRAN_REVALUESTANDARDCOSTINVENTORY_STATUS	Revalue Standard Cost Inventory Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=REVALUESTDCOSTINVENTORY&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_REVARRNGAPPRV	Approve Revenue Arrangements	/app/accounting/transactions/transactionapproval.nl?type=RevArrng
TRAN_REVIEWNEGATIVEINVENTORY	Review Negative Inventory	/app/accounting/transactions/inventory/reviewnegativeinventory.nl
TRAN_REVRECCREATEJE	Revenue Recognition Schedules	/app/accounting/transactions/revreccreatje.nl
TRAN_RTNAUTHAPPRV	Approve Return Authorizations	/app/accounting/transactions/returnauthmanager.nl?type=apprv
TRAN_RTNAUTHCREDIT	Refund Returns	/app/accounting/transactions/returnauthmanager.nl?type=credit
TRAN_RTNAUTHRECEIVE	Receive Returned Order	/app/accounting/transactions/returnauthmanager.nl?type=receive
TRAN_RTNAUTHREVERSEREVCOMMITMENT	Generate Revenue Commitment Reversals	/app/accounting/transactions/returnauthmanager.nl?type=reverserevcommitment
TRAN_SALESORDAPPRV	Approve Sales Orders	/app/accounting/transactions/salesordermanager.nl?type=apprv
TRAN_SALESORDCOMMITREVENUE	Generate Revenue Commitments	/app/accounting/transactions/salesordermanager.nl?type=commitrevenue
TRAN_SALESORDFULFILL	Fulfill Orders	/app/accounting/transactions/salesordermanager.nl?type=fulfill

Task ID	Page Label in NetSuite	URL
TRAN_SALESORDPROC	Invoice Sales Orders	/app/accounting/transactions/salesordermanager.nl?type=proc
TRAN_SALESORDREVENUECONTRACT	Generate Single Order Revenue Contracts	/app/accounting/transactions/salesordermanager.nl?type=createrevenuecontracts
TRAN_SAVEDASHBOARD	Publish Dashboard	/app/center/setup/savedashboard.nl
TRAN_SEARCH	Search	/app/common/search/search.nl
TRAN_SHORTCUTS	Add Shortcuts	/app/center/shortcuts.nl
TRAN_SNAPSHOTCOMPOSER	Custom Snapshot Report	/app/reporting/snapshotcomposer.nl
TRAN_SNAPSHOTS	Setup Snaphots	/app/center/setup/snapshots.nl
TRAN_STATSCHEDULE	Statistical Schedule Status	/app/accounting/transactions/statistical/statisticalschedulestatus.nl
TRAN_STPICKUP	Manage Store Pickup	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=StPickUp
TRAN_TAXPERIODS	Generate Tax Reporting Periods	/app/setup/period/generatetaxperiods.nl
TRAN_TIMEAPPROVAL	Approve Time	/app/accounting/transactions/timeapproval.nl
TRAN_TIMEBILL	Track Time	/app/accounting/transactions/timebill.nl
TRAN_TIMEBILL_WEEKLY	Weekly Time Sheet	/app/accounting/transactions/timebill.nl?weekly=T
TRAN_TIMECALC	Calculate Time	/core/pages/timecalc.nl
TRAN_TIMEENTRYAPPROVAL	Approve Time Entry	/app/accounting/transactions/timeentryapproval.nl
TRAN_TIMEPOST	Post Time	/app/accounting/transactions/timepost.nl
TRAN_TIMER	Timer	/core/pages/timer.nl
TRAN_TIMESHEETAPPROVAL	Approve Time	/app/accounting/transactions/timesheetapproval.nl
TRAN_TIMEVOID	Void Time	/app/accounting/transactions/timevoid.nl
TRAN_TRNFRORDAPPRV	Approve Transfer Orders	/app/accounting/transactions/transferordermanager.nl?type=apprv
TRAN_UPDATEREVENUERECOGNITIONPLANS	Edit Revenue Recognition Plans	/app/accounting/revrec/updaterevenuerecognitionplans.nl
TRAN_UPDATEREVENUERECOGNITIONPLANS_LOG	Update Revenue Recognition Plans Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=UPDATEREVRECPLANS&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_USERPREFS	Set Preferences	/app/center/userprefs.nl
TRAN_VENDAUTHAPPRV	Approve Vendor Returns	/app/accounting/transactions/vendauthmanager.nl?type=apprv
TRAN_VENDAUTHCREDIT	Credit Vendor Returns	/app/accounting/transactions/vendauthmanager.nl?type=credit

Task ID	Page Label in NetSuite	URL
TRAN_VENDAUTHRETURN	Ship Vendor Returns	/app/accounting/transactions/vendauthmanager.nl?type=return
TRAN_VENDBILLAPPRV	Approve Bills	/app/accounting/transactions/vendorbillmanager.nl?type=apprv
TRAN_VENDBILLPURCHORD	New Purchase Order	/app/accounting/transactions/vendbillpurchord.nl
TRAN_VENDPYMTS	Pay Bills	/app/accounting/transactions/vendpymts.nl
TRAN_VENDPYMT_LOG	Pay Bills Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=BULKPAYBILLS&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_WITHDRAWINVENTORY_LOG	Withdraw By Transfer Order Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=WITHDRAWINVENTORY&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_WORKORDBUILD	Build Work Orders	/app/accounting/transactions/salesordermanager.nl?type=build
TRAN_WORKORDBUILD_LOG	Build Work Orders Status	/app/accounting/bulkprocessing/bulkprocessingstatus.nl?bulkproctype=BUILDWORKORDERS&BulkProcSubmission_CREATEDDATE=TODAY
TRAN_WORKORDCLOSE	Close Work Orders	/app/accounting/transactions/workordermanager.nl?type=close
TRAN_WORKORDCOMPLETE	Enter Completions	/app/accounting/transactions/workordermanager.nl?type=complete
TRAN_WORKORDISSUE	Issue Components	/app/accounting/transactions/workordermanager.nl?type=issue
TRAN_WORKORDMARKBUILT	Mark Work Orders Built	/app/accounting/transactions/workordermanager.nl?type=markbuilt
TRAN_WORKORDMARKFIRMED	Mark Work Orders Firmed	/app/accounting/transactions/workordermanager.nl?type=markfirmed
TRAN_WORKORDMARKRELEASED	Mark Work Orders Released	/app/accounting/transactions/workordermanager.nl?type=markreleased

Button IDs

Note: The content in this help topic pertains to all versions of SuiteScript. Be aware that currently it may only include links or examples for SuiteScript 1.0.

The following table lists the internal IDs for standard NetSuite buttons that support SuiteScript. When using SuiteScript to rename or hide buttons, you will use the nlobjButton methods setLabel(label) and setVisible(visible), respectively.

On some records, it is possible that certain buttons will appear as actions in the More Actions menu. You can still use the nlobjButton methods to change the labels of these actions and to hide or show the actions in the menu. (For information on the More Action menu, see the help topic Configuring Buttons and Actions.)

Important: Customizing the Save, Edit, Cancel, Back, and Reset buttons is not supported in SuiteScript or in point-and-click customization.

Note: Also note that you cannot use SuiteScript to change the display of an inline button to an action in the More Actions menu. Similarly, you cannot use SuiteScript to display an action as an inline button. To change the display type of buttons and actions, you must use SuiteBuilder point-and-click customization. See the help topic Configuring Buttons and Actions for details.

Button UI Label	Button Internal ID
Add Items	addmatrix
Accept	accept
Accept Payment	acceptpayment
Apply	apply
Approve	approve
Approve Return	approvereturn
Authorize Return	return
Auto Fill	autofill
Bill	bill
Bill Remaining	billremaining
Cancel Order	cancelorder
Cancel Return	cancelreturn
Clear Splits	clearsplits
Close	closeremaining
Convert	convertlead
Convert to Inventory	convertinvt
Convert to Lot Numbered Inventory	convertlot
Convert to Serialized Inventory	convertserial
Create Build	createbuild
Create Matrix	creatematrix
Credit	credit
Decline	decline
Delete	delete
Email	email
Fax	fax
Fulfill	process
Generate Price List	generatepricelist
Generate Statement	generatestatement
GL Impact	glimpact

Button UI Label	Button Internal ID
Go To Register	gotoregister
Grab	grab
Make Copy	makecopy
Make Payment	payment
Make Standalone Copy	makestandalonecopy
Memorize	memorize
Merge	merge
New	new
New Event Field	neweventfield
Next Bill	nextbill
Next Week	next
Prev Week	prev
Print	print
Print Bill of Materials	printbom
Print Label	printlabel
Print Labels	printlabels
Print Picking Ticket	printpicktick
Print Summary	depositsummary
Quick Accept	quickaccept
Recalc	recalc
Receive	receive
Refund	refund
Reject	reject
Renew	renewal
Reset	resetter
Revenue Commitment Reversal	revcomrv
Save As	submitas
Save & Bill	submitbill
Save & Convert	submitconvert
Save & Copy	submitcopy
Save & Edit	submitedit
Save & Email	saveemail
Save & Fulfill	submitfulfill
Save & New	submitnew
Save & Next	submitnext

Button UI Label	Button Internal ID
Save & Print	saveprint
Save & Print BOM	saveprintbom
Save & Print Label	saveandprintlabel
Save & Refund	submitrefund
Save & Same	submitsame
Save Baseline	savebaseline
Search	search
Show Activity	showactivity
Submit Invoice	submitinvoice
Tentative	tentative
Unbuild	createunbuild
Update Matrix	updatematrix
Update VSOE	updatevsoe
View All Transactions	viewalltransactions
Void	void
W4 Worksheet	w4data