



SuiteTalk (Web Services) Records Guide

May 23, 2011
Version 2011.1.0

Copyright NetSuite, Inc. 2010 All rights reserved.

SuiteTalk Records Guide - Version 2011_1

May 23, 2011

This document is the property of NetSuite, Inc., and may not be reproduced in whole or in part without prior written approval of NetSuite, Inc.

Trademarks

The following marks are registered trademarks or service marks of NetSuite, Inc. in the United States and other countries.

- NETSUITE
- The "N" in NetSuite Logo
- SUITESCRIPT
- SUITEFLEX
- ONE SYSTEM. NO LIMITS.
- PREMIER PAYROLL SERVICE

OpenAir is a trademark of OpenAir Inc.

Other trademarks and service marks used or referenced in this document are the property of their respective owners and are hereby acknowledged.

Contents

Chapter 1

SuiteTalk Records Overview

In this Guide	14
How to Use This Guide	14
<i>Using the Help Center Version of the SuiteTalk Records Guide</i>	15
<i>Using the PDF Version of the SuiteTalk Records Guide</i>	15
<i>Downloading the SuiteTalk Schema Browser</i>	15
Web Services Supported Records	15
Using the SuiteTalk Schema Browser	22

Chapter 2

Entities

Customer	25
<i>Supported Operations</i>	25
<i>Field Definitions</i>	25
<i>Usage Notes</i>	26
Customer Status	29
<i>Supported Operations</i>	29
<i>Field Definitions</i>	29
Contact	29
<i>Supported Operations</i>	29
<i>Field Definitions</i>	30
<i>Usage Notes</i>	30
Employee	30
<i>Supported Operations</i>	30
<i>Field Definitions</i>	30
<i>Usage Notes</i>	31
Group (Entity Group)	31
<i>Supported Operations</i>	31
<i>Field Definitions</i>	32
<i>Usage Notes</i>	32
Project (Job)	32
<i>Supported Operations</i>	32
<i>Field Definitions</i>	32
<i>Usage Notes</i>	33
Project Type (Job Type)	34
<i>Supported Operations</i>	34
<i>Field Definitions</i>	34
Project Status (Job Status)	34
<i>Supported Operations</i>	35
<i>Field Definitions</i>	35
Partner	35
<i>Supported Operations</i>	35
<i>Field Definitions</i>	35
<i>Usage Notes</i>	35

Vendor	36
<i>Supported Operations</i>	36
<i>Field Definitions</i>	36
<i>Usage Notes</i>	36
Entity Search	36
<i>Field Definitions</i>	36

Chapter 3

Activities

Events (CalendarEvent)	38
<i>Supported Operations</i>	38
<i>Field Definitions</i>	38
<i>Usage Notes</i>	38
<i>Non-Recurring (One-Time) Events</i>	38
<i>Recurring Events</i>	39
Phone Call	42
<i>Supported Operations</i>	42
<i>Field Definitions</i>	42
Project Task	43
<i>Supported Operations</i>	43
<i>Field Definitions</i>	43
<i>Usage Notes</i>	44
Project Task Assignment Joined Search	44
<i>Field Definitions</i>	44
Tasks	45
<i>Supported Operations</i>	45
<i>Field Definitions</i>	45
<i>Usage Notes</i>	45

Chapter 4

Communications

Note	47
<i>Supported Operations</i>	47
<i>Field Definitions</i>	47
<i>Usage Notes</i>	47
Message	49
<i>Supported Operations</i>	49
<i>Field Definitions</i>	49
<i>Usage Notes</i>	50

Chapter 5

Transactions

Usage Notes for Transaction Record Types	55
Shared Field Definitions for Transactions	55
<i>BillAddress</i>	55
<i>ShipAddress</i>	56
<i>Setting Shipping and Billing Addresses on Transactions</i>	56
Transaction Search	58

Assembly Build	59
<i>Supported Operations</i>	59
<i>Field Definitions</i>	60
Assembly Unbuild	60
<i>Supported Operations</i>	60
<i>Field Definitions</i>	60
Budget	60
<i>Supported Operations</i>	60
<i>Field Definitions</i>	61
Opportunity	61
<i>Supported Operations</i>	61
<i>Field Definitions</i>	61
<i>Usage Notes</i>	61
Purchase Order	64
<i>Supported Operations</i>	64
<i>Field Definitions</i>	64
<i>Usage Notes</i>	64
Sales Order	64
<i>Supported Operations</i>	64
<i>Field Definitions</i>	64
<i>Usage Notes</i>	65
Invoice	70
<i>Supported Operations</i>	70
<i>Field Definitions</i>	71
<i>Usage Notes</i>	71
Journal Entry	72
<i>Supported Operations</i>	72
<i>Field Definitions</i>	72
<i>Usage Notes</i>	72
Intercompany Journal Entry	72
<i>Supported Operations</i>	72
<i>Field Definitions</i>	73
Customer Deposit	73
<i>Supported Operations</i>	73
<i>Field Definitions</i>	73
Deposit Application	73
<i>Supported Operations</i>	73
<i>Field Definitions</i>	73
Item Fulfillment	74
<i>Supported Operations</i>	74
<i>Field Definitions</i>	74
<i>Usage Notes</i>	74
Item Receipt	78
<i>Supported Operations</i>	78
<i>Field Definitions</i>	78
Transfer Order	78
<i>Supported Operations</i>	78
<i>Field Definitions</i>	78
<i>Usage Notes</i>	79
Intercompany Transfer Order	80

Supported Operations.....	80
Field Definitions.....	81
Usage Notes.....	81
Checks	82
Supported Operations.....	82
Field Definitions.....	82
Inventory Adjustment	82
Supported Operations.....	82
Field Definitions.....	83
Return Authorization	83
Supported Operations.....	83
Field Definitions.....	83
Usage Notes.....	83
Credit Memo	83
Supported Operations.....	84
Field Definitions.....	84
Cash Sale	84
Supported Operations.....	84
Field Definitions.....	84
Usage Notes.....	84
Estimate/Quote	86
Supported Operations.....	86
Field Definitions.....	86
Cash Refund	86
Supported Operations.....	86
Field Definitions.....	86
Customer Payment	87
Supported Operations.....	87
Field Definitions.....	87
Usage Notes.....	87
Customer Refund	89
Supported Operations.....	89
Field Definitions.....	89
Vendor Bill	89
Supported Operations.....	89
Field Definitions.....	89
Vendor Payment	90
Supported Operations.....	90
Field Definitions.....	90
Usage Notes.....	90
Time Bill (Track Time)	90
Supported Operations.....	90
Field Definitions.....	91
Expense Report	91
Supported Operations.....	91
Field Definitions.....	91
Multiple Shipping Routes in Web Services	92
What is the Multiple Shipping Routes feature?	92
How does MSR work in Web Services?.....	93
Which fields and elements are associated with MSR?.....	94

Does MSR work on custom forms?	95
Multiple Shipping Routes Code Samples for Web Services	95

Chapter 6

Items

Usage Notes for Item Record Types	114
Working with Matrix Items in Web Services.	114
<i>Item Types that Support Matrix Options</i>	<i>115</i>
<i>Creating a Parent Matrix Item</i>	<i>115</i>
<i>Creating a Child Matrix Item</i>	<i>115</i>
<i>Specifying Matrix Options</i>	<i>116</i>
<i>Matrix Dimensions</i>	<i>116</i>
<i>Updating Child and Parent Matrix Items</i>	<i>116</i>
<i>Deleting Child and Parent Matrix Items</i>	<i>116</i>
<i>Matrix Item Code Sample</i>	<i>116</i>
<i>Working with the Pricing Matrix</i>	<i>121</i>
Shared Field Definitions for Items	121
<i>Pricing Matrix List</i>	<i>122</i>
<i>Billing Rates Matrix List</i>	<i>123</i>
<i>Item Member List</i>	<i>123</i>
<i>Item Options List</i>	<i>124</i>
<i>Translation List</i>	<i>124</i>
<i>Item Vendor List</i>	<i>125</i>
<i>Site Category List</i>	<i>125</i>
Item Search	126
Assembly Item (BOM Item)	126
<i>Supported Operations</i>	<i>126</i>
<i>Field Definitions</i>	<i>126</i>
Download Item	128
<i>Supported Operations</i>	<i>128</i>
<i>Field Definitions</i>	<i>128</i>
<i>Usage Notes</i>	<i>129</i>
Inventory Item	131
<i>Supported Operations</i>	<i>131</i>
<i>Field Definitions</i>	<i>131</i>
<i>Usage Notes</i>	<i>131</i>
Kit/Package Item	132
<i>Supported Operations</i>	<i>132</i>
<i>Field Definitions</i>	<i>132</i>
Gift Certificate Item	132
<i>Supported Operations</i>	<i>132</i>
<i>Field Definitions</i>	<i>132</i>
<i>Usage Notes</i>	<i>133</i>
Lot Numbered Assembly Item	134
<i>Supported Operations</i>	<i>134</i>
<i>Field Definitions</i>	<i>134</i>
Lot Numbered Inventory Item	134
<i>Supported Operations</i>	<i>134</i>
<i>Field Definitions</i>	<i>134</i>

Usage Notes.....	135
Serialized Inventory Item	135
Supported Operations.....	135
Field Definitions.....	135
Usage Notes.....	135
Description Item	136
Supported Operations.....	136
Field Definitions.....	136
Discount Item.....	136
Supported Operations.....	136
Field Definitions.....	136
Markup Item.....	137
Supported Operations.....	137
Field Definitions.....	137
Payment Item	137
Supported Operations.....	137
Field Definitions.....	137
Subtotal Item	137
Supported Operations.....	137
Field Definitions.....	138
Noninventory Purchase Item.....	138
Supported Operations.....	138
Field Definitions.....	138
NonInventory Sales Item	138
Supported Operations.....	138
Field Definitions.....	138
Noninventory Resale Item	139
Supported Operations.....	139
Field Definitions.....	139
Other Charge Resale Item.....	139
Supported Operations.....	139
Field Definitions.....	139
Other Charge Purchase Item	139
Supported Operations.....	139
Field Definitions.....	140
Serialized Assembly Item	140
Supported Operations.....	140
Field Definitions.....	140
Service Resale Item.....	140
Supported Operations.....	140
Field Definitions.....	140
Service Purchase Item	141
Supported Operations.....	141
Field Definitions.....	141
Service Sale Item.....	141
Supported Operations.....	141
Field Definitions.....	141
Other Charge Sale Item.....	141
Supported Operations.....	141

Field Definitions	142
-------------------------	-----

Chapter 7

Support

Issue	143
Supported Operations	143
Field Definitions	143
Support Case	143
Supported Operations	144
Field Definitions	144
Usage Notes	144
Support Case Status	145
Supported Operations	145
Field Definitions	145
Support Case Type	145
Supported Operations	145
Field Definitions	145
Support Case Origin	146
Supported Operations	146
Field Definitions	146
Support Case Issue	146
Supported Operations	146
Field Definitions	146
Support Case Priority	146
Supported Operations	147
Field Definitions	147
Solution	147
Supported Operations	147
Field Definitions	147
Topic	147
Supported Operations	147
Field Definitions	148

Chapter 8

Website

Site Category	149
Supported Operations	149
Field Definitions	149
Usage Notes	149

Chapter 9

Lists

Account	151
Supported Operations	151
Field Definitions	152
Usage Notes	152
Accounting Period	152
Supported Operations	152

<i>Field Definitions</i>	152
<i>Usage Notes</i>	152
Bin	153
<i>Supported Operations</i>	153
<i>Field Definitions</i>	153
<i>Usage Notes</i>	153
Budget Category	154
<i>Supported Operations</i>	154
<i>Field Definitions</i>	154
<i>Usage Notes</i>	154
Classification (Class)	155
<i>Supported Operations</i>	155
<i>Field Definitions</i>	155
Currency	155
<i>Supported Operations</i>	155
<i>Field Definitions</i>	155
Department	155
<i>Supported Operations</i>	156
<i>Field Definitions</i>	156
Location	156
<i>Supported Operations</i>	156
<i>Field Definitions</i>	156
Expense Category	156
<i>Supported Operations</i>	156
<i>Field Definitions</i>	157
Gift Certificate	157
<i>Supported Operations</i>	157
<i>Field Definitions</i>	157
<i>Usage Notes</i>	157
Partner Category	159
<i>Supported Operations</i>	159
<i>Field Definitions</i>	159
Sales Tax Item	159
<i>Supported Operations</i>	159
<i>Field Definitions</i>	159
Subsidiary	159
<i>Supported Operations</i>	159
<i>Field Definitions</i>	160
Tax Group	160
<i>Supported Operations</i>	160
<i>Field Definitions</i>	160
Tax Type	160
<i>Supported Operations</i>	160
<i>Field Definitions</i>	160
Unit Type	161
<i>Supported Operations</i>	161
<i>Field Definitions</i>	161
Vendor Category	161
<i>Supported Operations</i>	161
<i>Field Definitions</i>	161

Chapter 10

Other Lists

Usage Notes for Other List Record Types	162
Contact Category	163
<i>Supported Operations</i>	163
<i>Field Definitions</i>	163
Contact Role	163
<i>Supported Operations</i>	163
<i>Field Definitions</i>	163
Customer Category	164
<i>Supported Operations</i>	164
<i>Field Definitions</i>	164
Lead Source	164
<i>Supported Operations</i>	164
<i>Field Definitions</i>	164
Note Type	165
<i>Supported Operations</i>	165
<i>Field Definitions</i>	165
Payment Method	165
<i>Supported Operations</i>	165
<i>Field Definitions</i>	165
Price Level	165
<i>Supported Operations</i>	165
<i>Field Definitions</i>	166
Pricing Group	166
<i>Supported Operations</i>	166
<i>Field Definitions</i>	166
Sales Role	166
<i>Supported Operations</i>	166
<i>Field Definitions</i>	167
Term	167
<i>Supported Operations</i>	167
<i>Field Definitions</i>	167
Win Loss Reason	167
<i>Supported Operations</i>	167
<i>Field Definitions</i>	167

Chapter 11

Customization

Custom Record	169
<i>Supported Operations</i>	169
<i>Field Definitions</i>	170
<i>Usage Notes</i>	170
Custom Record Type	171
<i>Supported Operations</i>	171
<i>Field Definitions</i>	171
Custom Record Custom Field	172
<i>Supported Operations</i>	172
<i>Field Definitions</i>	172

Usage Notes.....	173
Custom List	173
Supported Operations.....	173
Field Definitions.....	173
Usage Notes.....	174
CRM Custom Field	174
Supported Operations.....	174
Field Definitions.....	175
Entity Custom Field	175
Supported Operations.....	175
Field Definitions.....	175
Item Custom Field	175
Supported Operations.....	175
Field Definitions.....	176
Item Number Custom Field	176
Supported Operations.....	176
Field Definitions.....	176
Item Option Custom Field	176
Supported Operations.....	176
Field Definitions.....	176
Other Custom Field	177
Supported Operations.....	177
Field Definitions.....	177
Transaction Body Custom Field	177
Supported Operations.....	177
Field Definitions.....	177
Transaction Column Custom Field	177
Supported Operations.....	177
Field Definitions.....	178
Usage Notes.....	178

Chapter 12

Marketing

Campaign	180
Supported Operations.....	180
Field Definitions.....	180
Campaign Category	180
Supported Operations.....	181
Field Definitions.....	181
Campaign Audience	181
Supported Operations.....	181
Field Definitions.....	181
Campaign Family	181
Supported Operations.....	181
Field Definitions.....	181
Campaign Search Engine	182
Supported Operations.....	182
Field Definitions.....	182
Campaign Channel	182

<i>Supported Operations</i>	182
<i>Field Definitions</i>	182
Campaign Offer	182
<i>Supported Operations</i>	182
<i>Field Definitions</i>	183
Campaign Response	183
<i>Supported Operations</i>	183
<i>Field Definitions</i>	183
<i>Usage Notes</i>	183
Campaign Vertical	184
<i>Supported Operations</i>	184
<i>Field Definitions</i>	184
Campaign Subscription	184
<i>Supported Operations</i>	184
<i>Field Definitions</i>	184
Promotion Code	185
<i>Supported Operations</i>	185
<i>Field Definitions</i>	185
<i>Usage Notes</i>	185

Chapter 13

File Cabinet

File	188
<i>Supported Operations</i>	188
<i>Field Definitions</i>	188
<i>Usage Notes</i>	188
Folder	190
<i>Supported Operations</i>	190
<i>Field Definitions</i>	190

Chapter 14

Country and State Enumerations

Country Enumerations	191
Setting State Values in Web Services	198
<i>Creating Custom State Values</i>	201
<i>Setting the Allow Free-Form States in Addresses Preference</i>	201

Chapter 1 SuiteTalk Records Overview

Important: For information on SuiteTalk platform features, operations, types, warnings, errors, and faults, see the *SuiteTalk (Web Services) Platform Guide* in the NetSuite Help Center. Note that the material in the *SuiteTalk (Web Services) Records Guide* currently pertains to the **2011.1** endpoint.

In this Guide

This manual contains the following information:

- Information on using the PDF version of this guide in conjunction with the SuiteTalk Schema Browser
- All records supported in SuiteTalk
- A list of all supported operations for each record
- Usage notes or sample code to show how to work with specific records
- Within each record description, links to the SuiteTalk Schema Browser for all field definitions and field level help
- Guidelines for using the SuiteTalk Schema Browser. The SuiteTalk Schema Browser provides up-to-date reference information for each record supported in SuiteTalk.

How to Use This Guide

This guide must be used in conjunction with the [SuiteTalk Schema Browser](#). Each record listed in this guide includes a link to the SuiteTalk Schema Browser. Within the Schema Browser you will find all available fields, sublists, and search filter fields for that record. The Schema Browser also provides field level help for each field that appears on the record.

Within this guide, the description of each record includes these sections:

- **Supported Operations:** lists all operations that can be used with the record
- **Field Definitions:** provides a link to the Schema Browser. Also provides information on where to locate the record's XSD in the Schema Browser.

Note: For general information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

- **Usage Notes:** provides details and code samples that explain how to use the record. Be aware that not every record contains a Usage Notes section.

Important: For a list of all NetSuite records that are supported in SuiteTalk, see [Web Services Supported Records](#).

Using the Help Center Version of the SuiteTalk Records Guide

If you are logged in to your NetSuite account and are accessing the information in this guide through online Help, simply click the links to the SuiteTalk Schema Browser that are provided within each record description. The Schema Browser will automatically open.

Using the PDF Version of the SuiteTalk Records Guide

When using the PDF version of the *SuiteTalk (Web Services) Records Guide*, you must be logged in to your NetSuite account to open the links to the SuiteTalk Schema Browser.

If you are not logged in to your NetSuite account, it is recommended that you download the .zip version of the Schema Browser onto your local machine. After extracting the .zip, click the index.html file to open the Schema Browser. All Schema Browser links will work without having to be logged in to NetSuite.

See also [Downloading the SuiteTalk Schema Browser](#).

Downloading the SuiteTalk Schema Browser

A .zip file of the Schema Browser is located on the NetSuite Developer Portal. On the Developer Portal, to go Developer Resources > SuiteTalk > Documentation. After extracting all files, click *index.html* to open the Schema Browser.

Web Services Supported Records

The following records are currently supported in the SuiteTalk platform.

Record Type	Records
Entities	Contact Customer Customer Status Employee Entity Search Group (Entity Group) Partner Project (Job) Project Status (Job Status) Project Type (Job Type) Vendor
Activities	Events (CalendarEvent) Phone Call Project Task Tasks

Record Type	Records
Marketing	Campaign Campaign Audience Campaign Category Campaign Channel Campaign Family Campaign Offer Campaign Response Campaign Search Engine Campaign Subscription Campaign Vertical Promotion Code

Record Type	Records
Transactions	Assembly Build Assembly Unbuild Budget Cash Refund Cash Sale Checks Credit Memo Customer Deposit Customer Payment Customer Refund Deposit Application Estimate/Quote Expense Report Invoice Journal Entry Intercompany Journal Entry Intercompany Transfer Order Inventory Adjustment Transfer Order Item Fulfillment Item Receipt Opportunity Purchase Order Return Authorization Sales Order Time Bill (Track Time) Transaction Search Vendor Bill Vendor Payment
Support	Issue Support Case Support Case Issue Support Case Origin Support Case Priority Support Case Status Support Case Type Solution Topic

Record Type	Records
File Cabinet	File Folder

Record Type	Records
Items	Assembly Item (BOM Item) Description Item Discount Item Download Item Gift Certificate Item Inventory Item Item SearchKit/Package Item Lot Numbered Assembly Item Lot Numbered Inventory Item Markup Item Markup Item Noninventory Purchase Item Noninventory Resale Item NonInventory Sales Item Other Charge Purchase Item Other Charge Resale Item Other Charge Sale Item Payment Item Serialized Assembly Item Serialized Inventory Item Service Purchase Item Service Resale Item Service Sale Item Subtotal Item

Record Type	Records
Communications	Message Note
Website	Site Category
Lists	Account Accounting Period Bin Budget Category Classification (Class) Currency Department Location Expense Category Gift Certificate Partner Category Sales Tax Item Subsidiary Tax Group Tax Type Unit Type Vendor Category

Record Type	Records
Other Lists	Contact Category Contact Role Customer Category Lead Source Note Type Payment Method Price Level Pricing Group Sales Role Term Win Loss Reason
Customization	Custom Record Custom Record Custom Field Custom Record Type Custom List CRM Custom Field Entity Custom Field Item Custom Field Item Option Custom Field Other Custom Field Transaction Body Custom Field Transaction Column Custom Field

Using the SuiteTalk Schema Browser

The SuiteTalk Schema Browser provides a Web-based view of all records, fields, sublists, enumerations, operations, warnings, errors, and faults in SuiteTalk. It also includes field level help for most fields on a record. Note that when Web-services specific field level help is not available for a field, the field help defaults to the standard field help that displays within NetSuite.

The following table provides a summary of the schemas presented in the SuiteTalk Schema Browser.

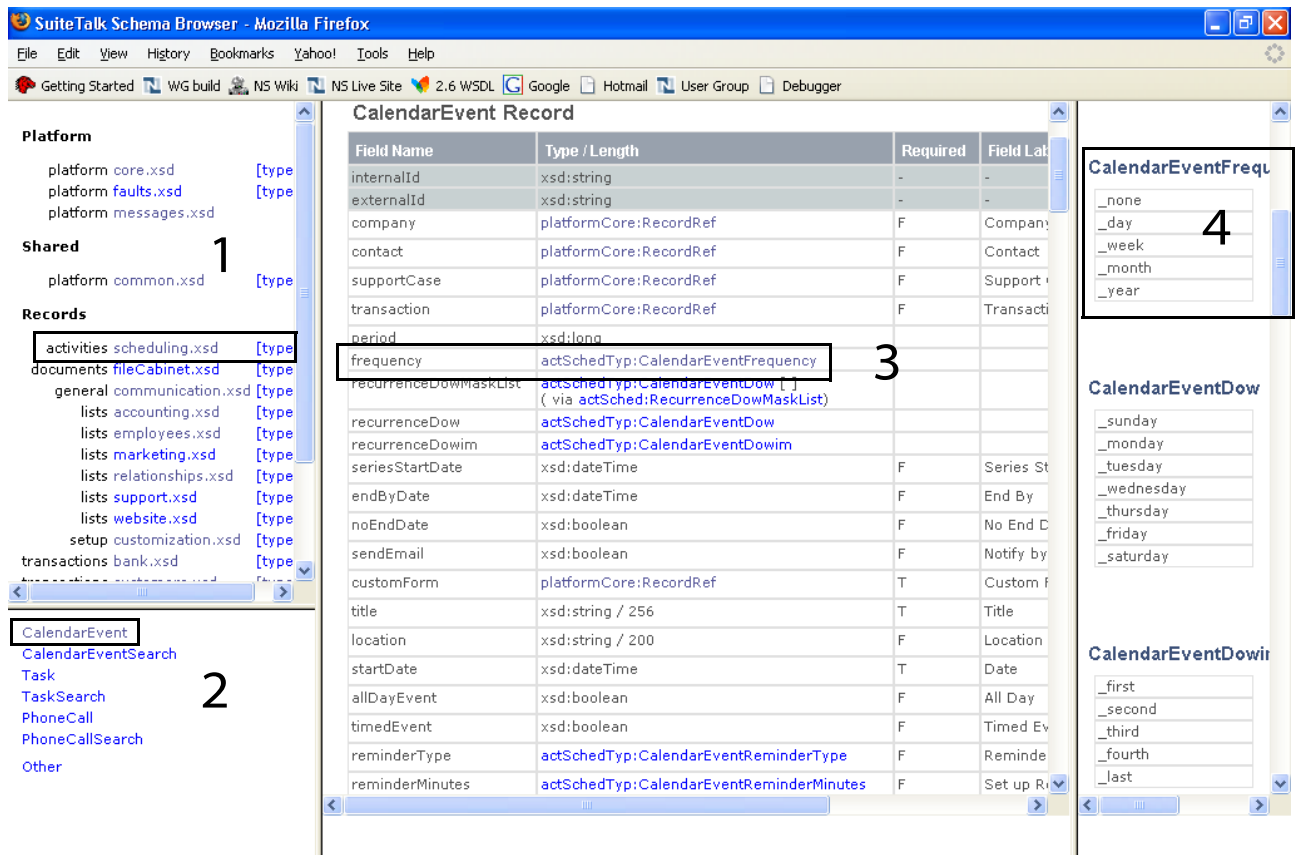
SuiteTalk Schema Browser	Notes
<p>Platform</p> <ul style="list-style-type: none"> core.xsd - defines all core-level objects shared across all other XSDs faults.xsd - defines all SOAP faults that can be thrown for each supported operation messages.xsd - defines all request and response objects used by the WSDL to interface with the SuiteTalk schema 	<p>In the upper-left frame (frame 1 in the figure below), click any of the Platform XSDs. In the lower-left frame (frame 2 in the figure below) you will see a link called "Other".</p> <p>Next, click the "Other" object link to display the supported objects for either the core, faults, or messages XSDs. The details will appear in the main frame (frame 3 in the figure below).</p>
<p>Shared</p> <ul style="list-style-type: none"> common.xsd - defines all search records 	<p>Click the common.xsd link to see a list of all NetSuite search records. The list of records will appear in the lower-left frame (frame 2 in the figure below).</p> <p>To see all search filter fields available on each search record, click the <Record>SearchBasic links in the lower-left frame. Filter fields will appear in the main frame (frame 3 in the figure below).</p>
<p>Records</p> <ul style="list-style-type: none"> Provides links to all exposed NetSuite business records 	<p>See the steps provided in To use the SuiteTalk Schema Browser.</p>

To use the SuiteTalk Schema Browser:

- In **frame 1**, click the XSD links to see the list of records defined in the each schema. The list of supported records appears in frame 2.
For example, the figure below shows a link to employees.xsd in frame 1. After clicking the link, the records defined in this XSD (Employee and EmployeeSearch) appear in frame 2.
- In **frame 2**, click each record to see the fields associated with the record. The list of fields appears in frame 3.
The figure shows a link to the Employee record in frame 2. After clicking the link, the fields for that record appear in frame 3.
- In **frame 3**, view the following information:
 - Field Name** - system name (internal ID) for the field

- **Type/Length** - the field data type, and if the field is a string, the maximum string length. Where applicable, click the data type link. For example, the payFrequency field on the Employee record is a multi-select list type (see figure). Click the data type link to see the list of enumeration values for this field in frame 4.
- **Required** - states whether the field is required in Web services; F equals false. For more information on understanding field requiredness, see [Required Fields](#) in the NetSuite Help Center.
- **Field Label** - field label as it appears in the UI
- **Notes** - provides field level help for each field. Note that when Web-services specific field level help is not available for a field, the field help defaults to the standard field help that displays within NetSuite.

4. In frame 4, view the enumeration types associated with a specific record or field.



Important: Schema fragments that represent business objects (records, lists on records, searches, search joins, and system enumerations) are displayed as tables for ease of browsing, and to provide extra information on the object where available. All other schema fragments are displayed as raw XSD and are listed as "Other Objects" in the Schema Browser table of contents (lower-left)

frame. The true schema for any objects can be found through the WSDL at https://webservices.netsuite.com/wsd/v2011_1_0/netsuite.wsd

Chapter 2 Entities

The following entity records are supported in SuiteTalk:

- Customer
- Customer Status
- Contact
- Employee
- Group (Entity Group)
- Project (Job)
- Project Type (Job Type)
- Project Status (Job Status)
- Partner
- Vendor
- Entity Search

Customer

Customer records represent people or companies that purchase goods and services from your business. Use the Customer record to manage these customers. Client applications can create, update, or delete customer records.

The Customer record is defined in the [listRel \(relationship\)](#) XSD.

Supported Operations

The following operations can be used to modify Customer records:

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchMoreWithId | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists relationships.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Usage notes are provide for the following topics:

- [Understanding Customer Stages](#)
- [Customer Status and Stage Internal IDs](#)
- [Creating Customers When Online Bill Pay is Enabled](#)
- [Returning a Contact List for a Customer](#)
- [Working with Customer Sublists](#)

Understanding Customer Stages

In the NetSuite UI, there are three possible stages that can be defined for a customer — Lead, Prospect or Customer. When entering new customers into NetSuite, you can create the record as a Lead, Prospect or Customer record, or you can create the customer as a Lead and allow NetSuite to automatically update the stage as certain criteria are met.

- **Lead:** allows you to track all the information you need to convert a lead into a customer. Leads have no estimates, opportunities or transactions associated with them. If an estimate or opportunity is created for a lead, the lead becomes a prospect. If you create a sales transaction for a lead, the lead becomes a customer.

Note: When adding leads through Web services, you can use the rules you have defined in the SFA (Sales Force Automation) feature to assign leads. Be aware that SFA rules are applied regardless of the setting for the “Use Conditional Defaults on Add” preference.

- **Prospect:** allows you to track all the information you need to convert a prospect into a customer. Prospects have no sales orders, invoices, cash sales or other sales transactions associated with them. They can have opportunities and estimates associated with them, however. If a sales transaction is created for a prospect, or an opportunity is closed for a prospect, the prospect becomes a customer.
- **Customer:** allows you to track all the information on your current customers.

This workflow is also maintained when working with Web services. The **stage** field indicates whether the customer is a Lead, Prospect or Customer as defined above. For example, if a customer record is defined as a Lead via Web services and then that customer record has an opportunity record associated with it via Web services, the Customer record stage field will automatically be updated to reflect the new Prospect stage. Once the opportunity record is updated to Closed-Won, the record stage field is automatically updated to Customer. To enter a new customer record at a specified stage, set the stage field to Lead, Prospect or Customer as desired.

Customer Status and Stage Internal IDs

The following table lists the internal IDs for all standard Customer Status and Stage values that can be used to populate the entityStatus or stage field.

Note: In addition to the following standard custom status values, your organization may have defined *custom* Customer Status values. If the Show Internal IDs preference is enabled, you can confirm the internal ID values in the associated list. See Setting the Internal ID Preference in the Web Services Platform Guide.

Customer Statuses		Customer Stages	
ID	Status	ID	Stage
17	Dead	LEAD	Lead
6	New	PROSPECT	Prospect
18	Qualified	CUSTOMER	Customer
14	Closed Lost		
9	Identified Decision Makers		
8	In Discussion		
11	In Negotiation		
7	Opportunity Identified		
10	Proposal		
12	Purchasing		
13	Closed Won		
16	Lost Customer		
15	Renewal		

Creating Customers When Online Bill Pay is Enabled

If your account has Online Bill Pay enabled, any customers created using Web services will NOT be enabled for Online Bill Pay. In the UI, if Online Bill Pay is enabled, by default a customer is enabled for this with the option to change the setting. However, since required Online Bill Pay specific fields are not currently exposed in Web services, when creating a new customer record with Web services the default value for Online Bill Pay is false regardless of whether Online Bill Pay is enabled.

Important: To create a relationship between a contact and a customer, you can set the **company** field to the Customer's Internal Id when updating or adding a contact. See [Contact](#).

Returning a Contact List for a Customer

In order to return a list of contacts associated to a given customer record, you must first get the customer record and then perform a search for the contacts associated to that customer.

Sample Java Code

```
// First get your customer
Customer c = (Customer)port.get(new RecordRef("17",RecordType.customer)).getRecord();

// Now do a specific search for the Contacts
ContactSearch cts = new ContactSearch();
```

```
// Search for an exact match between Customer.EntityId and the ContactSearch Field
Company.cts.setCompany(new SearchStringField(c.getEntityId()),SearchStringFieldOperator.is));

// Execute the search and you have your contactlist.
SearchResult scts = port.search(cts);
```

Working with Customer Sublists

The SuiteTalk Schema Browser includes all sublists associated with the Customer record. See the following information for usage notes regarding specific Customer sublists. Usage notes are not provided for every sublist type.

- [CustomerCreditCardsList](#)
- [CustomerAddressbookList](#)
- [CustomerSalesTeamList](#)
- [CustomerDownloadList](#)
- [SubscriptionsList](#)

For general information about working with sublists in SuiteTalk, see [Working with Sublists in Web Services](#) in the NetSuite Help Center.

CustomerCreditCardsList

The creditCardsList field is a list field that allows you to provide multiple credit card entries for a customer. The customer can then use these entries for payments. These fields are all mapped to the Financial tab in the UI.

CustomerAddressbookList

The addressBookList field is a list field that allows you to provide multiple addresses for a customer. These fields are all mapped to the Address tab in the UI.

When you work with the addressbookList on Customer records in the 2009.2 endpoint and beyond, be sure to use **internalId** as the key. Do not use **label**. For the 2009.1 and lower endpoints, the addressbookList is not a keyed sublist, but you can use label to identify records.

Important: Sales territory assignments are based on customers' default billing addresses. If you do not set defaultBilling to True for a customer address, it is not assigned to a territory automatically.

CustomerSalesTeamList

This list is only available when the Team Selling feature is enabled.

CustomerDownloadList

The Sell Downloadable Items feature must be enabled in an account in order to provide downloads for customers in the Customer Center.

SubscriptionsList

The subscriptionsList sublist on the Customer record can be updated. Use the replaceAll flag to update specific lines in this sublist, using **subscription** as the key. Be aware that Web services updates to this sublist follow the logic that is permitted in the NetSuite UI. Once an entity is unsubscribed from a category, only the entity itself can re-subscribe.

Customer Status

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 Customer Status record is defined in the [listRel \(relationship\)](#) XSD.

Supported Operations

The following operations can be used with Customer Status records:

add | addList | delete | deleteList | get | getDeleted | getList | getSavedSearch | search | searchMore | searchMoreWithId | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists relationships.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

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.

The Contact record is defined in the [listRel \(relationship\)](#) XSD.

Supported Operations

The following operations can be used to manipulate the Contact record.

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchMoreWithId | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists relationships.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Working with Contact Sublists

The SuiteTalk Schema Browser includes all sublists (lists) associated with the Contact record. See the following information for usage notes regarding specific Contact lists. Note that usage notes are not provided for every list type.

AddressbookList

The addressBookList field is a list field that allows you to provide multiple addresses for a contact. These fields are all mapped to the Address tab in the UI.

When you work with the addressbookList in the 2009.2 endpoint and beyond, be sure to use **internalId** as the key. Do not use **label**. For the 2009.1 and lower endpoints, the addressbookList is not a keyed sublist, but you can use label to identify records.

Employee

Employee records represent your employees. Use the employee record to store information for contact, login, payroll and human resources purposes.

The Employee record is defined in the [employees XSD](#).

Supported Operations

The following operations can be used to modify Employee records:

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchMoreWithId | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists employees.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Entering Expense Reports in Web Services

Web Services support for expense reports is different from user interface behavior. In the user interface, expense reports can only be entered for active employees. If an employee's release date is before the current date, expense reports are not supported. In Web services, expense reports can be entered for employees that have a release date before the current date.

Working with Employee List Fields

The SuiteTalk Schema Browser includes all lists (sublists) associated with the Employee record. However, see the following information for usage notes regarding specific Employee lists. Note that usage notes are not provided for every list type.

- [EmployeeAddressbookList](#)

Note that these fields on these lists are not applicable to CSV Imports.

EmployeeAddressbookList

The addressBookList field is a list field that allows you to provide multiple addresses for an employee. These fields are all mapped to the Address tab in the UI.

When you work with the addressbookList in the 2009.2 endpoint and beyond, be sure to use **internalId** as the key. Do not use **label**. For the 2009.1 and lower endpoints, the addressbookList is not a keyed sublist, but you can use label to identify records.

Group (Entity Group)

The Group record is defined in [listRel \(relationships\) XSD](#).

Specify the group type (contact, customer, employee, partner, or vendor) by setting the type element on an add operation. Note that the Group record is not customizable. Also note that in NetSuite, Groups can be dynamic or static. To create a dynamic group, on the Group record you must set the group type to dynamic. You can then reference a saved search when adding the Group record. The members will change based on the results of the saved search. To create a static group, first add the Group record to NetSuite. Then use the **attach** operation to associate members with it. (For information on the attach operation, see the NetSuite Help Center.)

Supported Operations

The following operations can be used with the Group record.

add | addList | attach / detach | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchMoreWithId | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists relationships.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

When searching for members of a **dynamic** group, you must use the GroupMemberSearchBasic search interface; you cannot use the group's corresponding entity search. The GroupMemberSearch interface retrieves members of a dynamic group by referencing the groupId.

GroupMemberSearchBasic

Field Name	Type
groupId	platformCore:SearchMultiSelectField

If the entity group is not dynamic, you can use the standard EntityGroupSearch interface to execute a search against the group.

Project (Job)

Important: Starting with the 2008.2 version of NetSuite, the Job record was renamed to **Project**. If you have existing code that references the Job record, this code will not break. The name change applies to external UI label only. In the [listRel \(relationships\) XSD](#), you should continue to reference the Job complex type in your code.

The Project record is defined in [listRel \(relationships\) XSD](#).

Supported Operations

The following operations can be used with the Project record:

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchMoreWithId | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists relationships.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

To use the Project record you must have the Projects feature enabled. Go to Setup > Company > Enable Features. Click the Company tab, and select **Projects**.

If you plan to do advanced project tracking, you must enable both **Projects** and **Advanced Projects**. If you do not see the Advanced Projects check box, your company must first purchase the Advanced Projects add-on from NetSuite.

Sample Code

The following sample shows how to add a new Project (Job) record.

SOAP Request

```
<soapenv:Body>
  <add xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
    <record xsi:type="ns6:Job" xmlns:ns6="urn:relationships_2_6.lists.webservices.netsuite.com">
      <ns6:entityId xsi:type="xsd:string">XYZ Inc</ns6:entityId>
      <ns6:companyName xsi:type="xsd:string">Scripting requirements</ns6:companyName>
      <ns6:entityStatus internalId="4" xsi:type="ns7:RecordRef"
        xmlns:ns7="urn:core_2_6.platform.webservices.netsuite.com" />
    </record>
  </add>
</soapenv:Body>
```

SOAP Response

```
<soapenv:Body>
  <addResponse xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
    <writeResponse>
      <ns2:status isSuccess="true" xmlns:ns2="urn:core_2_6.platform.webservices.netsuite.com" />
      <baseRef internalId="165" type="job" xsi:type="ns3:RecordRef"
        xmlns:ns3="urn:core_2_6.platform.webservices.netsuite.com" />
    </writeResponse>
  </addResponse>
</soapenv:Body>
```

Java

```
public void addJob() throws RemoteException, ExceededRecordCountFault,
    ExceededUsageLimitFault, InsufficientPermissionFault,
    InvalidSessionFault {
    // This operation requires a valid session
    this.login(true);

    Job job = new Job();

    job.setEntityId("XYZ Inc");
    job.setCompanyName("Scripting requirements");

    // Set entity status. The nsKey can be obtained from Setup > Accounting > Accounting List
    // Job Status.
    RecordRef status = new RecordRef();
    // 1 = Closed, 2 = In Progress, 3 = Not Awarded, 4 = Pending, 5 = Awarded
    status.setInternalId("4");

    job.setEntityStatus(status);
}
```

```

// Invoke add() operation
WriteResponse response = _port.add(job);

// Print the document id from the SOAP header
// _console.info(
// "\n\nThe add() operation with document id " + _port.documentInfo.nsID +
// " was processed " );

// Process the response
if (response.getStatus().isIsSuccess()) {
    _console.info("\n\nThe following job was added successfully:"
        + "\nkey="
        + ((RecordRef) response.getBaseRef()).getInternalId()
        + "\nentityId="
        + job.getEntityId()
        + "\ncompanyName="
        + job.getCompanyName()
        + "\nstatusKey="
        + job.getEntityStatus().getInternalId());
} else {
    _console.error("The job was not added:", true);
    _console.error(getStatusDetails(response.getStatus()));
}
}

```

Project Type (Job Type)

The Project Type record is defined in [listRel \(relationships\)](#) XSD.

Supported Operations

The following operations can be used with the Project Type record:

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchMoreWithId | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists relationships.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Project Status (Job Status)

The Project Status record is defined in [listRel \(relationships\)](#) XSD.

Supported Operations

The following operations can be used with the Project Status record:

add | addList | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchMoreWithId | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists relationships.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Partner

A partner is a company you have a business agreement with who isn't a customer or a vendor. To use Partner records the Partner Relationship Management feature must be enabled at Setup > Enable Features > CRM.

The Partner record is defined in the [listsRel \(relationships\) XSD](#).

Supported Operations

The following operations can be used with Partner records:

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchMoreWithId | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists relationships.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

When creating a Partner record, you can specify that partner as either an individual or a company by setting the isPerson field to TRUE or FALSE respectively. The available required and non-required fields vary depending on the setting of this field as detailed in the field definitions table.

Vendor

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.

The Vendor record is defined in the [listsRel \(relationships\) XSD](#).

Supported Operations

The following operations can be used to modify Vendor records:

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchMoreWithId | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists relationships.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Working with Vendor Lists

The SuiteTalk Schema Browser includes all lists (sublists) associated with the Vendor record. However, see the following information for usage notes regarding specific vendor lists. Note that usage notes are not provided for every list type.

VendorPricingScheduleList

This list can only be edited on an update. It defines the quantity pricing schedules to use for the vendor. The Quantity Pricing feature must be enabled to access this list.

Entity Search

The EntitySearchBasic record is not a stand-alone search record. It must be used in conjunction with a search join from another record. The EntitySearchBasic interface is used when you are unsure of the entity type you are searching against.

EntitySearchBasic is defined in the [platformCommon XSD](#).

Field Definitions

See [platform common.xsd](#) in the [SuiteTalk Schema Browser](#) for a list of all entity search filter fields.

For information on working with search filter fields and search joins, see the `search()` operation in the Operations section of the *SuiteTalk (Web Services) Platform Guide*.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Chapter 3 Activities

The following activity records are supported in SuiteTalk:

- [Events \(CalendarEvent\)](#)
- [Phone Call](#)
- [Project Task](#)
- [Tasks](#)

Events (CalendarEvent)

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

The Event record is defined in the [actSched \(scheduling\)](#) XSD.

Supported Operations

[add](#) | [addList](#) | [attach / detach](#) | [delete](#) | [deleteList](#) | [get](#) | [getDeleted](#) | [getList](#) | [getSavedSearch](#) | [getSelectValue](#) | [search](#) | [searchMore](#) | [searchNext](#) | [update](#) | [updateList](#)

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [activities scheduling.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Events can be scheduled as one-time events that are set to occur on a specific day at a specific time. Events can also be designated as recurring events, which recur over a period of several days, weeks, months, or years. See the following sections for more information:

- [Non-Recurring \(One-Time\) Events](#)
- [Recurring Events](#)

Non-Recurring (One-Time) Events

A non-recurring event takes place one time during a single day. Note that a non-recurring event cannot extend beyond midnight in a user's specified time zone. Both the *startDate* and

endDate fields must contain the same date. For more information on setting these two fields, see their descriptions in CalendarEvent record table in the [SuiteTalk Schema Browser](#).

Recurring Events

Recurring events recur on a daily, weekly, monthly, or yearly basis. You can schedule events to automatically recur in your calendar in both basic and advanced patterns.

Important: To add, delete, or update an event with advanced recurrence patterns, you must upgrade to at least the SuiteTalk 2.6 WSDL.

Basic Recurrence Patterns

A “basic” event recurrence pattern includes events such as daily departmental status meetings. These events begin and end on the same day, and they recur in a simple Monday through Friday pattern. An event with a basic recurrence pattern does not require that you set the *recurrenceDow*, *recurrenceDowim*, or *recurrenceDowMaskList* fields.

To set a basic recurrence pattern:

1. Set the *frequency* field.
2. Set the *period* field.
3. Set the *startDate* field.
4. Set the *endDate* field only if there is a known date by which you want to end the recurring event. Events that recur indefinitely do not require that you set the *endDate* field.

Advanced Recurrence Patterns

Events that are scheduled with advanced recurrence patterns may include some of the following types of patterns:

- A certain number of days apart, such as every 3 days
- The same day of the week with weeks off in between, such as Tuesday every 2 weeks
- The same date of every month
- The same date in a month with months off in between, such as every 3 months on the 20th
- The same day of the week every month, such as the first Friday of every month
- The same day of the week every few months, such as the first Tuesday every other month

To set an advanced recurrence pattern for an event, you must set the following fields, in addition to setting the *frequency*, *period*, and *startDate* fields:

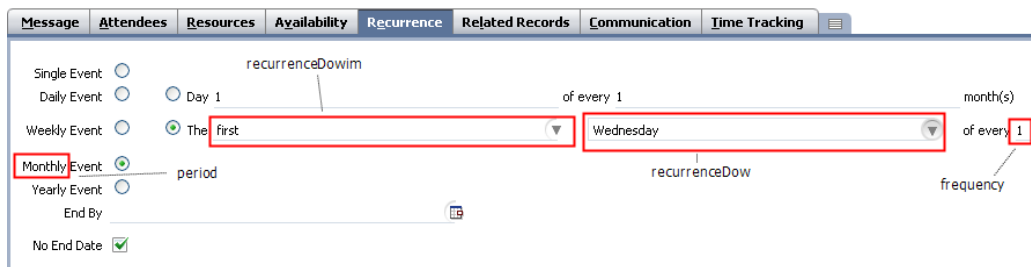
- *recurrenceDow*
- *recurrenceDowim*

- recurrenceDowMaskList

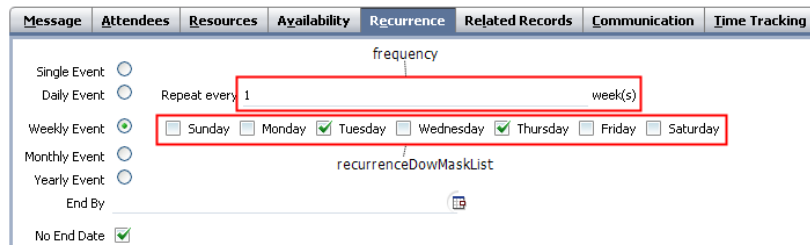
The recurrenceDow, recurrenceDowim, and recurrenceDowMaskList fields do not have corresponding field labels in the UI. These fields represent a collection of functions which are shown in the figure below.

Note: The recurrenceDow, recurrenceDowim, and recurrenceDowMaskList fields are hidden fields in SuiteScript. These fields are currently supported in server-side SuiteScript development; they are not supported in client development.

The following figure represents the scheduling of a monthly event that contains an advanced recurrence pattern. The figure shows the equivalent of *frequency* set to **_month**, *period* set to **1**, *recurrenceDowim* set to **_first**, and *recurrenceDow* set to **_wednesday**.



The following figure represents the scheduling of a weekly event that contains an advanced recurrence pattern. This figure shows the equivalent of *frequency* set to **_week** and *recurrenceDowMaskList* set to **_tuesday** and **_thursday**.



To set an advanced recurrence pattern that recurs monthly or yearly:

1. Set the *frequency* field to indicate how often the event will recur. The value specified for *frequency* dictates the values that will be set for the *recurrenceDow* and *recurrenceDowim* fields.
2. Next, set the *period* field.
3. Set the value for *recurrenceDow*.
4. Set the value for *recurrenceDowim*.

To set an advanced recurrence patterns that recurs weekly:

1. Set the *frequency* field to indicate how often the event will occur. The value specified for *frequency* dictates the value that will be set for the *recurrenceDowMaskList*.
2. Next, set the *period* field.
3. Finally, set *recurrenceDowMaskList*.

Updating a Series of Events that Include Advanced Recurrence Patterns

When updating a record in Web services, generally you submit only the values you intend to change. However, when updating an event record that includes advanced recurrence patterns, you must use *nullFieldList* to explicitly remove the values that were previously set.

For example, to update a **monthly** or a **yearly** event from a given day of the week in a month to a given day of the month, use *nullFieldList* to explicitly remove the values in *recurrenceDowim* and *recurrenceDow*.

Updating a Single Instance of a Recurring Event

A recurring event can be modified to change or delete an individual event in the series. In SuiteTalk this is handled using the **ExclusionDateList** type.

To delete individual events in a series:

Update the recurring event with a list of exclusions (in the form of dates) in the *ExclusionDateList* type. The *ExclusionDateList* type sets a list of timestamps of individual events that have been excluded from the series.

Important: Once an event has been excluded, it is separated from the rest of the events in the series, and it can NOT be associated with the event series.

To update an individual instance of a recurring event

1. Update the recurring event with a list of exclusions (in the form of dates) in the *exclusionDateList* field.
2. For each exclusion, add a new event that contains the updated data.
The original event and the new modified event have different internal ID values and the new updated event is no longer associated with the event series.

To retrieve excluded events from a series:

Submit a get operation for the desired event series and return the values listed in the *ExclusionDateList* type.

Working with Event Lists

The SuiteTalk Schema Browser includes all lists (sublists) associated with the Event record. See the following information for usage notes regarding specific Event lists. Note that usage notes are not provided for every list type.

- [ExclusionDateList](#)
- [AttendeeList](#)
- [ResourceList](#)

ExclusionDateList

This list is used to denote individual events in a series that have been modified or deleted. For detailed information, see [Updating a Single Instance of a Recurring Event](#).

AttendeeList

This list is used to identify each attendee for the event.

ResourceList

This list is used to schedule available resources for the event.

Phone Call

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 Phone Call record is defined in the [actSched \(scheduling\)](#) XSD.

Supported Operations

The following operations can be used to manipulate the Phone Call record.

[add](#) | [addList](#) | [attach / detach](#) | [delete](#) | [deleteList](#) | [get](#) | [getDeleted](#) | [getList](#) | [getSavedSearch](#) | [getSelectValue](#) | [search](#) | [searchMore](#) | [searchNext](#) | [update](#) | [updateList](#)

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [activities scheduling.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Project Task

When the Advanced Projects feature is enabled, you can create project task records to track activities against the project (job). For details on working with the project task record in NetSuite, see [Advanced Project Tasks](#) in the NetSuite Help Center.

The Project Task record is defined in the [actSched \(scheduling\)](#) XSD.

Important: To enable the Advanced Projects feature, see [Enabling Project Features](#). An error is thrown if you attempt to access a project task and Advanced Projects is not enabled. Also see [Permissions](#) for information on assigning the Project Task permission to a role.

For general information on the project task record, see these topics in the NetSuite Help Center:

- [Advanced Projects Overview](#)
- [Creating an Advanced Project Task Record](#)
- [Assigning Resources to Advanced Project Tasks](#)

Supported Operations

The following operations can be used with the Project Task record.

`add` | `addList` | `attach / detach` | `delete` | `deleteList` | `get` | `getDeleted` | `getList` | `getSavedSearch` | `getSelectValue` | `search` | `searchMore` | `searchMoreWithId` | `searchNext` | `update` | `updateList`

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [activities scheduling.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Project task records cannot be created or exist as standalone records. They must be attached to a project. For information on working with the project record in Web services, see [Project \(Job\)](#).

Permissions

Even if the Advanced Project feature is enabled, users must have Edit or Full permission for project tasks before they can create, update, or delete this record. For search, they must have at least View permission. NetSuite administrators can enable the **Project Task** permission for specific roles by going to Setup > Users/Roles > Manage Roles > click Permissions tab > click CRM tab > click the Project Task record > click Save.

Entering Time on Project Tasks

See these topics in Help to learn how time entered on a project task gets recorded in the project record.

Setting up Project Task Resources

To use the project task record in Web services, it is recommended that you read the Help for setting up resources. See Advanced Projects in Help.

Note that to set up project resources, the project resource flag on the employee record must be set. Then the employee must be listed as a resource on the project before you can create a project task and assign it to that resource.

Attaching Files to Project Tasks

You can attach files to project tasks using the attach operation. See [attach / detach](#) in the NetSuite Help Center for details on the attach operation.

Project Task Assignment Joined Search

The ProjectTaskAssignmentSearchBasic record is not a stand-alone search record. It must be used in conjunction with a search join from the projectTaskSearch record.

The ProjectTaskAssignmentSearchBasic is defined in the [platformCommon](#) XSD.

Field Definitions

See [platform common.xsd](#) in the [SuiteTalk Schema Browser](#) for a list of all project task assignment search filter fields.

For information on working with search filter fields and search joins, see the search() operation in the Operations section of the *SuiteTalk (Web Services) Platform Guide*.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Tasks

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 Task record is defined in the [actSched \(scheduling\)](#) XSD.

Supported Operations

The following operations can be used with Task records.

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [activities scheduling.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

In Web services tasks are ordered by creation date. In the UI, tasks can be ordered by specifying an Insert Before parameter that places the current task before an existing task assigned to the same user. This functionality is NOT available through Web services.

Identifying Unique TaskContact List Records

Entries in the TaskContact list are identified by unique combinations of the customer/contact pairing. Therefore, if you have multiple entries in the TaskContact list with the same customer, the contact is also needed to further identify the record. In the event that two or more entries have the same customer/contact pairing, these entries are treated as a single unit.

Restricting Time Entry on Tasks

Unless the Web services user is an administrator or the employee specified on the task, the user will not be able to set the reminderType and reminderMinutes fields on a Task record. If the user attempts to set these fields, the values specified will not be saved.

In the UI, you can set the **Check the Limit Time to Assignees** box to restrict time entry against this job. When you check this box, only assigned resources can enter time for this job's tasks.

Setting Start and End Times for Timed Tasks

In Web services the startTime and endTime field are not exposed. Therefore, to set start and end times for a timed task, users must first set the timedEvent field to TRUE. Then the time component of the DateTime value that is specified in startDate will be set for startTime, and the time component of the DateTime value that is specified in endDate will be set for endTime.

Note that the fields `reminderMinutes` and `reminderType` are also only settable if `timedEvent` is set to `TRUE`.

Code Sample

The following sample shows how to add a task to a job record.

Java

```
public void addTaskToJob() throws RemoteException {
    this.login(true);

    RecordRef custRef = new RecordRef();
    custRef.setInternalId("95");
    custRef.setType(RecordType.job);

    Task task = new Task();
    task.setCompany(custRef);
    task.setTitle("Race Car Tuning");
    task.setMessage("Includes baseline dyno, 5 hours tuning session & turbo upgrade.");

    WriteResponse response = _port.add(task);

    if (response.getStatus().isSuccess()) {
        _console.info("\n\nThe following task was added successfully: " + ((RecordRef)
response.getBaseRef()).getInternalId());
    } else {
        _console.error("The task was not added:", true);
        _console.error(getStatusDetails(response.getStatus()));
    }
}
```

Chapter 4 Communications

The following communications records are supported in SuiteTalk:

- Note
- Message

Note

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 Note record is defined in the [generalComm \(communication\) XSD](#).

Supported Operations

The following operations can be used to modify the Note record.

add | addList | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **general communications.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

A note must be associated with one and only one of these records. If values for more than one of these fields is submitted, an error is thrown.

Sample Code

The following SOAP and C# samples show how to add a Note.

SOAP Request

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <preferences xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
      <warningAsError>>false</warningAsError>
```

```

        <useConditionalDefaultsOnAdd>false</useConditionalDefaultsOnAdd>
    </preferences>
</soap:Header>
<soap:Body>
    <add xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
        <record xsi:type="q1:Note" xmlns:q1="urn:communication_2_6.general.webservices.netsuite.com">
            <q1:title>-- note title goes here --</q1:title>
            <q1:noteType internalId="7" type="noteType" />
            <q1:direction>_outgoing</q1:direction>
            <q1:noteDate>2008-04-09T00:00:00</q1:noteDate>
            <q1:note>-- memo goes here --</q1:note>
            <q1:activity internalId="39" type="calendarEvent" />
            <q1:author internalId="-5" type="employee" />
        </record>
    </add>
</soap:Body>
</soap:Envelope>

```

SOAP Response

```

<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <ns1:documentInfo xmlns:ns1="urn:messages_2_6.platform.webservices.netsuite.com">
      <ns1:nsId>WEBSERVICES_721410_040920088803062742117685935_252c2fc8bf8a0</ns1:nsId>
    </ns1:documentInfo>
  </soapenv:Header>
  <soapenv:Body>
    <addResponse xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
      <writeResponse>
        <ns2:status isSuccess="true" xmlns:ns2="urn:core_2_6.platform.webservices.netsuite.com" />
        <baseRef internalId="27" type="note" xsi:type="ns3:RecordRef"
          xmlns:ns3="urn:core_2_6.platform.webservices.netsuite.com" />
      </writeResponse>
    </addResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

C#

```

private void addNotes()
{
    this.login(true);

    Note note = new Note();

    RecordRef activityref = new RecordRef();
    activityref.internalId = "39";
    activityref.type = RecordType.calendarEvent;
    activityref.typeSpecified = true;
    note.activity = activityref;

    RecordRef authorref = new RecordRef();
    authorref.internalId = "-5";
    authorref.type = RecordType.employee;
    authorref.typeSpecified = true;

```

```

note.author = authorref;

note.direction = NoteDirection._outgoing;
note.directionSpecified = true;

RecordRef notetyperef = new RecordRef();
notetyperef.internalId = "7";
notetyperef.type = RecordType.noteType;
notetyperef.typeSpecified = true;
note.noteType = notetyperef;

DateTime searchDate = new DateTime();
searchDate = DateTime.Now;
searchDate = DateTime.Parse(searchDate.ToString("dd/MM/yyyy"));
note.noteDate = searchDate;
note.noteDateSpecified = true;

note.title = "-- note title goes here --";
note.note = "-- memo goes here --";

WriteResponse writeRes = _service.add(note);
if (writeRes.status.isSuccess)
{
    _out.writeLn("\n\nThe note " + note.internalId + " has been added successfully");
}
else
{
    _out.error(getStatusDetails(writeRes.status));
}
}

```

Message

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. Once 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.

The Message record is defined in the [generalComm \(communication\) XSD](#).

Note: Adding letters, PDFs, or faxes through Web services is currently not supported.

Supported Operations

add | addList | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [general communications.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

The bcc and cc fields represent the list of secondary email addresses associated with the message. These are string fields that accept a space delimited list of email addresses. For endpoints prior to 2009.2, invalid email addresses in the bcc and cc fields do not result in errors. For the 2009.2 and later endpoints, invalid email addresses in the bcc and cc fields return Invalid Field Value errors.

Note that Message recordref fields cannot be referred to by externalId.

Working with Attachments

You cannot update an attachment in the context of a message, but you can do so through an update operation on the File record. File records represent files that are stored in the NetSuite file cabinet. Use the File record to define email attachments sent or received via the Messages record. For additional usage notes on working with the File record, see [File](#).

The File record is defined in the [docFileCab \(fileCabinet\)](#) XSD.

Encoding

The schema has a data type of xsd:base64Binary which is mapped to a byte array in .NET and Axis (Java). On the client side we only need to pass a text or binary file as a byte array with no further base64 encoding. Therefore, do NOT Base64 Encode the file before sending it to Axis or .NET as these layers will do so themselves.

For example:

- Correct

```
-----
fis = new FileInputStream(inFile);
b = new byte[(int)inFile.length()];
fis.read(b);
myMediaItem.setContent(b);
-----
```

- Incorrect

```
-----
fis = new FileInputStream(inFile);
b = new byte[(int)inFile.length()];
fis.read(b);
myMediaItem.setContent(new String(Base64.encode(b)));
```

Storing Attachments

In order to maintain uniqueness for each file attachment, when attachments are stored in NetSuite two levels of sub-folders are automatically created for the Attachments to Send and Attachments Received respectively.

If the incoming field of the message record is set to **false**, the attachment is saved in the Attachments to Send folder. If the incoming filed is set to **true**, the attachment is saved in the Attachments Received folder.

The folder structure being generated is as follows:

- File Cabinet
- Attachments to Send or Attachments Received
- Entity Name
- Date & Time / Msg ID
- File name

For example:

A user is using auto email reply capture feature. They send a message to their contact John Smith from within the application. When the contact replies to the message they also attach a file to the message. When the message is created in NetSuite (through the auto reply capture feature) it is saved to a sub-folder that is created when the attachment is saved. The path to the sub-folder is as follows:

file cabinet > attachments received > Attachments Received > John Smith > 20050620_Message_525 > File.doc.

Note: If an attachment is being stored with an extension that does not match the actual file type (for example a .txt file as a .exe file), the file is listed in NetSuite as other text or other binary but this is just a label — there is no logical effect on the file.

Adding Messages

The following is a basic sample that shows how to add a message record.

SOAP Request

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Header>
<preferences xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
<warningAsError>false</warningAsError>
<useConditionalDefaultsOnAdd>false</useConditionalDefaultsOnAdd>
</preferences>
</soap:Header>
<soap:Body>
<add xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
<record xsi:type="q1:Message" xmlns:q1="urn:communication_2_6.general.webservices.netsuite.com">
<q1:author internalId="-5" type="employee" />
<q1:recipient internalId="-5" type="employee" />
<q1:messageDate>2008-04-09T00:00:00</q1:messageDate>
<q1:subject>-- subject goes here --</q1:subject>
<q1:message>This is a sample message</q1:message>
<q1:activity internalId="39" type="calendarEvent" />
</record>
```

```
</add>
</soap:Body>
</soap:Envelope>
```

SOAP Response

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<ns1:documentInfo xmlns:ns1="urn:messages_2_6.platform.webservices.netsuite.com">
<ns1:nsId>WEBSERVICES_721410_040920088711404551689975949_f1a83e87c1bf0</ns1:nsId>
</ns1:documentInfo>
</soapenv:Header>
<soapenv:Body>
<addResponse xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
<writeResponse>
<ns2:status isSuccess="true" xmlns:ns2="urn:core_2_6.platform.webservices.netsuite.com" />
<baseRef internalId="46" type="message" xsi:type="ns3:RecordRef"
xmlns:ns3="urn:core_2_6.platform.webservices.netsuite.com" />
</writeResponse>
</addResponse>
</soapenv:Body>
</soapenv:Envelope>
```

C#

```
private void addMessage()
{
this.login(true);

Message msg = new Message();

RecordRef activityref = new RecordRef();
activityref.internalId = "39";
activityref.type = RecordType.calendarEvent;
activityref.typeSpecified = true;
msg.activity = activityref;

RecordRef authorref = new RecordRef();
authorref.internalId = "-5";
authorref.type = RecordType.employee;
authorref.typeSpecified = true;
msg.author = authorref;

RecordRef recipientref = new RecordRef();
recipientref.internalId = "-5";
recipientref.type = RecordType.employee;
recipientref.typeSpecified = true;
msg.recipient = recipientref;

DateTime searchDate = new DateTime();
searchDate = DateTime.Now;
searchDate = DateTime.Parse(searchDate.ToString("dd/MM/yyyy"));
msg.messageDate = searchDate;
msg.messageDateSpecified = true;
```

```
msg.subject = "-- subject goes here --";  
msg.message = "This is a sample message";  
msg.incoming = true;  
  
WriteResponse writeRes = _service.add(msg);  
if (writeRes.status.isSuccess)  
{  
  _out.writeln("\nThe message " + msg.internalId + " has been added successfully");  
}  
else  
{  
  _out.error(getStatusDetails(writeRes.status));  
}  
}
```

Chapter 5 Transactions

Review the following for general information about SuiteTalk transactions:

- [Usage Notes for Transaction Record Types](#)
- [Shared Field Definitions for Transactions](#)
- [Multiple Shipping Routes in Web Services](#)

The following transaction record types are supported in SuiteTalk:

- Transaction Search
- Assembly Build
- Assembly Unbuild
- Budget
- Opportunity
- Purchase Order
- Sales Order
- Invoice
- Journal Entry
- Intercompany Journal Entry
- Customer Deposit
- Deposit Application
- Item Fulfillment
- Item Receipt
- Transfer Order
- Intercompany Transfer Order
- Checks
- Inventory Adjustment
- Return Authorization
- Credit Memo
- Cash Sale
- Estimate/Quote

- Cash Refund
- Customer Payment
- Customer Refund
- Vendor Bill
- Vendor Payment
- Time Bill (Track Time)
- Expense Report

Usage Notes for Transaction Record Types

Note the following when working with transactions:

- In order to maintain performance, 1000 lines per transaction is the maximum recommended limit for transactions submitted through Web services.
- There is a 200 record limit for Web services transactions.
- If you have the Multiple Shipping Routes (MSR) feature turned on in your NetSuite account, and you want to enabled MSR on a specific transaction, see [Multiple Shipping Routes in Web Services](#).
- The *rate* field on a sublist can be set without explicitly setting price level to **custom**. Even though NetSuite does not treat the price level as required in order to set the rate field, we recommend that users set the item price level to "-1" (custom) at the time they are setting the item rate.
- See the section [Shared Field Definitions for Transactions](#) for details on complex types that are referenced in several transaction types.

Shared Field Definitions for Transactions

The BillAddress and ShipAddress are two complex types that are referenced by multiple transaction types. See these sections for more information:

- [BillAddress](#)
- [ShipAddress](#)
- [Setting Shipping and Billing Addresses on Transactions](#)

BillAddress

BillAddress is a complex type that provides settable custom bill-to address fields on transactions. Be aware that in the UI the BillAddress fields are hidden.

Note: BillAddress fields are visible on the Item Fulfillment record only.

See [Setting Shipping and Billing Addresses on Transactions](#) for information on how to set billing addresses on transactions.

BillAddress is defined in the `platformCommon (common)` XSD.

ShipAddress

ShipAddress is a complex type that provides settable custom ship-to address fields on transactions. Note that in the UI the ShipAddress fields are hidden.

Note: ShipAddress fields are visible on the Item Fulfillment record only.

See [Setting Shipping and Billing Addresses on Transactions](#) for information on how to set shipping addresses on transactions.

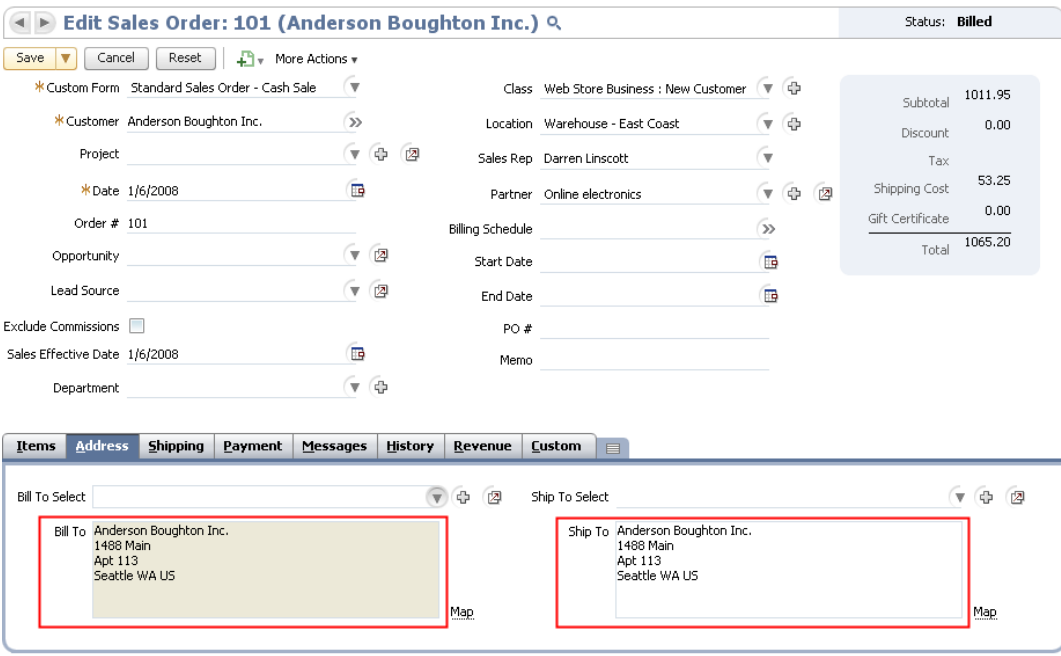
ShipAddress is defined in the `platformCommon (common)` XSD.

Setting Shipping and Billing Addresses on Transactions

On add/update operations, you can set ship-to and bill-to addresses on transactions using the following three approaches. Your use case will ultimately determine the approach you take, however, NetSuite recommends setting address references in the following order:

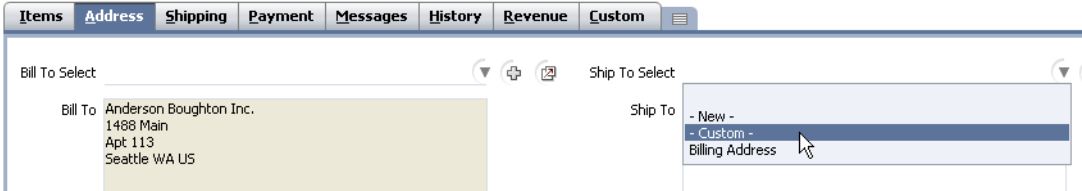
1. Set the address fields on the Customer AddressbookList sublist. These fields will become the default addresses referenced in transactions. (See [CustomerAddressbookList](#) for a list of available fields.)

The following figure shows how values set in the Customer AddressbookList sublist are used by default for shipping and billing addresses on the customer's transactions.

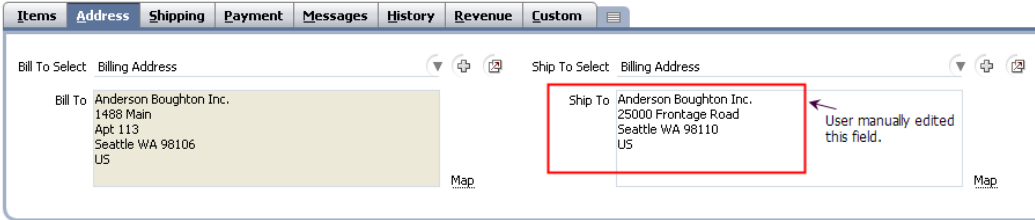


- Set custom ship-to/bill-to addresses on a transaction using the ShipAddress/BillAddress object fields. Be aware that the ShipAddress/BillAddress fields are hidden in the UI. Note that the ShipAddress/BillAddress objects are references by **transactionShipAddress/transactionBillAddress** elements, respectively. These elements appears on several transaction types. (See [ShipAddress](#) and [BillAddress](#) for available fields.)

In the UI, the second approach is equivalent to selecting Custom from the **Ship To Select** drop-down list on a transaction. When the Custom Address popup opens, you can set address values in the address fields provided.



- Set ship-to or bill-to addresses using the **shipAddress** and **billAddress** fields that appear on the specific transaction. The UI equivalent to approach number three is seen in the following screenshot.



Note that override behavior for shipping and billing addresses varies depending upon the endpoint:

- For endpoints prior to 2010.1:
 - If you set address values in the Customer AddressbookList sublist (the first approach), these values will **always** be used by the system, regardless of whether

you **also** set the ship-to and bill-to addresses using the next two approaches. For the system to read the ship-to/bill-to addresses set using either of the next two approaches, the fields on the AddressbookList sublist must be blank. If these fields are already populated, you must do an update on this sublist and submit an empty sublist. See [Non-Keyed Sublists](#) in the NetSuite Help Center.

- If you set address information using both the second and third approaches simultaneously, then the values set in the third approach will override the values set in the second approach. Note, however, that the values set in either the second approach or third approach will **ONLY** be read if the values on the Customer AddressbookList were never set.
- For 2010.1 and later endpoints:
 - Even if you set address values in the Customer AddressbookList sublist (the first approach), values you set by setting custom ship-to/bill-to addresses on a transaction using the ShipAddress/BillAddress object fields (the second approach) override the values set using the first approach.
 - Values you set by setting the ship-to or bill-to addresses using the **shipAddress** and **billAddress** fields that appear on the specific transaction (the third approach) override any values set using the first approach or the second approach.

Transaction Search

Nearly all transaction record types use the TransactionSearch record for search. The *basic* element in this record references the TransactionSearchBasic record, which lists all available search filter fields available in a transaction search.

The TransactionSearch record also lists all search joins available in a transaction search. In the [SuiteTalk Schema Browser](#), see **transactions sales.xsd** for the TransactionSearch record.

Be aware that the actual search filter fields available will vary depending on the transaction type you are searching against. Not *all* of the search filter fields defined in TransactionSearchBasic will exist on every single transaction type. For example, the Check transaction type may include search filter fields not available on the Journal Entry type. However, search filters values for both record types will be defined in TransactionSearchBasic.

Note: The Opportunity and the Time Bill records are the only transaction types that have their own search interfaces. In the [SuiteTalk Schema Browser](#), see the OpportunitySearch record by clicking **transactions sales.xsd**. For the TimeBillSearch record, click **transactions employees.xsd**.

Important: By default only a record's body fields are returned on a search. Therefore, you must set the **bodyFieldsOnly** element of the SearchPreferences type to **false** if you want to also return the information specified on a record's sublist. For information on setting search preferences, see the section "Setting Search Preferences" in the *SuiteTalk (Web Services) Platform Guide*. For general information on searching in Web services, see *search()* in the "Operations" section of the *SuiteTalk (Web Services) Platform Guide*.

Sample Code

The following sample returns three records, including one that was modified within two minutes the search was made.

SOAP Request

```
<soapenv:Body>
  <search xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
    <searchRecord xsi:type="ns9:TransactionSearchBasic"
      xmlns:ns9="urn:common_2_5.platform.webservices.netsuite.com">
      <ns9:lastModifiedDate operator="onOrAfter" xsi:type="ns10:SearchDateField"
        xmlns:ns10="urn:core_2_5.platform.webservices.netsuite.com">
        <ns10:predefinedSearchValue xsi:type="ns11:SearchDate"
          xmlns:ns11="urn:types.core_2_5.platform.webservices.netsuite.com">today</ns10:predefinedSearchValue>
        </ns9:lastModifiedDate>
        <ns9:type operator="anyOf" xsi:type="ns12:SearchEnumMultiSelectField"
          xmlns:ns12="urn:core_2_5.platform.webservices.netsuite.com">
        <ns12:searchValue xsi:type="xsd:string">_salesOrder</ns12:searchValue>
        </ns9:type>
      </searchRecord>
    </search>
  </soapenv:Body>
```

Java

```
TransactionSearchBasic basic = new TransactionSearchBasic();

SearchEnumMultiSelectField soType = new SearchEnumMultiSelectField();
soType.setSearchValue(new String[1]);
soType.setSearchValue(0, TransactionType.__salesOrder);
soType.setOperator(SearchEnumMultiSelectFieldOperator.anyOf);
basic.setType(soType);

SearchDateField todayLastMod = new SearchDateField();
todayLastMod.setOperator(SearchDateFieldOperator.onOrAfter);
todayLastMod.setPredefinedSearchValue(SearchDateField.today);

basic.setLastModifiedDate(todayLastMod);
```

Assembly Build

The Assembly Build record is defined in the [tranInvt \(inventory\) XSD](#).

Supported Operations

The following operations can be used with the Assembly Build record.

add | addList | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | initialize / initializeList | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions inventory.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Assembly Unbuild

The Assembly Unbuild record is defined in the [tranInvt \(inventory\)](#) XSD.

Supported Operations

The following operations can be used with the Assembly Unbuild record.

add | addList | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | initialize / initializeList | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions inventory.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Budget

A budget records the expected values of income and expenses for your business. You can create budgets for specific customers, items, departments, classes, locations, or any combination of these criteria. You can also create budgets for multiple subsidiaries in NetSuite OneWorld. For general information about budgets, see “Setting Up a Budget” in the NetSuite Help Center.

In Web services, the Budget record is defined in the [tranFin \(financial\)](#) XSD. Note that before updating a row on this record, you must perform a **get** on the entire record.

Supported Operations

The following operations can be used with the Budget record.

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions financial.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Opportunity

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.

The Opportunity record is defined in the [tranSales \(sales\) XSD](#).

Supported Operations

The following operations can be used with the Opportunity record.

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions sales.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

For details about working with opportunities in NetSuite, see the help topic [Opportunity Records](#).

Associating Transactions with Opportunities

You can associate estimates, cash sales, sales orders, and invoices with opportunities. Once a transaction other than an estimate is associated with an opportunity, the opportunity's status is automatically set to **Closed Won**. Once an opportunity's status is set to **Closed Won**, it is no longer available to be selected on other cash sale, sales order, or invoice records.

Projected Total, Range High and Range Low Fields

When working with Opportunity records in the UI, the range low, range high and projected total values must comply with the following rules:

- The rangeLow value must be lower than or equal to the projectedTotal value. This value represents the Worst Case projected total.
- The rangeHigh value must be greater than or equal to the projectedTotal value. This amount represents the Upside projected total.

In the UI, these values are calculated but can be overridden by the user. The calculated values are determined by relationships between the summation of amounts of entries in the itemList, projectedTotal values, and range low / range high values. When any of the values are overridden by the user, client side validation ensures that the fields are set correctly according to the above rules and automatically recalculates where appropriate.

In order to maintain proper rangeHigh, rangeLow and projectedTotal relationships when using Web services, the behavior is as follows:

If Advanced Forecasting is Off

On Add

- projectedTotal is required unless there's an item in the item list
- If projectedTotal is provided and there are items, then projectedTotal is calculated (with the sum of the item amounts)
- If projectedTotal is provided then it is set unless it is equal to the sum of all items, where in that case it is calculated

On update

- If projectedTotal is not provided and there are no items in the update, then projectedTotal is not changed
- If projectedTotal is not provided and there are items in the update request, then projectedTotal is set to the sum of all items
- If projectedTotal is not provided and there are items in the original but no items in the update request, then projectedTotal is not changed except where projectedTotal is equal to the sum of all item amounts. In that case it is calculated.
- If projectedTotal is provided and there are items in the original, then projectedTotal is set
- If projectedTotal is provided are there are items in the update request, then projectedTotal is set except where projectedTotal is equal to the sum of all items in the update request. In the case it is calculated.

If Advanced Forecasting is On

On Add

- If no item list is sent in the request and projectedTotal is not provided an error will be thrown.
- If no item list is sent and one or both of rangeLow or rangeHigh are missing then both are set to projectedTotal

- If a value is NOT provided for projectedTotal, rangeLow or rangeHigh fields, then each field is calculated.
- If a value is submitted for any one field then that field is NOT calculated except when a projectedTotal value is equal to the sum of the amounts for entries in the itemList at any point during item summation. In this case the projectedTotal value is calculated (set to the sum of all item amounts) even though a value was submitted.

On Update

- If any of the three fields has the same value as previously set, whether calculated or not, all three fields are calculated.
- If all three fields have different values they are not calculated except when a projectedTotal value is equal to the sum of the amounts for entries in the itemList at any point during item summation. In this case the projectedTotal value is calculated (set to the sum of all item amounts) even though a value was submitted.
- If any of the three fields is empty, then all three fields are populated with calculated values except when there is no change to the itemList.

Working with Opportunity Sublists

The SuiteTalk Schema Browser includes all sublists associated with the Opportunity record. See the following information for usage notes regarding specific Opportunity sublists. Usage notes are not provided for every sublist type.

- [OpportunityItemList](#)
- [OpportunitySalesTeam](#)

OpportunityItemList

In an opportunity, a list of items can be added, modified or deleted in the items sublist. However, each item listed in the item sublist is not a keyed entry — the list is simply an array of items. Therefore, in order to modify the contents of the items sublist for a given opportunity record, follow these guidelines:

- To update the list of items, resubmit the **entire** list (array) of items associated with the opportunity record.
- To delete a subset of items on the list, submit a partial list of what was retrieved with the get operation.
- To delete the entire list, submit an empty list.

Important: If you are using a customForm that has location, department and class customized at the item level, you can NOT set these fields at the body level.

OpportunitySalesTeam

This list is only available when the Team Selling feature is enabled.

Purchase Order

The Purchase Order record is defined in the [tranPurch \(purchases\)](#) XSD.

Supported Operations

The following operations can be used with Purchase Order records:

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions purchases.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Working with Purchase Order Sublists

The SuiteTalk Schema Browser includes all sublists associated with the Purchase Order record. See the following information for usage notes regarding specific Purchase Order sublists. Usage notes are not provided for every list type.

PurchaseOrderApprovalsList

This is a read-only list available on an update and maps to the History/Approvals tab in the UI.

Sales Order

Sales orders are record items and services you promise to customers. The Sales Order record is defined in the [tranSales \(sales\)](#) XSD.

Supported Operations

The following operations can be used to modify Sales Order records:

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | initialize / initializeList | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions sales.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Adding a Sales Order with an Item Sublist

.NET

```
public virtual void addSalesOrder()
{
    // This operation requires a valid session
    this.login(true);

    SalesOrder so = new SalesOrder();
    // Set customer entity

    _out.WriteLine("\nPlease enter the following customer information. "
        + "Note that some fields have already been populated. ");
    _out.write(" Customer entity name: ");

    CustomerSearch custSearch = new CustomerSearch();
    SearchStringField customerEntityID = new SearchStringField();
    customerEntityID.@operator = SearchStringFieldOperator.@is;
    customerEntityID.operatorSpecified = true;
    customerEntityID.searchValue = _out.ReadLine();

    CustomerSearchBasic custBasic = new CustomerSearchBasic();
    custBasic.entityId = customerEntityID;

    //custSearch.setEntityId(customerEntityID);

    custSearch.basic = custBasic;

    // Search for the customer entity
    SearchResult res = _service.search(custSearch);

    if (res.status.isSuccess)
    {
        if (res.recordList != null && res.recordList.Length == 1)
        {
            RecordRef customer = new RecordRef();
            customer.type = RecordType.customer;
            customer.typeSpecified = true;
            System.String entID = ((Customer) (res.recordList[0])).entityId;
            customer.name = entID;
            customer.internalId = ((Customer) (res.recordList[0])).internalId;
            so.entity = customer;

            // set the transaction date and status
            so.tranDate = new System.DateTime();
            so.orderStatus = SalesOrderOrderStatus._pendingApproval;

            // Enter the nsKey for inventory items to be added to the SO
            _out.WriteLine("\nPlease enter the nsKey values for
```

```
INVENTORY ITEMS separated by
commas (do not enter discount or subtotal items).");
System.String itemKeys = _out.readLn();
System.String[] itemKeysArray = itemKeys.Split(new Char[] {','});

ReadResponse[] readRes = getInventoryItemList(itemKeysArray);

// Determine the number of valid inventory item nsKeys entered
ArrayList vec = new ArrayList();
for (int a = 0; a < readRes.Length; a++)
{
    if (readRes[a].record != null)
    {
        vec.Add(readRes[a].record);
    }
}
SalesOrderItem[] salesOrderItemArray = new SalesOrderItem[vec.Count];

// Create the correct sales order items and populate the
// quantity
for (int i = 0; i < vec.Count; i++)
{
    if (vec[i] is InventoryItem)
    {
        RecordRef item = new RecordRef();
        item.type = RecordType.inventoryItem;
        item.typeSpecified = true;
        item.internalId = ((InventoryItem) (vec[i])).internalId;
        salesOrderItemArray[i] = new SalesOrderItem();
        salesOrderItemArray[i].item = item;

        _out.writeLn("\nPlease enter quantity for " + ((InventoryItem)
        (vec[i])).itemId);
        System.Double quantity = System.Double.Parse(_out.readLn());
        salesOrderItemArray[i].quantity = quantity;
        salesOrderItemArray[i].quantitySpecified = true;
    }
}

//SalesOrderItemList salesOrderItemList = new
SalesOrderItemList(salesOrderItemArray, true);
//TODO: No constructor present for two argument.
SalesOrderItemList salesOrderItemList = new SalesOrderItemList();
salesOrderItemList.item = salesOrderItemArray;

so.itemList = salesOrderItemList;

WriteResponse writeRes = _service.add(so);
if (writeRes.status.isSuccess)
{
    _out.writeLn("\nSales order created successfully");
}
else
{
    _out.error(getStatusDetails(writeRes.status));
}
```

```

    }
  }
  else
  {
    _out.writeln("\nSales order is not created because 0 or more than 1
      customer records found for the entityID given");
  }
}
else
{
  _out.error(getStatusDetails(res.status));
}
}

```

Java

```

public void addSalesOrder() throws RemoteException,
    ExceededUsageLimitFault, UnexpectedErrorFault, InvalidSessionFault,
    ExceededRecordCountFault {
  // This operation requires a valid session
  this.login(true);

  SalesOrder so = new SalesOrder();
  // Set customer entity

  _console.writeln("\nPlease enter the following customer information. "
    + "Note that some fields have already been populated. ");
  _console.write(" Customer entity name: ");

  CustomerSearch custSearch = new CustomerSearch();
  SearchStringField customerEntityID = new SearchStringField();
  customerEntityID.setOperator(SearchStringFieldOperator.is);
  customerEntityID.setSearchValue(_console.readLine());

  CustomerSearchBasic custBasic = new CustomerSearchBasic();
  custBasic.setEntityId(customerEntityID);

  //custSearch.setEntityId(customerEntityID);

  custSearch.setBasic(custBasic);

  // Search for the customer entity
  SearchResult res = _port.search(custSearch);

  if (res.getStatus().isIsSuccess()) {
    if (res.getRecordList().getRecord()!=null && res.getRecordList().getRecord().length == 1) {
      RecordRef customer = new RecordRef();
      customer.setType(RecordType.customer);
      String entID = ((Customer) (res.getRecordList().getRecord(0)))
        .getEntityId();
      customer.setName(entID);
      customer.setInternalId(((Customer) (res.getRecordList()
        .getRecord(0))).getInternalId());
      so.setEntity(customer);

      // set the transaction date and status
      so.setTranDate(Calendar.getInstance());
    }
  }
}

```

```

so.setOrderStatus(SalesOrderOrderStatus._pendingApproval);

// Enter the nsKey for inventory items to be added to the SO
_console
    .write("\nPlease enter the Internal Ids values for INVENTORY ITEMS separated
by commas (do not enter discount or subtotal items): ");
String itemKeys = _console.readLine();
String[] itemKeysArray = itemKeys.split(",");

ReadResponse[] readRes = getInventoryItemList(itemKeysArray);

// Determine the number of valid inventory item nsKeys entered
Vector vec = new Vector();
for (int a = 0; a < readRes.length; a++) {
    if (readRes[a].getRecord() != null) {
        vec.add(readRes[a].getRecord());
    }
}
SalesOrderItem[] salesOrderItemArray = new SalesOrderItem[vec
    .size()];

// Create the correct sales order items and populate the
// quantity
for (int i = 0; i < vec.size(); i++) {
    if (vec.get(i) instanceof InventoryItem) {
        RecordRef item = new RecordRef();
        item.setType(RecordType.inventoryItem);
        item.setInternalId(((InventoryItem) (vec.get(i)))
            .getInternalId());
        salesOrderItemArray[i] = new SalesOrderItem();
        salesOrderItemArray[i].setItem(item);

        _console.write("\nPlease enter quantity for "
            + ((InventoryItem) (vec.get(i))).getItemId() + ": ");
        Double quantity = Double.valueOf(_console.readLine());
        salesOrderItemArray[i].setQuantity(quantity);
    }
}
SalesOrderItemList salesOrderItemList = new SalesOrderItemList(
    salesOrderItemArray, true);
so.setItemList(salesOrderItemList);

WriteResponse writeRes = _port.add(so);
if (writeRes.getStatus().isSuccess()) {
    _console.writeln("\nSales order created successfully");
} else {
    _console.error(getStatusDetails(writeRes.getStatus()));
}
} else {
    _console
        .writeln("\nSales order is not created because 0 or more than 1 customer
records found for the entityID given");
}
} else {
    _console.error(getStatusDetails(res.getStatus()));
}

```

```

    }
}

```

Setting the Shipping Address on a Sales Order

The following code illustrates how to set the shipTo address on a sales order.

SOAP Request

```

<soapenv:Body>
<add xmlns="urn:messages_1_3.platform.webservices.netsuite.com">
<record xsi:type="ns6:SalesOrder" xmlns:ns6="urn:sales_1_3.transactions.webservices.netsuite.com">
  <ns6:entity internalId="103 customer" xsi:type="ns7:RecordRef"
    xmlns:ns7="urn:core_1_3.platform.webservices.netsuite.com">
    <ns7:name xsi:type="xsd:string">Vintrust</ns7:name>
  </ns6:entity>
  <ns6:tranDate xsi:type="xsd:dateTime">2006-03-07T18:16:44.000Z</ns6:tranDate>
  <ns6:orderStatus xsi:type="ns8:SalesOrderOrderStatus"
    xmlns:ns8="urn:types.sales_1_3.transactions.webservices.netsuite.com">_pendingApproval
  </ns6:orderStatus>
  <ns6:shipAddressList internalId="84" xsi:type="ns9:RecordRef"
    xmlns:ns9="urn:core_1_3.platform.webservices.netsuite.com"/>
  <ns6:itemList replaceAll="true" xsi:type="ns6:SalesOrderItemList">
    <ns6:item xsi:type="ns6:SalesOrderItem">
      <ns6:item internalId="18 inventoryItem" xsi:type="ns10:RecordRef"
        xmlns:ns10="urn:core_1_3.platform.webservices.netsuite.com"/>
      <ns6:quantity xsi:type="xsd:double">2.0</ns6:quantity>
    </ns6:item>
  </ns6:itemList>
</record>
</add>
</soapenv:Body>

```

.NET

```

Customer customer = // some customer
CustomerAddressbook shipTo = // some customer address book entry
RecordRef rr = new RecordRef();
rr.setInternalId(shipTo.getInternalId());
rr.setName("addressBook");
salesOrder.setShipAddressList(rr);

```

Java

```

//block setting shipping address to something other than default
RecordRef altShipAddress = new RecordRef();
altShipAddress.setInternalId("84");
so.setShipAddressList(altShipAddress);
//end block
WriteResponse writeRes = _port.add(so);

```

Working with Sales Order Sublists

The SuiteTalk Schema Browser includes all sublists associated with the Sale Order record. See the following information for usage notes regarding specific Sales Order sublists. Note that usage notes are not provided for every sublist type.

SalesOrderSalesTeamList

This list is only available when the Team Selling feature is enabled.

Calculating Shipping Cost for Sales Orders

Web services calculation of shipping cost for sales orders varies slightly from the UI behavior. Because the sales order includes a Calculate Shipping button in the UI, updates to item quantity or other values that may impact shipping cost do not automatically recalculate that field's value in the UI. In Web services, updates to item quantity or other values that impact shipping cost always result in recalculation of altshippingcost. If your account is set up to automatically charge for shipping, this defaults to the value that corresponds to the shipping method selected. To prevent this recalculation, resubmit the value for altshippingcost whenever you update item quantity or other fields that impact shipping cost.

Note that this Web services behavior applies when the Multiple Shipping Routes feature is not enabled. For details about Web services behavior with this feature enabled, see [Multiple Shipping Routes in Web Services](#).

Sales Order Payment Terms and Payment Methods

The NetSuite UI validates payment method values against payment terms values on sales order records, preventing values that do not match. In endpoints prior to 2010.1, Web services allows the saving of both fields' values, even if they do not match. This behavior may result in unexpected errors on payments. It is recommended that you do not attempt to set payment method and terms at the same time for endpoints prior to 2010.1.

Associating Sales Orders with Opportunities

You can associate estimates, cash sales, sales orders, and invoices with opportunities. Once a transaction other than an estimate is associated with an opportunity, the opportunity's status is automatically set to **Closed Won**. Once an opportunity's status is set to **Closed Won**, it is no longer available to be selected on other cash sale, sales order, or invoice records.

Invoice

An invoice is a record of sale to a customer. Submit an invoice when payment for goods or services is not received at the time of delivery.

Record sales on invoices by adding each item sold from your Items list. Each invoice consists of line-items and their sales amounts. The sum of all sales amounts on an invoice equals the total amount the customer owes.

The Invoice record is defined in the [tranSales \(sales\)](#) XSD.

Supported Operations

The following operations can be used with the Invoice record:

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | initialize / initializeList | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transaction sales.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Working with Invoice Sublists

The SuiteTalk Schema Browser includes all sublists associated with the Invoice record. See the following information for usage notes regarding specific Invoice sublists. Usage notes are not provided for every sublist type.

- [InvoiceSalesTeamList](#)
- [InvoiceItemCostList](#)
- [InvoiceExpCostList](#)
- [InvoiceTimeList](#)
- [GiftCertRedemptionList](#)

[InvoiceSalesTeamList](#)

The InvoiceSalesTeam list defines the sales team for a given invoice. This list is only available when the Team Selling feature is enabled in the account.

On add, a single line item is entered by default when a sales rep is currently associated with the entity set in the entity field. When a sales group is currently associated with the entity, the list is populated by the group.

[InvoiceItemCostList](#)

These fields map to the Billable Items subtab on the Item list of an invoice record. This is where you define how and when to bill item costs back to a customer. Billing items back to customers enables you to purchase items and supplies for an order or job, and then bill the cost to the customer. The Bill Costs To Customers feature must be enabled in order to use this list.

[InvoiceExpCostList](#)

These fields map to the Billable Expenses subtab on the Item list of an invoice record. This is where you define how and when to bill expenses back to a customer. The Bill Costs To Customers feature must be enabled in order to use this list.

[InvoiceTimeList](#)

These fields map to the Billable Time subtab on the Item list of an invoice record. This is where you define how and when to bill time back to a customer. The Bill Costs To Customers feature must be enabled in order to use this list.

GiftCertRedemptionList

This sublist is available on the Invoice, Sales Order, and Cash Sale records.

Associating Invoices with Opportunities

You can associate estimates, cash sales, sales orders, and invoices with opportunities. Once a transaction other than an estimate is associated with an opportunity, the opportunity's status is automatically set to **Closed Won**. Once an opportunity's status is set to **Closed Won**, it is no longer available to be selected on other cash sale, sales order, or invoice records.

Journal Entry

A general journal entry records debits and credits to be posted to ledger accounts. General journal entries adjust the value of any set of accounts without entering transactions, such as invoices or bills.

The Journal Entry record is defined in the [tranGeneral](#) XSD.

Supported Operations

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | isearch | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions general.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

You must have at least one line item in order to submit a journal entry transaction. The maximum number of Journal Entries you can submit in one transaction is **200**, like all other Web services transactions.

Intercompany Journal Entry

The Intercompany Journal Entry record is defined in [tranGeneral \(general\)](#) XSD.

If you are using subsidiary management and consolidation in your NetSuite account, you can post intercompany journal entries to keep accurate consolidated books.

Supported Operations

The following operations can be used with the Intercompany Journal Entry record.

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions general.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Customer Deposit

The Customer Deposit record is defined in the [tranCust \(customers\)](#) XSD.

Supported Operations

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | initialize / initializeList | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions customers.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Deposit Application

The Deposit Application record is defined in [tranCust \(customers\)](#) XSD.

Supported Operations

add | addList | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | initialize / initializeList | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions customers.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Item Fulfillment

The Item Fulfillment record is defined in the [tranSales \(sales\) XSD](#).

Supported Operations

The following operations can be used to manipulate the item fulfillment record.

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | initialize / initializeList | search | searchMore | searchNext | update | updateList

Field Definitions

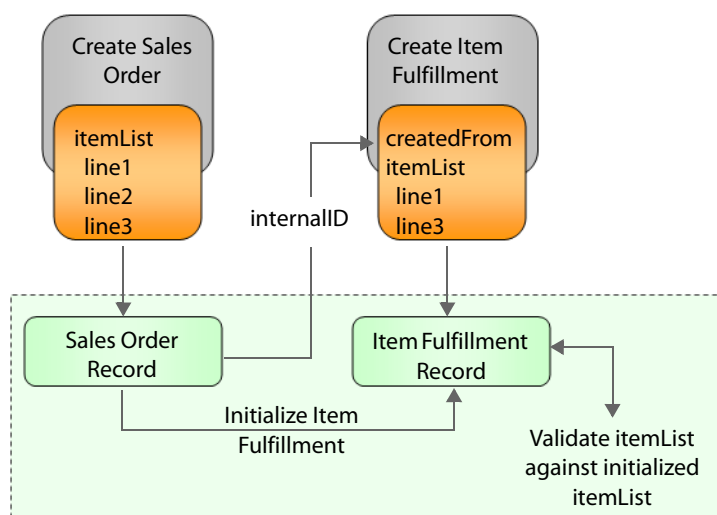
The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [transactions sales.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Item Fulfillment Workflow

When submitting an Item Fulfillment record, NetSuite initializes the item fulfillment record on the server with data from the Sales Order referenced in the **createdFrom** field. (In cases of transfers of inventory, the record is initialized with data from the Transfer Order or Intercompany Transfer Order.) Then the item fulfillment record is updated with the item list provided in your request and that data is validated against the initialized data from the Sales Order. If validation succeeds, the Item List respects the `replaceAll=TRUE` attribute just like any other sublist and **replaces** the item list for the given item fulfillment record.



Important: Validation for Fed Ex or UPS related fields occurs once a record is routed to FedEx or UPS respectively and NOT by the NetSuite application just as in the UI. Therefore, it is the responsibility of the client application developer to ensure that the correct fields are populated with the required information for shipping.

Sample Code

The following code illustrates adding a Sales Order record with a single item and then fulfilling that order. In this case, the account is set up for shipping with FedEx and FedEx is set as the method of shipping on the Sales Order record.

First, a sales order is created, set to the pendingFulfillment status, and populated with an itemList.

Create Sales Order Java

```
sessMgr.login();
SalesOrder salesOrder = new SalesOrder();
salesOrder.setEntity(Util.makeRecordRef("87", RecordType.customer));
salesOrder.setOrderStatus(SalesOrderOrderStatus._pendingFulfillment);
SalesOrderItem soi = new SalesOrderItem();
soi.setItem (Util.makeRecordRef("15", RecordType.inventoryItem));
soi.setQuantity(new Double (1));
soi.setAmount(new Double (14));
SalesOrderItemList soil = new SalesOrderItemList(new SalesOrderItem[] {soi}, true);
salesOrder.setItemList(soil);
salesOrder = sessMgr.getNetsuitePort().add(salesOrder);
```

Create Sales Order SOAP Request

```
<soapenv:Body>
<add xmlns="urn:messages_2_0.platform.webservices.netsuite.com">
<record xsi:type="ns1:SalesOrder" xmlns:ns1="urn:sales_2_0.transactions.webservices.netsuite.com">
<ns1:entity internalId="87 customer" xsi:type="ns2:RecordRef"
xmlns:ns2="urn:core_2_0.platform.webservices.netsuite.com"/>
<ns1:orderStatus xsi:type="ns3:SalesOrderOrderStatus"
xmlns:ns3="urn:types.sales_2_0.transactions.webservices.netsuite.com">_pendingFulfillment</
ns1:orderStatus>
<ns1:itemList replaceAll="true" xsi:type="ns1:SalesOrderItemList">
<ns1:item xsi:type="ns1:SalesOrderItem">
<ns1:item internalId="15 inventoryItem" xsi:type="ns4:RecordRef"
xmlns:ns4="urn:core_2_0.platform.webservices.netsuite.com"/>
<ns1:quantity xsi:type="xsd:double">1.0</ns1:quantity>
<ns1:amount xsi:type="xsd:double">14.0</ns1:amount>
</ns1:item>
</ns1:itemList>
</record>
</add>
</soapenv:Body>
```

Using a Get operation on the Sales Order created above, the itemList information can be determined.

Get Sales Order SOAP Request

```

<soapenv:Body>
<get xmlns="urn:messages_2_0.platform.webservices.netsuite.com">
<baseRef internalId="1505 salesOrder" xsi:type="ns1:RecordRef"
xmlns:ns1="urn:core_2_0.platform.webservices.netsuite.com"/>
</get>
</soapenv:Body>

```

Get Sales Order SOAP Response

```

<soapenv:Body>
<getResponse xmlns="urn:messages_2_0.platform.webservices.netsuite.com">
<readResponse xmlns="urn:messages_2_0.platform.webservices.netsuite.com">
<ns1:status isSuccess="true" xmlns:ns1="urn:core_2_0.platform.webservices.netsuite.com"/>
<record internalId="1505" xsi:type="ns2:SalesOrder" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:ns2="urn:sales_2_0.transactions.webservices.netsuite.com">
<ns2:entity internalId="87">
<ns3:name xmlns:ns3="urn:core_2_0.platform.webservices.netsuite.com">Abe Simpson</ns3:name>
</ns2:entity>
<ns2:tranDate>2006-08-22T00:00:00.000-07:00</ns2:tranDate>
<ns2:tranId>SORD10069</ns2:tranId>
<ns2:orderStatus>_pendingFulfillment</ns2:orderStatus>
<ns2:salesRep internalId="43">
<ns4:name xmlns:ns4="urn:core_2_0.platform.webservices.netsuite.com">Jon Baker</ns4:name>
</ns2:salesRep>
<ns2:leadSource internalId="-2">
<ns5:name xmlns:ns5="urn:core_2_0.platform.webservices.netsuite.com">Ad</ns5:name>
</ns2:leadSource>
<ns2:excludeCommission>>false</ns2:excludeCommission>
<ns2:isTaxable>>false</ns2:isTaxable>
<ns2:toBePrinted>>false</ns2:toBePrinted>
<ns2:toBeEmailed>>false</ns2:toBeEmailed>
<ns2:email>asimpson@boo.com</ns2:email>
<ns2:toBeFaxed>>false</ns2:toBeFaxed>
<ns2:billAddress>Abe Simpson<br><br>34 Elm St<br><br>Great Falls MT</ns2:billAddress>
<ns2:shipAddress>Abe Simpson<br><br>34 Elm St<br><br>Great Falls MT</ns2:shipAddress>
<ns2:shipDate>2006-08-22T00:00:00.000-07:00</ns2:shipDate>
<ns2:subTotal>14.0</ns2:subTotal>
<ns2:total>14.0</ns2:total>
<ns2:balance>0.0</ns2:balance>
<ns2:lastModifiedDate>2006-08-22T15:49:00.000-07:00</ns2:lastModifiedDate>
<ns2:status>Pending Fulfillment</ns2:status>
<ns2:itemList>
<ns2:item>
<ns2:item internalId="15">
<ns6:name xmlns:ns6="urn:core_2_0.platform.webservices.netsuite.com">Tongue Depressor</ns6:name>
</ns2:item>
<ns2:quantity>1.0</ns2:quantity>
<ns2:price internalId="1">
<ns7:name xmlns:ns7="urn:core_2_0.platform.webservices.netsuite.com">Base Price</ns7:name>
</ns2:price>
<ns2:rate>14.00</ns2:rate>
<ns2:amount>14.0</ns2:amount>
<ns2:commitInventory>_availableQty</ns2:commitInventory>
<ns2:isTaxable>true</ns2:isTaxable>
<ns2:isClosed>>false</ns2:isClosed>
<ns2:line>1</ns2:line>

```

```

<ns2:quantityBackOrdered>0.0</ns2:quantityBackOrdered>
<ns2:quantityBilled>0.0</ns2:quantityBilled>
<ns2:quantityCommitted>1.0</ns2:quantityCommitted>
<ns2:quantityFulfilled>0.0</ns2:quantityFulfilled>
</ns2:item>
</ns2:itemList>
</record>
</readResponse>
</getResponse>
</soapenv:Body>

```

Next the sales order created above is referenced by setting the `createdFrom` field to the `internalId` of the Sales Order. The specific items to be fulfilled are referenced by setting the `Line` field to the desired item line from the Sales Order.

Item Fulfillment Java

```

ItemFulfillment itemFulfillment = new ItemFulfillment();
itemFulfillment.setCreatedFrom(Util.makeRecordRef(salesOrder.getInternalId(), RecordType.salesOrder));
ItemFulfillmentItemList ifil = new ItemFulfillmentItemList();
ItemFulfillmentItem ifi = new ItemFulfillmentItem();
ifi.setOrderLine(salesOrder.getItemList().getItem(0).getLine());
ifil.setItem(new ItemFulfillmentItem[]{ifi});
itemFulfillment.setItemList(ifil);
itemFulfillment = sessMgr.getNetsuitePort().add(itemFulfillment);

```

Item Fulfillment SOAP Request

```

<soapenv:Body>
<add xmlns="urn:messages_2_0.platform.webservices.netsuite.com">
<record xsi:type="ns1:ItemFulfillment" xmlns:ns1="urn:sales_2_0.transactions.webservices.netsuite.com">
<ns1:createdFrom internalId="1505 salesOrder" xsi:type="ns2:RecordRef"
xmlns:ns2="urn:core_2_0.platform.webservices.netsuite.com"/>
<ns1:itemList replaceAll="false" xsi:type="ns1:ItemFulfillmentItemList">
<ns1:item xsi:type="ns1:ItemFulfillmentItem">
<ns1:orderLine xsi:type="xsd:long">1</ns1:orderLine>
</ns1:item>
</ns1:itemList>
</record>
</add>
</soapenv:Body>

```

Important: Item fulfillment creation is affected by the preference set at Setup > Accounting > Accounting Preferences > Order Management tab > Fulfillment Based on Commitment. The best practice for creating an item fulfillment in Web services is to use the `initialize` operation, as it returns the item fulfillment with all the defaults set and users will not have to specify the item quantities themselves. Following this practice prevents users from generating a partially fulfilled order without knowing it. (See [initialize / initializeList](#) in the NetSuite Help Center for details on this operation.)

Item Receipt

The Item Receipt record is defined in [tranPurch \(purchases\)](#) XSD.

Supported Operations

add | addList | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | initialize / initializeList | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions purchases.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Transfer Order

The Transfer Order transaction is used to move inventory between locations when the Multi-Location Inventory feature is enabled. Existing integrations with external warehouse management systems can leverage this transaction to manage data about inventory movement between locations. The Transfer Order record is defined in the [tranInvt \(inventory\)](#) XSD.

Transfer orders can initialize item fulfillment and item receipt transactions. See [Item Fulfillment](#) and [Item Receipt](#) for details about these transactions.

Supported Operations

The following operations can be used to modify Transfer Order records:

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | initialize / initializeList | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions inventory.xsd** in the [SuiteTalk Schema Browser](#) for details for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Transfer Order Features and Preferences

The Transfer Order transaction is available when the Multi-Location Inventory feature is enabled. Note that the Pick, Pack and Ship feature can affect transfer order workflow.

The following accounting preferences, available at Setup > Accounting > Accounting Preferences, on the Order Management subtab, apply to transfer orders:

- Default Transfer Order Status indicates whether transfer orders require approval by default.
 - Pending Approval means that by default the status of someone with permission must approve the order before it is processed.
 - Pending Fulfillment means that by default transfer orders are sent directly to the fulfillment queue without requiring further approval.
- Use Item Cost as Transfer Cost affects how the Transfer Price from each transfer order is used on the related item receipt transaction.
 - When this option is enabled, the transfer price is used as a declared shipping value for reference only, such as for insurance or international shipping, and the item cost from the item record is used on the item receipt.
 - When this option is disabled, the transfer price on the transfer order is used as the item cost on the item receipt. In this case, both the item fulfillment and the item receipt resulting from the transfer order have G/L impact.

Transfer Order Workflow

The following is a rough outline of transfer order workflow. For more details, see the help topic Inventory Transfer Orders.

1. A transfer order is entered to schedule the movement of items and its status is set to either Pending Approval or Pending Fulfillment.

A transfer order can be approved by setting the status to Pending Fulfillment.

Note: A transfer order initially can be saved without a source location value, for example, in a case where an item is needed at the destination location but the source location has not yet been identified. However, the source location value is required for an approved transfer order.

2. Once its status is set to Pending fulfillment, the transfer order can be fulfilled from the source location.
 - An item receipt transaction is created.
 - Items are committed out of the source location's inventory.
 - The On Order quantity of items increases at the destination location.
 - Fulfilled items are removed from the On Hand count at the source location.

- The value of items in transit are removed from the Inventory Asset account and added to the Inventory in Transit account for the source location.
 - The status of the transfer order is set to Pending Receipt, or if the Pick, Pack and Ship feature is enabled, to Shipped.
3. Once the transfer order status has been set to Pending Receipt (or to Shipped), Transfer order items can be received at the destination location.
- An item receipt transaction is created.
 - The items are added to the destination location's inventory and increase the On Hand count.
 - The items' value is added to the Inventory Asset account for the destination location.
 - The On Order quantity of the item in the destination location decreases.

Serialized and Lot Inventory Transfer Orders

Serial or lot numbered items can be included as transfer order line items. Note the following requirements:

- Serial or lot numbers are limited to 3800 characters.
- Supported separators for serial numbers are: space, comma, or line break.
- A serial number is required for every serialized line item.

For example, if the quantity of a line item is 2, two serial numbers are required.

- Lot numbers must use the following format: <Lot #><Quantity>

For example, to indicate a quantity of 100 for an item with lot number AAA-3400, format would be: AAA-3400(100)

Intercompany Transfer Order

In accounts using NetSuite OneWorld, the Intercompany Transfer Order transaction is used to move inventory from a location for one subsidiary to a location for another subsidiary. The Intercompany Transfer Order record is defined in the [tranInvt \(inventory\)](#) XSD.

For information about how intercompany transfer orders are used, see the help topic [Intercompany Inventory Transfers](#).

Supported Operations

The following operations can be used to modify Intercompany Transfer Order records:

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | initialize / initializeList | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions inventory.xsd** in the [SuiteTalk Schema Browser](#) for details for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

It is recommended that you read the usage notes for [Transfer Order](#), because these notes also apply to the Intercompany Transfer Order. The following additional notes apply.

Source Subsidiary and Destination Subsidiary

An intercompany transfer order must have a Subsidiary (meaning the source subsidiary) and a Destination Subsidiary value.

All transfer order line items must be available to both the source and destination subsidiaries.

The Source Location for a transfer order must be from the (source) Subsidiary and the Destination Location must be from the Destination Subsidiary.

The Department and Class values for a transfer order must be from the (source) Subsidiary. These values are used for shipment transactions.

You must have permissions for both the source (Subsidiary) and Destination Subsidiary to create or edit an intercompany transfer order. Permissions for both the Source Location and Destination location also are required, whether the transfer order status is set to Pending Approval or Pending Fulfillment.

Use Item Cost as Transfer Cost Preference

Intercompany transfer orders are only available when transfer pricing is used, meaning the Use Item Cost as Transfer Cost, at Setup > Accounting > Accounting Preferences, on the Order Management subtab, must be disabled.

Transfer Pricing Notes

For intercompany transfer orders, the Transfer Price on the transfer order is always used as the item cost for accounting purposes.

The transfer order Transfer Price value defaults to the Transfer Price listed on the item record, but can be changed when the transfer order is created or edited.

- The item record Transfer Price is in the currency of the root parent subsidiary. On the transfer order, the Transfer Price should be converted to the correct value in the currency of the source subsidiary.
- If the item record Transfer Price is not set, the Transfer Price on the transfer order is 0. In this case, be sure to edit the transfer order Transfer Price as necessary.

The difference between the item cost and the transfer order Transfer Price is recorded in a Gain/Loss account, and G/L impact occurs after item receipt, as shown in the following table:

Account	Debit	Credit	Subsidiary
Inventory Asset	X		Destination Subsidiary
Inventory in Transit		X	Source Subsidiary
Intercompany Payable/Receivable		X	Destination Subsidiary (Payable)
Intercompany Payable/Receivable	X		Source Subsidiary (Receivable)

Be aware that the exchange rate between subsidiaries' currencies can affect the G/L impact to each subsidiary.

Checks

The Checks records is defined in the [tranBank \(bank\)](#) XSD.

Supported Operations

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions bank.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Inventory Adjustment

Use the Inventory Adjustment record to change the quantity and value of an inventory item to account for clerical errors, major changes in cost, thefts or miscounts.

Important: NetSuite recommends that you do not delete or change inventory transactions dated prior to an inventory distribution, as this can cause difficulties maintaining accurate inventory data.

The Inventory Adjustment record is defined in the [tranInvt \(inventory\)](#) XSD.

Supported Operations

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions inventory.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Return Authorization

Return authorizations are records of expected customer returns. The Return Authorizations Sales Transactions feature must be enabled in an account in order to submit this record.

The Return Authorization record is defined in the [tranCust \(customers\)](#) XSD.

Supported Operations

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | initialize / initializeList | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transaction customers.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Working with Return Authorization Sublists

The SuiteTalk Schema Browser includes all sublists associated with the Return Authorization record. See the following information for usage notes regarding specific Return Authorization sublists. Usage notes are not provided for every sublist type.

ReturnAuthorizationItemList

In order to access per line tax fields, the Per-Line Taxes on Transactions feature must be enabled.

Credit Memo

Use the credit memo record to decrease the amount a customer owes you.

The Credit Memo record is defined in the [tranCust \(customers\)](#) XSD.

Supported Operations

The following operations can be used with Credit Memo records:

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | initialize / initializeList | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions customers.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Cash Sale

Use the cash sale record to enter the sale of goods or services for which you receive immediate payment.

The Cash Sale record is defined in the [tranSales \(sales\) XSD](#).

Supported Operations

The following operations can be used with Cash Sale records:

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | initialize / initializeList | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions sales.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Working with Cash Sale Sublists

The SuiteTalk Schema Browser includes all sublists associated with the Cash Sale record. See the following information for usage notes regarding specific Cash Sale sublists. Usage notes are not provided for every sublist type.

- [CashSaleSalesTeamList](#)
- [Cash SaleItemCostList](#)

- CashSaleExpCostList
- CashSaleTimeList
- GiftCertRedemptionList

CashSaleSalesTeamList

The CashSaleSalesTeam list defines the sales team for a given cash sale. This list is only available when the Team Selling feature is enabled in the account.

On add, a single line item is entered by default when a sales rep is currently associated with the entity set in the entity field. When a sales group is currently associated with the entity, the list is populated by the group.

Cash SaleItemCostList

These fields map to the Billable Items subtab on the Item list of a cash sale record. This is where you define how and when to bill item costs back to a customer. Billing items back to customers enables you to purchase items and supplies for an order or job, and then bill the cost to the customer. The Bill Costs To Customers feature must be enabled in order to use this list.

CashSaleExpCostList

These fields map to the Billable Expenses subtab on the Item list of a cash sale record. This is where you define how and when to bill expenses back to a customer. The Bill Costs To Customers feature must be enabled in order to use this list.

CashSaleTimeList

These fields map to the Billable Time subtab on the Item list of a cash sale record. This is where you define how and when to bill time back to a customer. The Bill Costs To Customers feature must be enabled in order to use this list.

GiftCertRedemptionList

This sublist is available on the Invoice, Sales Order, and Cash Sale records.

Promotion Code Validation

Note that validation of promotion code date range for a cash sale transaction generated from a sales order is different in Web services than in the user interface. In the user interface, as long as sales order date was within the promotion code date range, cash sale can be generated without error, even if cash sale date is outside of promotion code date range. In Web services, cash sale date must be within promotion code date range.

Associating Cash Sales with Opportunities

You can associate estimates, cash sales, sales orders, and invoices with opportunities. Once a transaction other than an estimate is associated with an opportunity, the opportunity's status is automatically set to **Closed Won**. Once an opportunity's status is set to **Closed Won**, it is no longer available to be selected on other cash sale, sales order, or invoice records.

Estimate/Quote

Use the Estimate record to create an estimate and print, email or fax it to your customer. Once a customer accepts the estimate, it can be converted into a sales order, invoice or cash sale. To use this record, the Estimates feature must be enabled at Setup > Company > Enable Features. On the Transactions tab, click the Estimates check box, which appears in the Basic Features portlet.

Note: Estimates have no accounting impact until they are converted into invoices or cash sales.

The Estimate/Quote record is defined in the [tranSales \(sales\)](#) XSD.

Supported Operations

The following operations can be used with Estimate records:

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | initialize / initializeList | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions sales.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Cash Refund

Use the cash sale refund to return money back to a customer who immediately paid for goods or services using cash, a check or a credit card.

The Cash Refund record is defined in the [tranCust \(customers\)](#) XSD.

Supported Operations

The following operations can be used with the Cash Refund record:

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | initialize / initializeList | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions customers.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Customer Payment

Use the Customer Payment record to record a customer payment.

The Customer Payment record is defined in the `tranCust (customers)` XSD.

Supported Operations

The following operations can be used with Customer Payment records:

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | initialize / initializeList | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See `transactions customers.xsd` in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

If you specify the “total amount of the original payment” as the payment value in a customer payment, `autoApply` through Web services fails when there has already been an amount applied previously on the payment. This process fails when `autoApply` is set to `true`, and an invoice with `apply = true` already exists in the customer payment.

To work around this, you must set `apply=false` for the invoice. The following is the recommended workflow:

GET: (get the existing customer payment)

```
<get xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
  <baseRef internalId="2262" type="customerPayment" xsi:type="ns1:RecordRef"
  xmlns:ns1="urn:core_2_6.platform.webservices.netsuite.com"/>
</get>
```

RESPONSE: (the response will return all the invoices that were already applied)

```
<ns3:applyList>
  <ns3:apply>
    <ns3:apply>true</ns3:apply>
  </ns3:apply>
  <ns3:doc>1929</ns3:doc>
  <ns3:total>27.92</ns3:total>
  <ns3:due>27.92</ns3:due>
  <ns3:amount>27.92</ns3:amount>
</ns3:apply>
<ns3:apply>
  <ns3:apply>true</ns3:apply>
</ns3:apply>
```

```

<ns3:doc>1930</ns3:doc>
<ns3:total>27.92</ns3:total>
<ns3:due>27.92</ns3:due>
<ns3:amount>27.92</ns3:amount>
</ns3:apply>
</ns3:applyList>

```

UPDATE: (update the customer payment - set paymentAmount to the original amount, set autoApply = true, **AND** set apply = false on all the invoices that have been already applied)

```

<update xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
<record internalId="2262" xsi:type="ns1:CustomerPayment"
xmlns:ns1="urn:customers_2_6.transactions.webservices.netsuite.com">
<ns1:customer internalId="176" xsi:type="ns2:RecordRef"
xmlns:ns2="urn:core_2_6.platform.webservices.netsuite.com"/>
<ns1:payment xsi:type="xsd:double">80.0</ns1:payment>
<ns1:autoApply xsi:type="xsd:boolean">true</ns1:autoApply>
<ns1:applyList replaceAll="false" xsi:type="ns1:CustomerPaymentApplyList">
<ns1:apply xsi:type="ns1:CustomerPaymentApply">
<ns1:apply xsi:type="xsd:boolean">false</ns1:apply>
<ns1:doc xsi:type="xsd:long">1929</ns1:doc>
</ns1:apply>
<ns1:apply xsi:type="ns1:CustomerPaymentApply">
<ns1:apply xsi:type="xsd:boolean">false</ns1:apply>
<ns1:doc xsi:type="xsd:long">1930</ns1:doc>
</ns1:apply>
</ns1:applyList>
</record>
</update>

```

GET: (perform a get to verify that it worked)

```

<get xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
<baseRef internalId="2262" type="customerPayment" xsi:type="ns1:RecordRef"
xmlns:ns1="urn:core_2_6.platform.webservices.netsuite.com"/>
</get>

```

RESPONSE: (this response shows that the update worked. The invoices that were applied before are still applied, and the rest of the Payment Amount was applied to new invoices)

```

<ns3:applyList>
<ns3:apply>
<ns3:apply>true</ns3:apply>
<ns3:doc>1933</ns3:doc>
<ns3:total>27.92</ns3:total>
<ns3:due>27.92</ns3:due>
<ns3:amount>24.16</ns3:amount>
</ns3:apply>
<ns3:apply>
<ns3:apply>true</ns3:apply>
<ns3:doc>1929</ns3:doc>
<ns3:total>27.92</ns3:total>
<ns3:due>27.92</ns3:due>
<ns3:amount>27.92</ns3:amount>
</ns3:apply>
<ns3:apply>
<ns3:apply>true</ns3:apply>
<ns3:doc>1930</ns3:doc>

```

```
<ns3:total>27.92</ns3:total>  
&lt;...
```

Customer Refund

Use the customer refund record to return money to a customer who paid for goods or services using cash, a check or a credit card.

The Customer Refund record is defined in the [tranCust \(customers\)](#) XSD.

Supported Operations

The following operations can be used to modify Customer Refund records:

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | initialize / initializeList | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [transactions customers.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Vendor Bill

Use the Vendor Bill record to track payables by entering bills as they arrive from vendors and pay them from the payables list as they are due.

The Vendor Bill record is defined in the [tranPurch \(purchases\)](#) XSD.

Supported Operations

The following operations can be used with Vendor Bill records:

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | initialize / initializeList | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [transactions purchases.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Vendor Payment

The Vendor Payment record is defined in the [tranPurch \(purchases\)](#) XSD.

Supported Operations

The following operations can be used with Vendor Payment records:

add | addList | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | initialize / initializeList | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions purchases.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

As of the 2011.1 endpoint, you can use {doc, line} on the VendorPaymentApply sublist to identify the vendor bill line to which a payment should be applied. You can use the line element to uniquely identify a sublist entry when the doc element is not sufficient. The line element prevents problems with record creation due to duplicate keys.

Use the initialize operation to obtain the doc and line values from the vendor bill.

Warning: As of the 2011.1 endpoint, if you do not provide a value for the line element for an entry to VendorPaymentApply, you may receive the error: "Ambiguous sublist reference; multiple matches exist for key value <doc>". Earlier endpoints do not return this error.

Time Bill (Track Time)

Use time tracking to record the hours worked by employees. You can record billable hours and invoice your customers for them. The Time Tracking feature must be enabled at Setup > Company > Enable Features > Employees.

The TimeBill record is defined in the [tranEmp \(employees\)](#) XSD.

Important: The TimeBill record is labelled as the Track Time record in the UI.

Supported Operations

The following operations can be used with TimeBill records:

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions employees.xsd** the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Expense Report

The Expense Report record is defined in the [tranEmp \(employees\)](#) XSD.

Supported Operations

The following operations can be used with Expense Report records:

add | addList | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **transactions employees.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Multiple Shipping Routes in Web Services

The following topics are covered in this section. If you are unfamiliar with the Multiple Shipping Routes feature, it is recommended that you read each topic sequentially.

- What is the Multiple Shipping Routes feature?
- How does MSR work in Web Services?
- Which fields and elements are associated with MSR?
- Does MSR work on custom forms?
- Multiple Shipping Routes Code Samples for Web Services

What is the Multiple Shipping Routes feature?

The Multiple Shipping Routes (MSR) feature allows you to associate several items with one transaction, and then set different shipping addresses and shipping methods for each item. Transaction types such as sales order, cash sale, invoice, estimate, and item fulfillment all support MSR.

The following figure shows a sales order with two items. When MSR is enabled in your account, **and** the *Enable Item Line Shipping* check box is selected on the transaction, each item can have its own shipping address and shipping method. The shipping address is specified in the *Ship To* column; the shipping method is specified in the *Ship Via* column.

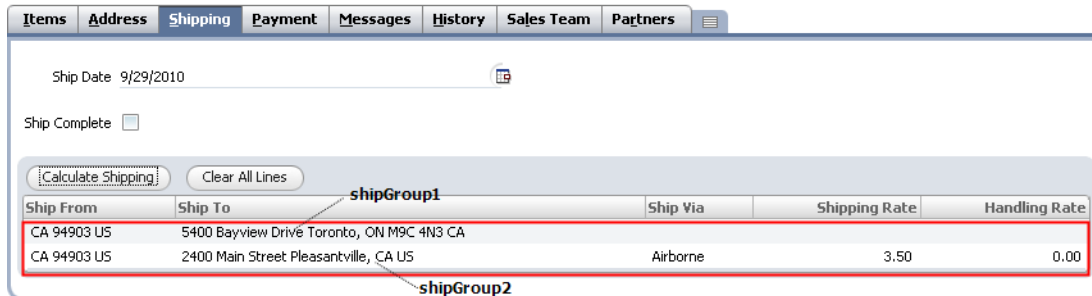
The screenshot shows a software interface with a navigation bar at the top containing tabs: Items, Address, Shipping, Payment, Messages, History, Sales Team, and Partners. Below the navigation bar, there are several fields: 'Enable Item Line Shipping' (checked), 'isMultiShipTo', 'Discount', '*Exchange Rate 1.00', 'Rate', 'Promotion Code', and 'Currency USA'. Below these fields are three buttons: 'Add Multiple', 'Upsell Items', and 'Clear All Lines'. Below the buttons is a table with the following columns: *Item, Quantity, Units, Description, Price Level, Rate, Amount, *Tax Code, Commit, Tax Rate, Options, Ship To, Carrier, and Ship Via. The table contains two rows of data:

*Item	Quantity	Units	Description	Price Level	Rate	Amount	*Tax Code	Commit	Tax Rate	Options	Ship To	Carrier	Ship Via
Product (Standard) : Anti-Virus 11.0 Standard	1		McGeever Anti-Virus 11.0 Standard	Monthly	37.99	37.99	-Not Taxable-	Available Qty	0.0%		5400 Bayview Drive		UPS 2nd Day Air A.M.®
Product (Professional) : Socket Manager 1.0 Professional	1		McGeever Socket Manager 1.0 Professional	Monthly	29.99	29.99	-Not Taxable-	Available Qty	0.0%		2400 Main Street		Airborne

In the UI, after all items have been added to the transaction (a sales order in this example), you must then create individual shipping groups by clicking the **Calculate Shipping** button on the Shipping tab. A shipping group includes all item details related to where the item is being shipped from, where it is being shipped to, the item’s shipping method, and its shipping rate.

Although there is no UI label called “Shipping Group,” Web services developers can verify that shipping groups have been created by doing a *get* on the transaction after it has been submitted.

This figure shows the UI equivalent of two separate shipping groups on a sales order.



Note: For additional information on MSR, as well as steps for enabling MSR in your account, see *Multiple Shipping Routes* in the NetSuite Help Center.

How does MSR work in Web Services?

Generally speaking, NetSuite Web services attempts to programmatically mirror the actions a user takes in the UI. However, when working with MSR-enabled transactions in Web services, developers should be aware of the following UI vs. Web services distinctions:

- If you attempt to set items and shipping rates in the same request, the following message is returned:
You cannot update items and shipping rates at the same time on transactions that have multiple shipping routes enabled. You must first update the items, then *get* the transaction and update the shipping rates separately.
- If you make any update to any item on MSR-enabled transactions, this action may result in changes to the shipping cost. Every time an item is updated and the record is submitted, NetSuite re-calculates the shipping rate. NetSuite calculates all orders based on “real-time” shipping rates.
- There is no Web services equivalent of the Calculate Shipping button that appears on the Shipping subtab. In Web services, shipping calculations are handled by the NetSuite backend when the transaction is submitted. Note that you can retrieve the shipping cost if you do a *get* on the transaction with the *bodyFieldsOnly* search preference set to false.

- The only *initialize* workflow that is impacted with MSR enabled is the sales order to fulfillment workflow. Invoicing and other transaction workflows are not impacted.
- The initialize operation includes an optional argument called *AuxReference*. In *AuxReference* users can set the type to *shippingGroup* by using the enum object *InitializeAuxRefType*. When working with MSR-enabled transactions, you must specify a value for *shippingGroup*. If you do not specify a value, the value 1 (for the first shipping group) is defaulted. This means that only the items belonging to the first shipping group will be fulfilled when the sales order is initialized. For a code sample, see [Initialize the sales order to create the item fulfillment](#).

Which fields and elements are associated with MSR?

The following table lists UI field labels and their corresponding Web services elements for all MSR-related fields.

UI Label	Element Name	Note
Enable Item Line Shipping	isMultishipTo	Set to <i>true</i> to allow for multiple items with separate shipping address/methods
Ship To	shipAddress	References the internal ID of the customer's shipping address. You can get the internal ID by clicking the Address tab on the customer's record. The address ID appears in the ID column. Note: The Show Internal ID preference must be enabled for address IDs to show.
Ship Via	shipMethod	References the internal ID of the item. Go to the Items list to see all item IDs. Note: The Show Internal ID preference must be enabled for address IDs to show.
There is no UI label for Shipping Group.	shipGroup	Each item that has a separate shipping address/shipping method will have a unique shipGroup number. When you initialize a sales order to create a fulfillment, and the sales order has MSR enabled, you will need to specify a shipGroup value (1, 2, 3, etc) for each shipping group you want fulfilled. See figure. Important: If no shipGroup value is specified, only the item associated with the first shipGroup (1) will be fulfilled.

This figure shows two different shipping groups: shipGroup 1 and shipGroup 2. When initializing the transaction, you must specify each item you want fulfilled based on its shipGroup value.

Ship From	Ship To	Ship Via	Shipping Rate	Handling Rate
CA 94903 US	5400 Bayview Drive Toronto, ON M9C 4N3 CA		3.50	0.00
CA 94903 US	2400 Main Street Pleasantville, CA US	Airborne	3.50	0.00

Does MSR work on custom forms?

Yes. However, after you enable Multiple Shipping Routes in your account, you must also enable MSR on your custom form by adding the Enable Line Item Shipping checkbox to the Items sublist. For steps on adding this checkbox to a custom form, see *Multiple Shipping Routes* in the NetSuite Help Center.

Note: Once MSR is enabled in your account, the Enable Line Item Shipping check box is automatically added to the Items sublist on standard forms.

Multiple Shipping Routes Code Samples for Web Services

The following samples provide a general workflow for Web services developers using MSR. These samples use sales order as the transaction type.

In this workflow you can:

1. Create a sales order and set your items
2. Get the sales order
3. Update the sales order items with MSR shipping information
4. Retrieve the updated shipGroup data and update the shipping cost
5. Update the sales order shipping group
6. Initialize the sales order to create the item fulfillment
7. Search for unfulfilled sales orders that are MSR-enabled

Create a sales order and set your items

This sample shows how to create a sales order and add items. Even if MSR is enabled in your NetSuite account, as this sample shows, you can continue to create sales orders that do not have MSR set for each items. MSR only becomes enabled on a transaction once *isMultiShipTo* is set to *true*.

Note: See the sample [Update the sales order items with MSR shipping information](#) for details on converting a non-MSR transaction to an MSR-enabled transaction.

C#

```
// First, initiate login
NetSuiteService nss = new NetSuiteService();

// provide login details...

// Create a sales order
SalesOrder so = new SalesOrder();

RecordRef customer = new RecordRef();
customer.internalId = "123";
so.entity = customer;

// Set the order status on the sales order
so.orderStatus = SalesOrderOrderStatus._pendingFulfillment;
so.orderStatusSpecified = true;

// Add items
SalesOrderItem [] items = new SalesOrderItem[2];
SalesOrderItemList itemList = new SalesOrderItemList();

SalesOrderItem item1 = new SalesOrderItem();
RecordRef item1Ref = new RecordRef();
item1Ref.internalId = "40";
item1.item = item1Ref;
item1.amountSpecified = true;
item1.amount = 3.0;

SalesOrderItem item2 = new SalesOrderItem();
RecordRef itemRef2 = new RecordRef();
itemRef2.internalId = "45";
item2.item = itemRef2;
item2.quantitySpecified = true;
item2.quantity = 2;

items[0] = item1;
items[1] = item2;

itemList.item = items;
so.itemList = itemList;

WriteResponse ws = nss.add(so);
```

SOAP Request

```
<record xmlns:q1="urn:sales_2008_2.transactions.webservices.netsuite.com" xsi:type="q1:SalesOrder">
  <q1:entity internalId="123" />
  <q1:orderStatus>_pendingFulfillment</q1:orderStatus>
  <q1:itemList>
    <q1:item>
      <q1:item internalId="40" />
      <q1:amount>3</q1:amount>
    </q1:item>
  </q1:itemList>
</record>
```

```

    <q1:item>
      <q1:item internalId="45" />
      <q1:quantity>2</q1:quantity>
    </q1:item>
  </q1:itemList>
</record>

```

SOAP Response

```

<writeResponse>
  <platformCore:status isSuccess="true"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com"/>
  <baseRef internalId="993" type="salesOrder" xsi:type="platformCore:RecordRef"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com"/>
</writeResponse>

```

Get the sales order

After adding items to a sales order, you can then get the sales order and enable MSR on the specific transaction. Note, you could have enabled MSR on the transaction when you originally created the sales order object. To do so, you just needed to set *isMultiShipTo* to true in the first sample.

C#

```

RecordRef rr = new RecordRef();
rr = (RecordRef) ws.baseRef;
String sKey = rr.internalId;
ReadResponse getResp = nss.get(rr);

```

SOAP Request

```

<get xmlns="urn:messages_2008_2.platform.webservices.netsuite.com">
  <baseRef xmlns:q1="urn:core_2008_2.platform.webservices.netsuite.com" xsi:type="q1:RecordRef"
    internalId="993" type="salesOrder" />
</get>

```

SOAP Response

```

<record internalId="993" xsi:type="tranSales:SalesOrder"
  xmlns:tranSales="urn:sales_2008_2.transactions.webservices.netsuite.com">
  <tranSales:createdDate>2008-09-15T17:29:00.000-07:00</tranSales:createdDate>
  <tranSales:entity internalId="123"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
    <platformCore:name>Carrie Davis</platformCore:name>
  </tranSales:entity>
  <tranSales:tranDate>2008-09-15T00:00:00.000-07:00</tranSales:tranDate>
  <tranSales:tranId>183</tranSales:tranId>
  <tranSales:source>Web Services</tranSales:source>
  <tranSales:orderStatus>_pendingFulfillment</tranSales:orderStatus>
  <tranSales:salesRep internalId="111"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
    <platformCore:name>Corporate Sales Team</platformCore:name>
  </tranSales:salesRep>
  <tranSales:partner internalId="129"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
    <platformCore:name>The Bay Times</platformCore:name>
  </tranSales:partner>
  <tranSales:leadSource internalId="100002"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">

```

```

        <platformCore:name>New Registration Promotion</platformCore:name>
    </tranSales:leadSource>
    <tranSales:salesEffectiveDate>2008-09-15T00:00:00.000-07:00</tranSales:salesEffectiveDate>
    <tranSales:excludeCommission>>false</tranSales:excludeCommission>
    <tranSales:promoCode internalId="2"
        xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
        <platformCore:name>New Registration Promotion</platformCore:name>
    </tranSales:promoCode>
    <tranSales:toBePrinted>>false</tranSales:toBePrinted>
    <tranSales:toBeEmailed>>false</tranSales:toBeEmailed>
    <tranSales:email>carrie10101@freeversion.1org</tranSales:email>
    <tranSales:toBeFaxed>>false</tranSales:toBeFaxed>
    <tranSales:transactionBillAddress
        xmlns:platformCommon="urn:common_2008_2.platform.webservices.netsuite.com">
        <platformCommon:billAddr1>550504 Mission Street</platformCommon:billAddr1>
        <platformCommon:billCity>Santa Cruz</platformCommon:billCity>
        <platformCommon:billState>CA</platformCommon:billState>
        <platformCommon:billZip>95060</platformCommon:billZip>
        <platformCommon:billCountry>_unitedStates</platformCommon:billCountry>
    </tranSales:transactionBillAddress>
    <tranSales:billAddress>550504 Mission Street<br>&lt;br>Santa Cruz CA 95060<br>&lt;br>United
        States</tranSales:billAddress>
    <tranSales:transactionShipAddress
        xmlns:platformCommon="urn:common_2008_2.platform.webservices.netsuite.com">
        <platformCommon:shipAddr1>550504 Mission Street</platformCommon:shipAddr1>
        <platformCommon:shipCity>Santa Cruz</platformCommon:shipCity>
        <platformCommon:shipState>CA</platformCommon:shipState>
        <platformCommon:shipZip>95060</platformCommon:shipZip>
        <platformCommon:shipCountry>_unitedStates</platformCommon:shipCountry>
        <platformCommon:shipIsResidential>>true</platformCommon:shipIsResidential>
    </tranSales:transactionShipAddress>
    <tranSales:shipAddress>550504 Mission Street<br>&lt;br>Santa Cruz CA 95060<br>&lt;br>United
        States</tranSales:shipAddress>
    <tranSales:shipDate>2008-0911a4-15T00:00:00.000-07:00</tranSales:shipDate>
    <tranSales:shipMethod internalId="3"
        xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
        <platformCore:name>Special Delivery</platformCore:name>
    </tranSales:shipMethod>
    <tranSales:shippingCost>5.0</tranSales:shippingCost>
    <tranSales:isMultiShipTo>>false</tranSales:isMultiShipTo>
    <tranSales:shippingTaxCode internalId="-7"
        xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
        <platformCore:name>-Not Taxable-</platformCore:name>
    </tranSales:shippingTaxCode>
    <tranSales:handlingTaxCode internalId="-7"
        xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
        <platformCore:name>-Not Taxable-</platformCore:name>
    </tranSales:handlingTaxCode>
    <tranSales:handlingCost>6.0</tranSales:handlingCost>
    <tranSales:shipComplete>>false</tranSales:shipComplete>
    <tranSales:ccApproved>>false</tranSales:ccApproved>
    <tranSales:altSalesTotal>0.0</tranSales:altSalesTotal>
    <tranSales:subTotal>24.9</tranSales:subTotal>
    <tranSales:altShippingCost>5.0</tranSales:altShippingCost>
    <tranSales:altHandlingCost>6.0</tranSales:altHandlingCost>
    
```

```

<tranSales:total>35.9</tranSales:total>
<tranSales:subsidiary internalId="1"
  xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
  <platformCore:name>Parent Company</platformCore:name>
</tranSales:subsidiary>
<tranSales:lastModifiedDate>2008-09-15T17:29:00.000-07:00</tranSales:lastModifiedDate>
<tranSales:status>Pending Fulfillment</tranSales:status>
<tranSales:itemList>
  <tranSales:item>
    <tranSales:item internalId="40"
      xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
      <platformCore:name>Accessories : Crusher Game Pad</platformCore:name>
    </tranSales:item>
    <tranSales:quantity>1.0</tranSales:quantity>
    <tranSales:description>Crusher Game Pad</tranSales:description>
    <tranSales:price internalId="1"
      xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
      <platformCore:name>Base Price</platformCore:name>
    </tranSales:price>
    <tranSales:rate>12.95</tranSales:rate>
    <tranSales:amount>3.0</tranSales:amount>
    <tranSales:commitInventory>_availableQty</tranSales:commitInventory>
    <tranSales:isClosed>>false</tranSales:isClosed>
    <tranSales:fromJob>>false</tranSales:fromJob>
    <tranSales:isEstimate>>false</tranSales:isEstimate>
    <tranSales:line>1</tranSales:line>
    <tranSales:quantityBackOrdered>0.0</tranSales:quantityBackOrdered>
    <tranSales:quantityBilled>0.0</tranSales:quantityBilled>
    <tranSales:quantityCommitted>1.0</tranSales:quantityCommitted>
    <tranSales:quantityFulfilled>0.0</tranSales:quantityFulfilled>
    <tranSales:taxCode internalId="-7"
      xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
      <platformCore:name>-Not Taxable-</platformCore:name>
    </tranSales:taxCode>
  </tranSales:item>
  <tranSales:item>
    <tranSales:item internalId="45"
      xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
      <platformCore:name>Accessories : Cable - Cat 5, 10 ft</platformCore:name>
    </tranSales:item>
    <tranSales:quantity>2.0</tranSales:quantity>
    <tranSales:description>Cat 5 Patch Cable 10 ft</tranSales:description>
    <tranSales:price internalId="1"
      xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
      <platformCore:name>Base Price</platformCore:name>
    </tranSales:price>
    <tranSales:rate>10.95</tranSales:rate>
    <tranSales:amount>21.9</tranSales:amount>
    <tranSales:commitInventory>_availableQty</tranSales:commitInventory>
    <tranSales:isClosed>>false</tranSales:isClosed>
    <tranSales:fromJob>>false</tranSales:fromJob>
    <tranSales:isEstimate>>false</tranSales:isEstimate>
    <tranSales:line>2</tranSales:line>
    <tranSales:quantityBackOrdered>0.0</tranSales:quantityBackOrdered>
    <tranSales:quantityBilled>0.0</tranSales:quantityBilled>
  </tranSales:item>
</tranSales:itemList>

```

```

<tranSales:quantityCommitted>2.0</tranSales:quantityCommitted>
<tranSales:quantityFulfilled>0.0</tranSales:quantityFulfilled>
<tranSales:taxCode internalId="-7"
  xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
  <platformCore:name>-Not Taxable-</platformCore:name>
</tranSales:taxCode>
</tranSales:item>
</tranSales:itemList>
<tranSales:customFieldList
  xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
<platformCore:customField internalId="custbody1"
  xsi:type="platformCore:StringCustomFieldRef">
  <platformCore:value>800-555-9177</platformCore:value>
</platformCore:customField>
</tranSales:customFieldList>
</record>

```

Update the sales order items with MSR shipping information

This sample shows how to convert a sales order that does not yet have MSR enabled to a sales order that does. To do this you must set *isMultiShipTo* to *true* and then specify the *shipAddress* and *shipMethod* for each item.

C#

```

// Keep a record of the sales order from previous query operation
SalesOrder soCreated = (SalesOrder) getResp.record;

SalesOrder soUpdate = new SalesOrder();
soUpdate.internalId = soCreated.internalId;

// Enable line level shipping. This is equivalent to selecting
// the Enable Line Level Shipping checkbox on the Items tab.
soUpdate.isMultiShipToSpecified = true;
soUpdate.isMultiShipTo = true;

// set shipMethods and shipAddresses
SalesOrderItem [] itemsUpdate = new SalesOrderItem[soCreated.itemList.item.Length];
SalesOrderItem item1Update = new SalesOrderItem();
item1Update.lineSpecified=true;
item1Update.line = soCreated.itemList.item[0].line;
RecordRef shipAddr1 = new RecordRef();
shipAddr1.internalId = "93";
item1Update.shipAddress = shipAddr1;

RecordRef shipMeth1 = new RecordRef();
shipMeth1.internalId = "2";
item1Update.shipMethod = shipMeth1;

SalesOrderItem item2Update = new SalesOrderItem();
item2Update.lineSpecified=true;
item2Update.line = soCreated.itemList.item[1].line;
RecordRef shipAddr2 = new RecordRef();
shipAddr2.internalId = "244715";
item2Update.shipAddress = shipAddr2;

```

```

RecordRef shipMeth2 = new RecordRef();
shipMeth2.internalId = "92";
item2Update.shipMethod = shipMeth2;

itemsUpdate[0] = item1Update;
itemsUpdate[1] = item2Update;
SalesOrderItemList itemListUpdate = new SalesOrderItemList();
itemListUpdate.item = itemsUpdate;
soUpdate.itemList = itemListUpdate;
nss.update(soUpdate);

```

SOAP Request

```

<update xmlns="urn:messages_2008_2.platform.webservices.netsuite.com">
  <record xmlns:q1="urn:sales_2008_2.transactions.webservices.netsuite.com"
    xsi:type="q1:SalesOrder" internalId="997">
    <q1:isMultiShipTo>true</q1:isMultiShipTo>
    <q1:itemList>
      <q1:item>
        <q1:line>1</q1:line>
        <q1:shipAddress internalId="93" />
        <q1:shipMethod internalId="2" />
      </q1:item>
      <q1:item>
        <q1:line>2</q1:line>
        <q1:shipAddress internalId="244715" />
        <q1:shipMethod internalId="92" />
      </q1:item>
    </q1:itemList>
  </record>
</update>

```

SOAP Response

```

<writeResponse>
  <platformCore:status isSuccess="true"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com"/>
    <baseRef internalId="997" type="salesOrder" xsi:type="platformCore:RecordRef"
      xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com"/>
</writeResponse>

```

Retrieve the updated shipGroup data and update the shipping cost

If you choose, you can get the sales order and then specify which item you want to update based on the item's shipGroup value.

C#

```
// Retrieve the updated shipGroup data and then update
// the shipping cost
ReadResponse getRespUpdated = nss.get(rr);
```

SOAP Request

```
<get xmlns="urn:messages_2008_2.platform.webservices.netsuite.com">
  <baseRef xmlns:q1="urn:core_2008_2.platform.webservices.netsuite.com" xsi:type="q1:RecordRef"
    internalId="997" type="salesOrder" />
</get>
```

SOAP Response

```
<record internalId="997" xsi:type="tranSales:SalesOrder"
  xmlns:tranSales="urn:sales_2008_2.transactions.webservices.netsuite.com">
  <tranSales:createdDate>2008-09-16T10:51:00.000-07:00</tranSales:createdDate>
  <tranSales:entity internalId="123"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
    <platformCore:name>Carrie Davis</platformCore:name>
  </tranSales:entity>
  <tranSales:tranDate>2008-09-16T00:00:00.000-07:00</tranSales:tranDate>
  <tranSales:tranId>186</tranSales:tranId>
  <tranSales:source>Web Services</tranSales:source>
  <tranSales:orderStatus>_pendingFulfillment</tranSales:orderStatus>
  <tranSales:salesRep internalId="111"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
    <platformCore:name>Corporate Sales Team</platformCore:name>
  </tranSales:salesRep>
  <tranSales:partner internalId="129"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
    <platformCore:name>The Bay Times</platformCore:name>
  </tranSales:partner>
  <tranSales:leadSource internalId="100002"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
    <platformCore:name>New Registration Promotion</platformCore:name>
  </tranSales:leadSource>
  <tranSales:salesEffectiveDate>2008-09-16T00:00:00.000-07:00</tranSales:salesEffectiveDate>
  <tranSales:excludeCommission>>false</tranSales:excludeCommission>
  <tranSales:promoCode internalId="2"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
    <platformCore:name>New Registration Promotion</platformCore:name>
  </tranSales:promoCode>
  <tranSales:toBePrinted>>false</tranSales:toBePrinted>
  <tranSales:toBeEmailed>>false</tranSales:toBeEmailed>
  <tranSales:email>carrie10101@freeversion.1org</tranSales:email>
  <tranSales:toBeFaxed>>false</tranSales:toBeFaxed>
  <tranSales:transactionBillAddress
    xmlns:platformCommon="urn:common_2008_2.platform.webservices.netsuite.com">
    <platformCommon:billAddr1>550504 Mission Street</platformCommon:billAddr1>
    <platformCommon:billCity>Santa Cruz</platformCommon:billCity>
    <platformCommon:billState>CA</platformCommon:billState>
```

```

        <platformCommon:billZip>95060</platformCommon:billZip>
        <platformCommon:billCountry>_unitedStates</platformCommon:billCountry>
    </tranSales:transactionBillAddress>
    <tranSales:billAddress>550504 Mission Street<br>&lt;br>Santa Cruz CA 95060&lt;br>&lt;br>United
        States</tranSales:billAddress>
    <tranSales:transactionShipAddress
        xmlns:platformCommon="urn:common_2008_2.platform.webservices.netsuite.com">
        <platformCommon:shipIsResidential>>false</platformCommon:shipIsResidential>
    </tranSales:transactionShipAddress>
    <tranSales:shipDate>2008-09-16T00:00:00.000-07:00</tranSales:shipDate>
    <tranSales:isMultiShipTo>>true</tranSales:isMultiShipTo>
    <tranSales:shipComplete>>false</tranSales:shipComplete>
    <tranSales:ccApproved>>false</tranSales:ccApproved>
    <tranSales:altSalesTotal>0.0</tranSales:altSalesTotal>
    <tranSales:subTotal>24.9</tranSales:subTotal>
    <tranSales:altShippingCost>11.87</tranSales:altShippingCost>
    <tranSales:altHandlingCost>4.0</tranSales:altHandlingCost>
    <tranSales:total>40.77</tranSales:total>
    <tranSales:subsidiary internalId="1"
        xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
        <platformCore:name>Parent Company</platformCore:name>
    </tranSales:subsidiary>
    <tranSales:lastModifiedDate>2008-09-16T10:51:00.000-07:00</tranSales:lastModifiedDate>
    <tranSales:status>Pending Fulfillment</tranSales:status>
    <tranSales:itemList>
    <tranSales:item>
    <tranSales:item internalId="40"
        xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
        <platformCore:name>Accessories : Crusher Game Pad</platformCore:name>
    </tranSales:item>
    <tranSales:quantity>1.0</tranSales:quantity>
    <tranSales:description>Crusher Game Pad</tranSales:description>
    <tranSales:price internalId="1"
        xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
        <platformCore:name>Base Price</platformCore:name>
    </tranSales:price>
    <tranSales:rate>12.95</tranSales:rate>
    <tranSales:amount>3.0</tranSales:amount>
    <tranSales:commitInventory>_availableQty</tranSales:commitInventory>
    <tranSales:isClosed>>false</tranSales:isClosed>
    <tranSales:fromJob>>false</tranSales:fromJob>
    <tranSales:isEstimate>>false</tranSales:isEstimate>
    <tranSales:line>1</tranSales:line>
    <tranSales:quantityBackOrdered>0.0</tranSales:quantityBackOrdered>
    <tranSales:quantityBilled>0.0</tranSales:quantityBilled>
    <tranSales:quantityCommitted>1.0</tranSales:quantityCommitted>
    <tranSales:quantityFulfilled>0.0</tranSales:quantityFulfilled>
    <tranSales:taxCode internalId="-7"
        xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
        <platformCore:name>-Not Taxable-</platformCore:name>
    </tranSales:taxCode>
    <tranSales:shipAddress internalId="93"
        xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
        <platformCore:name>Billing Address</platformCore:name>
    </tranSales:shipAddress>
    
```

```

        <tranSales:shipMethod internalId="2"
            xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
            <platformCore:name>Pick-up</platformCore:name>
        </tranSales:shipMethod>
    </tranSales:item>
    <tranSales:item>
        <tranSales:item internalId="45"
            xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
            <platformCore:name>Accessories : Cable - Cat 5, 10 ft</platformCore:name>
        </tranSales:item>
        <tranSales:quantity>2.0</tranSales:quantity>
        <tranSales:description>Cat 5 Patch Cable 10 ft</tranSales:description>
        <tranSales:price internalId="1"
            xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
            <platformCore:name>Base Price</platformCore:name>
        </tranSales:price>
        <tranSales:rate>10.95</tranSales:rate>
        <tranSales:amount>21.9</tranSales:amount>
        <tranSales:commitInventory>_availableQty</tranSales:commitInventory>
        <tranSales:isClosed>>false</tranSales:isClosed>
        <tranSales:fromJob>>false</tranSales:fromJob>
        <tranSales:isEstimate>>false</tranSales:isEstimate>
        <tranSales:line>2</tranSales:line>
        <tranSales:quantityBackOrdered>0.0</tranSales:quantityBackOrdered>
        <tranSales:quantityBilled>0.0</tranSales:quantityBilled>
        <tranSales:quantityCommitted>2.0</tranSales:quantityCommitted>
        <tranSales:quantityFulfilled>0.0</tranSales:quantityFulfilled>
        <tranSales:taxCode internalId="-7"
            xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
            <platformCore:name>-Not Taxable-</platformCore:name>
        </tranSales:taxCode>
        <tranSales:shipAddress internalId="244715"
            xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
            <platformCore:name>san mateo shipping</platformCore:name>
        </tranSales:shipAddress>
        <tranSales:shipMethod internalId="92"
            xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
            <platformCore:name>UPS Ground</platformCore:name>
        </tranSales:shipMethod>
    </tranSales:item>
</tranSales:itemList>
<tranSales:shipGroupList>
    <tranSales:shipGroup>
        <tranSales:id>1</tranSales:id>
        <tranSales:isFulfilled>>false</tranSales:isFulfilled>
        <tranSales:weight>1.0</tranSales:weight>
        <tranSales:sourceAddressRef internalId="DEFAULT"
            xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com"/>
        <tranSales:sourceAddress>1500 3rd St San Mateo, CA 94403 US</tranSales:sourceAddress>
        <tranSales:destinationAddressRef internalId="93"
            xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com"/>
        <tranSales:destinationAddress>CO United States</tranSales:destinationAddress>
        <tranSales:shippingMethodRef internalId="2"
            xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com"/>
        <tranSales:shippingMethod>Pick-up</tranSales:shippingMethod>
    </tranSales:shipGroup>
</tranSales:shipGroupList>

```

```

        <tranSales:handlingRate>0.0</tranSales:handlingRate>
        <tranSales:shippingRate>5.0</tranSales:shippingRate>
    </tranSales:shipGroup>
</tranSales:shipGroupList>
<tranSales:customFieldList>
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com"/>
    <platformCore:customField internalId="custbody1"
        xsi:type="platformCore:StringCustomFieldRef">
        <platformCore:value>800-555-9177</platformCore:value>
    </platformCore:customField>
</tranSales:customFieldList>
</record>

```

Update the sales order shipping group

By specifying a shipping group, you can then update the data associated with that shipping group. This data includes item shipping and handling rates.

C#

```

// Update the sales order shipping groups, for example, the shipping rate

// Keep a record of the SO updated
SalesOrder soUpdated = (SalesOrder) getRespUpdated.record;

SalesOrder soShipGroupUpdate = new SalesOrder();
soShipGroupUpdate.internalId = soUpdated.internalId;

TransactionShipGroup shipGroup1 = new TransactionShipGroup();
shipGroup1.idSpecified = true;
shipGroup1.id = soUpdated.shipGroupList.shipGroup[0].id;
shipGroup1.shippingRateSpecified = true;
shipGroup1.shippingRate = 10.0;
TransactionShipGroup [] shipGroups = new TransactionShipGroup[1];
shipGroups[0] = shipGroup1;

SalesOrderShipGroupList shipGroupList = new SalesOrderShipGroupList();
shipGroupList.shipGroup = shipGroups;

```

```
soShipGroupUpdate.shipGroupList = shipGroupList;
nss.update(soShipGroupUpdate);
```

SOAP Request

```
<update xmlns="urn:messages_2008_2.platform.webservices.netsuite.com">
  <record xmlns:q1="urn:sales_2008_2.transactions.webservices.netsuite.com" xsi:type="q1:SalesOrder"
    internalId="997">
    <q1:shipGroupList>
      <q1:shipGroup>
        <q1:id>1</q1:id>
        <q1:shippingRate>10</q1:shippingRate>
      </q1:shipGroup>
    </q1:shipGroupList>
  </record>
</update>
```

SOAP Response

```
<writeResponse>
  <platformCore:status isSuccess="true"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com"/>
  <baseRef internalId="997" type="salesOrder" xsi:type="platformCore:RecordRef"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com"/>
```

Initialize the sales order to create the item fulfillment

Next, you can run `initialize()` to obtain the item fulfillment record.

Note: The `initialize` operation includes an optional argument called *AuxReference*. In *AuxReference* users can set the type to *shippingGroup* by using the enum object `InitializeAuxRefType`.

When working with MSR-enabled transactions, you must specify a value for *shippingGroup*. If you do not specify a value, the value 1 (for the first shipping group) is defaulted. This means that only the items belonging to the first shipping group will be fulfilled when the sales order is initialized.

C#

```
for (int j=0; j<soUpdated.shipGroupList.shipGroup.Length; j++)
{
  InitializeRecord ir = new InitializeRecord();
  ir.type = InitializeType.itemFulfillment;
  InitializeRef iref = new InitializeRef();
  iref.typeSpecified=true;
  iref.type=InitializeRefType.salesOrder;
  iref.internalId = rr.internalId;
  ir.reference = iref;
  InitializeAuxRef iar = new InitializeAuxRef();
  iar.typeSpecified=true;
  iar.type = InitializeAuxRefType.shippingGroup;
  iar.internalId = soUpdated.shipGroupList.shipGroup[j].id.ToString();
  ir.auxReference = iar;
  ReadResponse getInitResp = nss.initialize(ir);
  if (getInitResp.status.isSuccess)
```

```

{
// Keep a record of the original item fulfillment
ItemFulfillment ifrec = (ItemFulfillment) getInitResp.record;

//SalesOrderShipGroupList sgListForIF = ifrec.shipGroupList;
//TransactionShipGroup [] shipGroupsForIF = sgListForIF.shipGroup;

ItemFulfillment recToFulfill = new ItemFulfillment();
// Set createdFrom field
recToFulfill.createdFrom = ifrec.createdFrom;
// Set createdFromShipGroup field for multiple shipping routes enabled orders
recToFulfill.createdFromShipGroupSpecified=true;
recToFulfill.createdFromShipGroup = soUpdated.shipGroupList.shipGroup[j].id;

ItemFulfillmentItemList ifitemlist = ifrec.itemList;
ItemFulfillmentItem [] ifitems = new ItemFulfillmentItem[ifitemlist.item.Length];
RecordRef locRef = new RecordRef();
locRef.internalId = "1";
    for (int i=0; i<ifitemlist.item.Length; i++)
    {
        ItemFulfillmentItem fffItem = new ItemFulfillmentItem();
        fffItem.item = ifitemlist.item[i].item;
        fffItem.orderLineSpecified=true;
        fffItem.orderLine = ifitemlist.item[i].orderLine;
        fffItem.location = locRef;
        ifitems[i] = fffItem;
    }
    ItemFulfillmentItemList ifitemlistToFulfill = new ItemFulfillmentItemList();
    ifitemlistToFulfill.item = ifitems;
    recToFulfill.itemList = ifitemlistToFulfill;
    nss.add(recToFulfill);
}
}
    
```

SOAP Request

```

<initialize xmlns="urn:messages_2008_2.platform.webservices.netsuite.com">
  <initializeRecord>
    <type xmlns="urn:core_2008_2.platform.webservices.netsuite.com">itemFulfillment</type>
    <reference type="salesOrder" internalId="997"
      xmlns="urn:core_2008_2.platform.webservices.netsuite.com" />
    <auxReference type="shippingGroup" internalId="1"
      xmlns="urn:core_2008_2.platform.webservices.netsuite.com" />
  </initializeRecord>
</initialize>
    
```

SOAP Response

```

<initializeResponse xmlns="urn:messages_2008_2.platform.webservices.netsuite.com">
  <readResponse>
    <platformCore:status isSuccess="true"
      xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com"/>
    <record xsi:type="tranSales:ItemFulfillment"
      xmlns:tranSales="urn:sales_2008_2.transactions.webservices.netsuite.com">
      <tranSales:createdDate>2008-09-16T10:51:00.000-07:00</tranSales:createdDate>
      <tranSales:lastModifiedDate>2008-09-16T10:51:00.000-07:00</tranSales:lastModifiedDate>
      <tranSales:postingPeriod internalId="132"
        xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
    
```

```

    <platformCore:name>Sep 2008</platformCore:name>
  </tranSales:postingPeriod>
  <tranSales:entity internalId="123"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
    <platformCore:name>Carrie Davis</platformCore:name>
  </tranSales:entity>
  <tranSales:createdFrom internalId="997"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
    <platformCore:name>Sales Order #186</platformCore:name>
  </tranSales:createdFrom>
  <tranSales:transactionShipAddress
    xmlns:platformCommon="urn:common_2008_2.platform.webservices.netsuite.com">
    <platformCommon:shipAttention>Billing Address</platformCommon:shipAttention>
    <platformCommon:shipAddressee>Carrie Davis</platformCommon:shipAddressee>
    <platformCommon:shipPhone>800-555-9177</platformCommon:shipPhone>
    <platformCommon:shipState>CO</platformCommon:shipState>
    <platformCommon:shipCountry>_unitedStates</platformCommon:shipCountry>
    <platformCommon:shipIsResidential>>false</platformCommon:shipIsResidential>
  </tranSales:transactionShipAddress>
  <tranSales:shipAddressList internalId="93"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
    <platformCore:name>Billing Address</platformCore:name>
  </tranSales:shipAddressList>
  <tranSales:shipAddress>Carrie DavisCO US</tranSales:shipAddress>
  <tranSales:tranDate>2008-09-16T00:00:00.000-07:00</tranSales:tranDate>
  <tranSales:tranId>39</tranSales:tranId>
  <tranSales:shipMethod internalId="2"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
    <platformCore:name>Pick-up</platformCore:name>
  </tranSales:shipMethod>
  <tranSales:shippingCost>10.0</tranSales:shippingCost>
  <tranSales:handlingCost>0.0</tranSales:handlingCost>
  <tranSales:itemList>
  <tranSales:item>
    <tranSales:itemReceive>>true</tranSales:itemReceive>
    <tranSales:itemName>Crusher Game Pad </tranSales:itemName>
    <tranSales:description>Crusher Game Pad</tranSales:description>
    <tranSales:quantity>1.0</tranSales:quantity>
    <tranSales:item internalId="40"
      xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
      <platformCore:name>Crusher Game Pad</platformCore:name>
    </tranSales:item>
    <tranSales:orderLine>1</tranSales:orderLine>
    <tranSales:quantityRemaining>1.0</tranSales:quantityRemaining>
  </tranSales:item>
</tranSales:itemList>
</record>
</readResponse>
</initializeResponse>

```

SOAP Request (Item Fulfillment)

```

<add xmlns="urn:messages_2008_2.platform.webservices.netsuite.com">
  <record xmlns:q1="urn:sales_2008_2.transactions.webservices.netsuite.com"
    xsi:type="q1:ItemFulfillment">
    <q1:createdFrom internalId="997">

```

```

        <name xmlns="urn:core_2008_2.platform.webservices.netsuite.com">Sales Order #186
        </name>
    </q1:createdFrom>
    <q1:itemList>
    <q1:item>
        <q1:location internalId="1" />
        <q1:item internalId="40">
            <name xmlns="urn:core_2008_2.platform.webservices.netsuite.com">Crusher Game
            Pad</name>
        </q1:item>
        <q1:orderLine>1</q1:orderLine>
    </q1:item>
    </q1:itemList>
</record>
</add>

```

SOAP Response (Item Fulfillment)

```

<writeResponse>
  <platformCore:status isSuccess="true"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com"/>
  <baseRef internalId="998" type="itemFulfillment" xsi:type="platformCore:RecordRef"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com"/>
</writeResponse>

```

Search for unfulfilled sales orders that are MSR-enabled

This sample shows how to search for unfulfilled sales orders that are MSR-enabled. For details on how to specify additional search criteria, see the section called “Advanced Searches in Web Services” in the *SuiteTalk (Web Services) Platform Guide*.

C#

```

// Define search criteria and return columns
TransactionSearchAdvanced tsa4 = new TransactionSearchAdvanced();

// Set search preference to return search columns
SearchPreferences sp = new SearchPreferences();
sp.returnSearchColumns = true;
nss.searchPreferences = sp;

// Instantiate SearchRow object
TransactionSearchRow tsr = new TransactionSearchRow();
TransactionSearchRowBasic tsrb = new TransactionSearchRowBasic();

// return internId
SearchColumnSelectField [] orderIdCols = new SearchColumnSelectField[1];
SearchColumnSelectField orderIdCol = new SearchColumnSelectField();
orderIdCol.customLabel = "Sales Order ID"; // Define a custom label
orderIdCols[0] = orderIdCol;
tsrb.internalId = orderIdCols;

// return item
SearchColumnSelectField [] itemIdCols = new SearchColumnSelectField[1];
SearchColumnSelectField itemId = new SearchColumnSelectField();
itemIdCols[0] = itemId;
tsrb.item = itemIdCols;

```

```
// return item fulfillment status
SearchColumnBooleanField [] isFulfilledCols = new SearchColumnBooleanField[1];
SearchColumnBooleanField isFulfilledCol = new SearchColumnBooleanField();
isFulfilledCol.customLabel = "Order Fulfilled";
isFulfilledCols[0] = isFulfilledCol;
tsrb.shipRecvStatusLine = isFulfilledCols;

// return tranDate
SearchColumnDateField [] tranDateCols = new SearchColumnDateField[1];
SearchColumnDateField tranDateCol = new SearchColumnDateField();
tranDateCols[0] = tranDateCol;
tsrb.tranDate = tranDateCols;

// return tranId
SearchColumnStringField [] tranIdCols = new SearchColumnStringField[1];
SearchColumnStringField tranIdCol = new SearchColumnStringField();
tranIdCols[0] = tranIdCol;
tsrb.tranId = tranIdCols;

tsr.basic = tsrb;

// Define search criteria
TransactionSearch ts = new TransactionSearch();
TransactionSearchBasic tsb = new TransactionSearchBasic();

// on SO only
SearchEnumMultiSelectField semsfTranType = new SearchEnumMultiSelectField();
semsfTranType.operatorSpecified = true;
semsfTranType.@operator = SearchEnumMultiSelectFieldOperator.anyOf;
String [] tranTypes = new String[1];
String tranType = "_salesOrder";
tranTypes[0] = tranType;
semsfTranType.searchValue = tranTypes;
tsb.type = semsfTranType;

// tranId contains 183
SearchStringField sfTranId = new SearchStringField();
sfTranId.searchValue = "183";
sfTranId.@operator = SearchStringFieldOperator.contains;
sfTranId.operatorSpecified = true;
tsb.tranId=sfTranId;

// SO item unfulfilled
SearchBooleanField sbfTranStatus = new SearchBooleanField();
sbfTranStatus.searchValue = false;
sbfTranStatus.searchValueSpecified = true;
tsb.shipRecvStatusLine = sbfTranStatus;

ts.basic = tsb;
tsa4.criteria = ts;

tsa4.columns = tsr;
nss.search(tsa4);
```

SOAP Request

```

<search xmlns="urn:messages_2008_2.platform.webservices.netsuite.com">
  <searchRecord xmlns:q1="urn:sales_2008_2.transactions.webservices.netsuite.com"
    xsi:type="q1:TransactionSearchAdvanced">
    <q1:criteria>
      <q1:basic>
        <shipRecvStatusLine xmlns="urn:common_2008_2.platform.webservices.netsuite.com">
          <searchValue xmlns="urn:core_2008_2.platform.webservices.netsuite.com">false
            </searchValue>
          </shipRecvStatusLine>
          <tranId operator="contains" xmlns="urn:common_2008_2.platform.webservices.netsuite.com">
            <searchValue xmlns="urn:core_2008_2.platform.webservices.netsuite.com">183
              </searchValue>
            </tranId>
          <type operator="anyOf" xmlns="urn:common_2008_2.platform.webservices.netsuite.com">
            <searchValue xmlns="urn:core_2008_2.platform.webservices.netsuite.com">_salesOrder
              </searchValue>
            </type>
          </q1:basic>
        </q1:criteria>
        <q1:columns>
          <q1:basic>
            <internalId xmlns="urn:common_2008_2.platform.webservices.netsuite.com">
              <customLabel xmlns="urn:core_2008_2.platform.webservices.netsuite.com">Sales Order ID
                </customLabel>
              </internalId>
            <item xmlns="urn:common_2008_2.platform.webservices.netsuite.com" />
            <shipRecvStatusLine xmlns="urn:common_2008_2.platform.webservices.netsuite.com">
              <customLabel xmlns="urn:core_2008_2.platform.webservices.netsuite.com">Order Item
                Fulfilled</customLabel>
              </shipRecvStatusLine>
            <tranDate xmlns="urn:common_2008_2.platform.webservices.netsuite.com" />
            <tranId xmlns="urn:common_2008_2.platform.webservices.netsuite.com" />
          </q1:basic>
        </q1:columns>
      </searchRecord>
    </search>
  </soap:Body>

```

SOAP Response

```

<searchResponse xmlns="urn:messages_2008_2.platform.webservices.netsuite.com">
  <platformCore:searchResult
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
    <platformCore:status isSuccess="true"/>
    <platformCore:totalRecords>1</platformCore:totalRecords>
    <platformCore:pageSize>10</platformCore:pageSize>
    <platformCore:totalPages>1</platformCore:totalPages>
    <platformCore:pageIndex>1</platformCore:pageIndex>
    <platformCore:searchId>WEBSERVICES_MSTRWLF_111420081646577959501232561_cb6492913ad
      1</platformCore:searchId>
    <platformCore:searchRowList>
      <platformCore:searchRow xsi:type="tranSales:TransactionSearchRow"
        xmlns:tranSales="urn:sales_2008_2.transactions.webservices.netsuite.com">
        <tranSales:basic
          xmlns:platformCommon="urn:common_2008_2.platform.webservices.netsuite.com">

```

```

<platformCommon:internalId>
  <platformCore:searchValue internalId="987"/>
  <platformCore:customLabel>Sales Order ID</platformCore:customLabel>
</platformCommon:internalId>
<platformCommon:item>
  <platformCore:searchValue internalId="40"/>
</platformCommon:item>
<platformCommon:shipRecvStatusLine>
  <platformCore:searchValue>>false</platformCore:searchValue>
  <platformCore:customLabel>Order Item Fulfilled</platformCore:customLabel>
</platformCommon:shipRecvStatusLine>
<platformCommon:tranDate>
  <platformCore:searchValue>2008-11-14T00:00:00.000-08:00</platformCore:searchValue>
</platformCommon:tranDate>
<platformCommon:tranId>
  <platformCore:searchValue>183</platformCore:searchValue>
</platformCommon:tranId>
</tranSales:basic>
</platformCore:searchRow>
</platformCore:searchRowList>
</platformCore:searchResult>
</searchResponse>
    
```

Chapter 6 Items

Review the following for general information about SuiteTalk items:

- [Usage Notes for Item Record Types](#)
- [Working with Matrix Items in Web Services](#)
- [Shared Field Definitions for Items](#)

The following item record types are supported in SuiteTalk:

- [Item Search](#)
- [Assembly Item \(BOM Item\)](#)
- [Download Item](#)
- [Inventory Item](#)
- [Kit/Package Item](#)
- [Gift Certificate Item](#)
- [Lot Numbered Assembly Item](#)
- [Lot Numbered Inventory Item](#)
- [Serialized Inventory Item](#)
- [Description Item](#)
- [Discount Item](#)
- [Markup Item](#)
- [Payment Item](#)
- [Subtotal Item](#)
- [Noninventory Purchase Item](#)
- [Noninventory Resale Item](#)
- [Other Charge Resale Item](#)
- [Other Charge Purchase Item](#)
- [Serialized Assembly Item](#)
- [Service Resale Item](#)
- [Service Purchase Item](#)
- [Service Sale Item](#)

- [Other Charge Sale Item](#)

Usage Notes for Item Record Types

Note the following when working with item record types:

- Items are the goods and services you sell to customers, and the parts and raw materials you purchase from vendors. They can also include line items on sales and purchase forms, such as discounts and miscellaneous charges.
- Depending on the features enabled for the account specified during the login call, some item types may not be available.
- Setting the item base price will auto-calculate other pricing levels. For more information on setting item prices, see [Item Pricing](#) in the NetSuite Help Center.
- To make an item featured, you must set the item category to the Home tab in the NetSuite UI.
- The getDeleted operation is NOT supported on item record types.
- All item records are defined in the `listAcct (accounting)` XSD.

Working with Matrix Items in Web Services

Before working with matrix items in Web services, you must first enable the Matrix Item feature in your NetSuite account. To enable this feature go to Setup > Company > Enable Features, on the Items & Inventory subtab, select the Matrix Items check box.

This section covers the following topics:

- [Item Types that Support Matrix Options](#)
- [Creating a Parent Matrix Item](#)
- [Creating a Child Matrix Item](#)
- [Specifying Matrix Options](#)
- [Matrix Dimensions](#)
- [Updating Child and Parent Matrix Items](#)
- [Deleting Child and Parent Matrix Items](#)
- [Matrix Item Code Sample](#)

Note: For general information about the matrix items feature, see the topic *Matrix Items Overview* in the NetSuite Help Center.

Item Types that Support Matrix Options

You can use Web services to create matrix items for the following record types:

- Inventory Item
- Lot Numbered Inventory Item
- Serialized Inventory Item
- Non-Inventory Item for Sale
- Non-Inventory Item for Resale
- Non-Inventory Item for Purchase
- Service Item for Sale
- Service Item for Resale
- Service Item for Purchase
- Other Charge Item for Sale
- Other Charge Item for Resale
- Other Charge Item for Purchase

Note that item types that supports matrix options includes the following elements:

- A *matrixType* enum field which can take the value of either *_parent* or *_child* to indicate if the submitted record is a parent or child matrix item. See [Creating a Parent Matrix Item](#).
- A *parent* (Subitem of) field that points to a matrix parent item and populates the MatrixOption list. See [Creating a Child Matrix Item](#).
- A *MatrixOptionList* element (similar to the Matrix tab on the UI). This element is an array of customFieldList of type select to populate matrix options on the child record. Users can specify only one value per matrix option field.

Creating a Parent Matrix Item

Users must set the *matrixType* field if they want to specify an item as either a parent or child matrix item.

Note: This is a deviation from the UI behavior. In the UI you cannot create a parent matrix item without selecting the child options, however, in Web services this is possible.

Creating a Child Matrix Item

Assuming that users have already setup matrix item options in their account, they can proceed with creating child matrix items. As opposed to the UI where selecting matrix options on the parent will result in creating the matrix children, in Web services the user has to add each child matrix item separately.

The user must set the RecordRef of the *parent* (Subitem of) field and specify one or more matrix options for the matrix child item record. Note that the parent item must have the *isMatrixParent* field set to TRUE, otherwise the following error is thrown:

Error code = INVALID_MATRIX_PARENT

Error message = Item {1} is not a parent matrix item.

Users can also set the *externalId* field to submit parent matrix item data and associate the child matrix items with the parent in a single *addList* request in Web services.

Important: A maximum of 2000 children are permitted for a parent item.

Specifying Matrix Options

A call to `getCustomization("itemOptionCustomField")` can be used to return valid matrix option fields and values. A *get* on the parent matrix item returns only the matrix options that have been used. Not all possible options are returned.

Matrix Dimensions

Once the first child is created, the dimensions for the matrix item are set and can not be changed. For example, if users create a large, blue dress as the first child matrix item, they cannot subsequently create a dress that has any property other than size and color. A small, red, cotton dress will not be a valid option.

If users try to add a new child matrix with options that already exists for the parent, they will receive an error.

Updating Child and Parent Matrix Items

Users cannot update child matrix items through the parent record. Users must submit updates for child items individually. Also note that the following error is thrown if users attempt to update an existing item to make it a parent.

Error Code = USER_ERROR

Error Message = You can not change an existing item to make it a parent matrix item {1}.

Note: The values that are returned in `MatrixOptionList` are read-only once the child matrix item is added.

Deleting Child and Parent Matrix Items

In Web services users cannot delete a parent item if it has children. They must delete the child items individually before deleting the parent.

Matrix Item Code Sample

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Body>
      <addList xmlns="urn:messaging_2008_2.platform.webservices.netsuite.com">
```

```

    <record externalId="parentSweater" xsi:type="ns1:InventoryItem"
xmlns:ns1="urn:accounting_2008_2.lists.webservices.netsuite.com">
    <ns1:matrixType xsi:type="ns2:ItemMatrixType"
xmlns:ns2="urn:types.accounting_2008_2.lists.webservices.netsuite.com">_parent</ns1:matrixType>
    <ns1:itemId xsi:type="xsd:string">sweater</ns1:itemId>
    </record>
    <record externalId="sweater-Red-Large" xsi:type="ns3:InventoryItem"
xmlns:ns3="urn:accounting_2008_2.lists.webservices.netsuite.com">
    <ns3:matrixType xsi:type="ns4:ItemMatrixType"
xmlns:ns4="urn:types.accounting_2008_2.lists.webservices.netsuite.com">_child</ns3:matrixType>
    <ns3:itemId xsi:type="xsd:string">sweater-Red-Large</ns3:itemId>
    <ns3:parent externalId="parentSweater" xsi:type="ns5:RecordRef"
xmlns:ns5="urn:core_2008_2.platform.webservices.netsuite.com"/>
    <ns3:matrixOptionList xsi:type="ns3:MatrixOptionList">
    <ns3:matrixOption internalId="CUSTITEM_COLOR" xsi:type="ns6:SelectCustomFieldRef"
xmlns:ns6="urn:core_2008_2.platform.webservices.netsuite.com">
    <ns6:value internalId="1" typeId="1" xsi:type="ns6:ListOrRecordRef">
    <ns6:name xsi:type="xsd:string">Red</ns6:name>
    </ns6:value>
    </ns3:matrixOption>
    <ns3:matrixOption internalId="CUSTITEM_SIZE" xsi:type="ns7:SelectCustomFieldRef"
xmlns:ns7="urn:core_2008_2.platform.webservices.netsuite.com">
    <ns7:value internalId="2" typeId="2" xsi:type="ns7:ListOrRecordRef">
    <ns7:name xsi:type="xsd:string">Large</ns7:name>
    </ns7:value>
    </ns3:matrixOption>
    </ns3:matrixOptionList>
    </record>
    <record externalId="sweater-Green-Small" xsi:type="ns8:InventoryItem"
xmlns:ns8="urn:accounting_2008_2.lists.webservices.netsuite.com">
    <ns8:matrixType xsi:type="ns9:ItemMatrixType"
xmlns:ns9="urn:types.accounting_2008_2.lists.webservices.netsuite.com">_child</ns8:matrixType>
    <ns8:itemId xsi:type="xsd:string">sweater-Green-Small</ns8:itemId>
    <ns8:parent externalId="parentSweater" xsi:type="ns10:RecordRef"
xmlns:ns10="urn:core_2008_2.platform.webservices.netsuite.com"/>
    <ns8:matrixOptionList xsi:type="ns8:MatrixOptionList">
    <ns8:matrixOption internalId="CUSTITEM_COLOR" xsi:type="ns11:SelectCustomFieldRef"
xmlns:ns11="urn:core_2008_2.platform.webservices.netsuite.com">
    <ns11:value internalId="2" typeId="1" xsi:type="ns11:ListOrRecordRef">
    <ns11:name xsi:type="xsd:string">Green</ns11:name>
    </ns11:value>
    </ns8:matrixOption>
    <ns8:matrixOption internalId="CUSTITEM_SIZE" xsi:type="ns12:SelectCustomFieldRef"
xmlns:ns12="urn:core_2008_2.platform.webservices.netsuite.com">
    <ns12:value internalId="3" typeId="2" xsi:type="ns12:ListOrRecordRef">
    <ns12:name xsi:type="xsd:string">Small</ns12:name>
    </ns12:value>
    </ns8:matrixOption>
    </ns8:matrixOptionList>
    </record>
    <record externalId="sweater-Blue-Large" xsi:type="ns13:InventoryItem"
xmlns:ns13="urn:accounting_2008_2.lists.webservices.netsuite.com">
    <ns13:matrixType xsi:type="ns14:ItemMatrixType"
xmlns:ns14="urn:types.accounting_2008_2.lists.webservices.netsuite.com">_child</ns13:matrixType>
    <ns13:itemId xsi:type="xsd:string">sweater-Blue-Large</ns13:itemId>

```

```

        <ns13:parent externalId="parentSweater" xsi:type="ns15:RecordRef"
xmlns:ns15="urn:core_2008_2.platform.webservices.netsuite.com"/>
        <ns13:matrixOptionList xsi:type="ns13:MatrixOptionList">
            <ns13:matrixOption internalId="CUSTITEM_COLOR" xsi:type="ns16:SelectCustomFieldRef"
xmlns:ns16="urn:core_2008_2.platform.webservices.netsuite.com">
                <ns16:value internalId="3" typeId="1" xsi:type="ns16:ListOrRecordRef">
                    <ns16:name xsi:type="xsd:string">Blue</ns16:name>
                </ns16:value>
            </ns13:matrixOption>
            <ns13:matrixOption internalId="CUSTITEM_SIZE" xsi:type="ns17:SelectCustomFieldRef"
xmlns:ns17="urn:core_2008_2.platform.webservices.netsuite.com">
                <ns17:value internalId="2" typeId="2" xsi:type="ns17:ListOrRecordRef">
                    <ns17:name xsi:type="xsd:string">Large</ns17:name>
                </ns17:value>
            </ns13:matrixOption>
        </ns13:matrixOptionList>
    </record>
    <record externalId="sweater-Red-Small" xsi:type="ns18:InventoryItem"
xmlns:ns18="urn:accounting_2008_2.lists.webservices.netsuite.com">
        <ns18:matrixType xsi:type="ns19:ItemMatrixType"
xmlns:ns19="urn:types.accounting_2008_2.lists.webservices.netsuite.com">_child</ns18:matrixType>
        <ns18:itemId xsi:type="xsd:string">sweater-Red-Small</ns18:itemId>
        <ns18:parent externalId="parentSweater" xsi:type="ns20:RecordRef"
xmlns:ns20="urn:core_2008_2.platform.webservices.netsuite.com"/>
        <ns18:matrixOptionList xsi:type="ns18:MatrixOptionList">
            <ns18:matrixOption internalId="CUSTITEM_COLOR" xsi:type="ns21:SelectCustomFieldRef"
xmlns:ns21="urn:core_2008_2.platform.webservices.netsuite.com">
                <ns21:value internalId="1" typeId="1" xsi:type="ns21:ListOrRecordRef">
                    <ns21:name xsi:type="xsd:string">Red</ns21:name>
                </ns21:value>
            </ns18:matrixOption>
            <ns18:matrixOption internalId="CUSTITEM_SIZE" xsi:type="ns22:SelectCustomFieldRef"
xmlns:ns22="urn:core_2008_2.platform.webservices.netsuite.com">
                <ns22:value internalId="3" typeId="2" xsi:type="ns22:ListOrRecordRef">
                    <ns22:name xsi:type="xsd:string">Small</ns22:name>
                </ns22:value>
            </ns18:matrixOption>
        </ns18:matrixOptionList>
    </record>
    <record externalId="sweater-Green-Large" xsi:type="ns23:InventoryItem"
xmlns:ns23="urn:accounting_2008_2.lists.webservices.netsuite.com">
        <ns23:matrixType xsi:type="ns24:ItemMatrixType"
xmlns:ns24="urn:types.accounting_2008_2.lists.webservices.netsuite.com">_child</ns23:matrixType>
        <ns23:itemId xsi:type="xsd:string">sweater-Green-Large</ns23:itemId>
        <ns23:parent externalId="parentSweater" xsi:type="ns25:RecordRef"
xmlns:ns25="urn:core_2008_2.platform.webservices.netsuite.com"/>
        <ns23:matrixOptionList xsi:type="ns23:MatrixOptionList">
            <ns23:matrixOption internalId="CUSTITEM_COLOR" xsi:type="ns26:SelectCustomFieldRef"
xmlns:ns26="urn:core_2008_2.platform.webservices.netsuite.com">
                <ns26:value internalId="2" typeId="1" xsi:type="ns26:ListOrRecordRef">
                    <ns26:name xsi:type="xsd:string">Green</ns26:name>
                </ns26:value>
            </ns23:matrixOption>
            <ns23:matrixOption internalId="CUSTITEM_SIZE" xsi:type="ns27:SelectCustomFieldRef"
xmlns:ns27="urn:core_2008_2.platform.webservices.netsuite.com">

```

```

        <ns27:value internalId="2" typeId="2" xsi:type="ns27:ListOrRecordRef">
            <ns27:name xsi:type="xsd:string">Large</ns27:name>
        </ns27:value>
    </ns23:matrixOption>
</ns23:matrixOptionList>
</record>
<record externalId="sweater-Blue-Small" xsi:type="ns28:InventoryItem"
xmlns:ns28="urn:accounting_2008_2.lists.webservices.netsuite.com">
    <ns28:matrixType xsi:type="ns29:ItemMatrixType"
xmlns:ns29="urn:types.accounting_2008_2.lists.webservices.netsuite.com">_child</ns28:matrixType>
    <ns28:itemId xsi:type="xsd:string">sweater-Blue-Small</ns28:itemId>
    <ns28:parent externalId="parentSweater" xsi:type="ns30:RecordRef"
xmlns:ns30="urn:core_2008_2.platform.webservices.netsuite.com"/>
    <ns28:matrixOptionList xsi:type="ns28:MatrixOptionList">
        <ns28:matrixOption internalId="CUSTITEM_COLOR" xsi:type="ns31:SelectCustomFieldRef"
xmlns:ns31="urn:core_2008_2.platform.webservices.netsuite.com">
            <ns31:value internalId="3" typeId="1" xsi:type="ns31:ListOrRecordRef">
                <ns31:name xsi:type="xsd:string">Blue</ns31:name>
            </ns31:value>
        </ns28:matrixOption>
        <ns28:matrixOption internalId="CUSTITEM_SIZE" xsi:type="ns32:SelectCustomFieldRef"
xmlns:ns32="urn:core_2008_2.platform.webservices.netsuite.com">
            <ns32:value internalId="3" typeId="2" xsi:type="ns32:ListOrRecordRef">
                <ns32:name xsi:type="xsd:string">Small</ns32:name>
            </ns32:value>
        </ns28:matrixOption>
    </ns28:matrixOptionList>
</record>
</addList>
</soapenv:Body>
</soapenv:Envelope>

```

Java

```

/** Create Sweaters as matrix items.
 * First create the parent - no matrix properties except "Matrix Type" is Parent
 * Second create the matrix children with a combination of sizes and colors.
 * This can be done in a single addList (as shown).
 */
//Define mrr method
public static RecordRef mrr(String internalId)
{
    RecordRef toRet = new RecordRef();
    toRet.setInternalId(internalId);
    return toRet;
}

// Define makeListOrRecordRef method
public static ListOrRecordRef makeListOrRecordRef(String sTypeId, String internalId, String sName)
{
    ListOrRecordRef toRet = new ListOrRecordRef();
    toRet.setInternalId(internalId);
    toRet.setName(sName);
    toRet.setTypeId(sTypeId);
    return toRet;
}

```

```

public void testMatrixSample() throws Exception
{
// Color is a Custom List of TypeId/RecType 1 that has already been created. 1,2,3 represent the
// internalIds of Red, Green, Blue
ListOrRecordRef[] colorArray = new
    ListOrRecordRef[] {makeListOrRecordRef("1","1","Red"), makeListOrRecordRef("1","2","Green"),
        makeListOrRecordRef("1","3","Blue")}; // Representing red, green and blue

// Size is a CustomList of TypeId/RecType 2 that has already been created
ListOrRecordRef[] sizeArray = new
ListOrRecordRef[] {makeListOrRecordRef("2","2","Large"),makeListOrRecordRef("2","3","Small")};

//Representing large and small
InventoryItem[] toSubmit = new InventoryItem[1+colorArray.length*sizeArray.length];

toSubmit[0] = new InventoryItem();
toSubmit[0].setExternalId("parentSweater");
toSubmit[0].setItemId("sweater");
toSubmit[0].setMatrixType(ItemMatrixType._parent);
// set other fields on the Parent

for (int i=0;i<colorArray.length*sizeArray.length;i++)
{
    toSubmit[i+1] = new InventoryItem();
    toSubmit[i+1].setMatrixType(ItemMatrixType._child);
    // mrr Creates a recordRef given an internal and externalId, the latter of which we specify.
    // This makes it so we can submit all the records at once
    toSubmit[i+1].setParent(mrr((String)null,"parentSweater"));
    // "sweater-large-red","sweater-large-green"...
    toSubmit[i+1].setItemId("sweater-"+colorArray[i%3].getName() + "-" +
        sizeArray[i % 2].getName());
    // set externalId so it's easier to find later
    toSubmit[i+1].setExternalId(toSubmit[i+1].getItemId());

    // CUSTITEM_COLOR,SIZE are the names of the Item Custom Fields, applied to
    //InventoryItem that were setup as a Matrix types.
    SelectCustomFieldRef colorRef = new SelectCustomFieldRef();
    colorRef.setInternalId("CUSTITEM_COLOR");
    colorRef.setValue(colorArray[i%3]);

    SelectCustomFieldRef sizeRef = new SelectCustomFieldRef();
    sizeRef.setInternalId("CUSTITEM_SIZE");
    sizeRef.setValue(sizeArray[i%2]);

    toSubmit[i+1].setMatrixOptionList(new MatrixOptionList(new
        SelectCustomFieldRef[] {colorRef,sizeRef}));

    // Set other matrix item child files
    //....
}

WriteResponseList wr = c.getPort().addList(toSubmit);
}

```

Working with the Pricing Matrix

The following method takes an `InventoryItem` record and sets its pricing matrix. It sets the base price at quantity 0. If an invalid price was entered (a non-numeric value), it does not set the pricing matrix.

Sample Java Code

```
private void createPricingMatrix(InventoryItem item)
{
    _console.write("\nPlease enter the base price: ");
    String priceString = _console.readLine();
    Price[] prices = new Price[1];
    prices[0] = new Price();
    try
    {
        prices[0].setValue(Double.valueOf(priceString));
        prices[0].setQuantity(null);

        PriceList priceList = new PriceList();
        priceList.setPrice(prices);

        Pricing[] pricing = new Pricing[1];
        pricing[0] = new Pricing();
        pricing[0].setPriceList(priceList);
        pricing[0].setDiscount(null);
        RecordRef priceLevel = new RecordRef();
        priceLevel.setInternalId(BASE_PRICE_LEVEL_INTERNAL_ID);
        priceLevel.setType(RecordType.priceLevel);
        pricing[0].setPriceLevel(priceLevel);
        RecordRef currUSD = new RecordRef();
        currUSD.setInternalId("1");
        pricing[0].setCurrency(currUSD);

        PricingMatrix pricingMatrix = new PricingMatrix();
        pricingMatrix.setPricing(pricing);
        pricingMatrix.setReplaceAll(false);

        item.setPricingMatrix(pricingMatrix);
    }
    catch (NumberFormatException e)
    {
        _console.error("\nInvalid base price entered: " + priceString + ".
            Proceed creating item without setting pricing matrix.");
    }
}
```

Shared Field Definitions for Items

The tables in this section provide field definitions for complex types that are shared across multiple item types. See the following shared item lists:

- [Pricing Matrix List](#)
- [Billing Rates Matrix List](#)

- Item Member List
- Item Options List
- Translation List
- Item Vendor List
- Site Category List

Pricing Matrix List

The Pricing Matrix List provides various pricings for a given item. For each item, you can set multiple pricings based on the following:

- Currency the item is offered in
- Price levels available for the item — as defined in the price level user defined list
- The quantity of items being sold — when the Quantity Pricing feature is enabled

Field Name	Type / Field Length	Req.	Mapping	Notes
currency	RecordRef	Y/N	NA	References the currency the price level will be set for. This field is required when the multi-currency preference is ON. Otherwise, the default currency is sourced for this field and it is NOT required.
discount	double	Y	Pricing / Default Discount %	This is a read-only field that returns the discount rate associated with the priceLevel.
priceLevel	RecordRef	Y	Pricing / Price Level	This is a read-only field that references values in a user-definable list at Setup > Accounting > Setup Tasks > Accounting Lists > New > Price Level. For details on how to edit this list, see "Price List" on page 122.
pricelist	List	N		See "Price List" on page 122.

Price List

To provide multiple entries in the Price List, the Quantity Pricing feature must be enabled. Otherwise, only one Price List entry should be submitted. Notice that in the UI, the quantity for multiple pricings is entered only once. In Web services, however, you must submit the quantity for **each** pricing. The quantity values for each price submitted in the same list **must match**. If they do NOT match, an error is returned.

Field Name	Type / Field Length	Req.	Mapping	Notes
quantity	double		Qty	Sets the quantity required to receive this price.
value	double		Pricing / amount text box	This is calculated based on the Base Price level and discount rate for the specified priceLevel but can be overwritten.

Billing Rates Matrix List

Use the Billing Rates list to set pricing for each billing class on service item records.

Because the Billing Classes feature is not compatible with the Quantity Pricing feature, you cannot use both features at once. If both are enabled, then billing classes replace quantity pricing on service item records.

Billing Rates

Field Name	Type / Field Length	Req.	Mapping	Notes
billingClass	RecordRef		Billing Rates / Billing Class	References a value in a user defined list at Setup > Accounting > Setup Tasks > Accounting Lists > New > Billing Class.
currency	RecordRef			References a value in a user-defined list at Lists > Accounting > Currencies. (Note that the Multiple Currencies feature must be enabled before you can set currency values.) This value sets the currency that all transactions involving this customer are conducted in. If defaults are OFF, this field is required. In order to retrieve a list of available values for this field, use the GetSelectValue operation. For more information, see getSelectValue of the Platform Guide.
rateList	List		Billing Rates	See "Rate" on page 123.

Rate

Field Name	Type / Field Length	Req.	Mapping	Notes
priceLevel	RecordRef		Billing Rates / [Various Price Level]	Sets the price Level this value is for. Price Levels are defined in a user defined list at Setup > Accounting > Setup Tasks > Accounting Lists > New > Price Level. For details on how to edit this list via Web services, see "Price Level" on page 165.
value	double		Billing Rates / Amount field for each price level	Sets the amount for the associated priceLevel.

Item Member List

Field Name	Type / Field Length	Req.	Mapping	Notes
item	RecordRef			
memberDescr	string			

Field Name	Type / Field Length	Req.	Mapping	Notes
memberUnit	string			
quantity	double			
replaceAll	boolean			
taxcode	string			
taxrate	double			
taxSchedule	RecordRef			

Item Options List

Field Name	Type / Field Length	Req.	Mapping	Notes
itemOptions	recordRef	N	Custom / Item Options	References an existing transaction item options record at Setup > Customization > Transaction Item Options.

Translation List

Field Name	Type / Field Length	Req.	Mapping	Notes
description	string			
displayName	string			
featuredDescription	string		Featured Description	
language	string			
name	string			
noPriceMessage	string			
outOfStockMessage	string			
pageTitle	string	N	Store / Page Title	Sets the display title in the upper-left corner of an Internet browser when customers view this item in your Web store.
replaceAll	boolean			
salesDescription	string	N	Basic / Sales Description	Sets the description displayed when an item's store display name is clicked.
specialsDescription	string / 4000	N	Specials / Specials Description	Settable only if onSpecial is set to True. You can provide letters, numbers and basic HTML code.
storeDescription	string / 999	N	Store / Store Description	Sets the item description. This field can contain plain text as well as basic html code.

Field Name	Type / Field Length	Req.	Mapping	Notes
storeDetailedDescription	string / 4000	N	Store / Detailed Description	Sets the detailed item description. This field can contain plain text as well as basic html code.
storeDisplayName	string			

Item Vendor List

Use the Vendors List to add a list of vendors that can be used for this item. This information can then be accessed from the item record and from the corresponding vendor record.

The Multiple Vendors feature must be enabled in order to define item codes and purchase prices for multiple vendors on each item record. To enable this feature, go to Setup > Company > Enable Features, on the Items & Inventory tab, in the Items section, check the Multiple Vendors box.

Field Name	Type / Field Length	Req.	Mapping	Notes
preferredVendor	boolean	N	Basic / Preferred	You can only set one vendor in the list as preferred. If this is set true for multiple vendors, then the last vendor in the list is set as true and all others revert to false.
purchasePrice	double	N	Basic / Purchase Price	Sets the purchase price for this item when purchased from this vendor.
schedule	RecordRef			This field points to the quantity pricing schedule record, but is unique to the vendor and can only be created from the vendor record. In the UI go to Financial > Schedules > New Pricing Schedule.
subsidiary	string	N	Subsidiary	
vendor	RecordRef	Y	Basic / Vendor	References an existing vendor record. This is required for each vendor being defined. In order to retrieve a list of available values for this field, use the <code>GetSelectValue</code> operation. For more information, see getSelectValue of the Platform Guide.
vendorCode	string / 15	N	Basic / Vendor	Sets the vendor's item code.
vendorCurrencyName	string			

Site Category List

This list defines the web site category for the item.

Field Name	Type / Field Length	Req.	Mapping	Notes
category	RecordRef	Y	Site Category	References a value in a user-defined list at Lists > Web Site > Categories.
categoryDescription	string / 50	Y	Description	A read-only field that returns the description defined in the category record.
website	RecordRef			

Item Search

Nearly all item record types use the ItemSearch record for search. The *basic* element in this record references the ItemSearchBasic record, which lists all available search filter fields available in an item search.

The ItemSearch record also lists all search joins available in an item search. In the [SuiteTalk Schema Browser](#), see **lists accounting.xsd** for the ItemSearch record.

Be aware that the actual search filter fields available will vary depending on the item type you are searching against. For example, not *all* of the search filter fields defined in ItemSearchBasic will exist on the Subtotal Item record.

When the Multiple Prices feature is enabled, the otherPrices field, in the ItemSearchAdvanced record, returns all existing prices other than the basePrice, including alternate prices and online price. This behavior matches that of the Other Prices search results field in UI advanced item searches.

Important: By default only a record's body fields are returned on a search. Therefore, you must set the **bodyFieldsOnly** element of the SearchPreferences type to **false** if you want to also return the information specified on a record's sublist. For information on setting search preferences, see the section "Setting Search Preferences" in the *SuiteTalk (Web Services) Platform Guide*. For general information on searching in Web services, see *search()* in the "Operations" section of the *SuiteTalk (Web Services) Platform Guide*.

Assembly Item (BOM Item)

Assembly Item records are also referred to as bill of material items (BOM items).

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Download Item

Create Download Item records for files you want customers 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 a Download Item record for each song. If you want to charge customers per album, you would create one Download Item record and attach each song on the album to the record.

In Web services, the Download Item record is defined in the [listAcct \(accounting\) XSD](#).

Important: The Download Item record type is not accessible until the **Sell Download Files** feature is enabled in your account. In Web services an error is thrown if you attempt create, update, delete or search for a Download Item without first enabling the **Sell Download Files** feature. To enable this feature, a NetSuite administrator can go to Setup > Company > Enable Features > select the Items & Inventory subtab > select the Sell Downloadable Files check box > click Save.

For general information on the Download Item record, see these topics in the NetSuite Help Center:

- [Download Items](#)
- [Setting Up Items for the Web Site](#)
- [Items and Inventory Overview](#)

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchMoreWithId | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Please note the following when working with the Download Items record:

- The Download Items record does not have its own search interface. Like all other item types in NetSuite, you must use the [Item Search](#) record to retrieve information for this record type.
- Working with the Download Item is generally a two-step process. You must first create the Download Item record using the **add** operation, then you must use the **attach** operation to attach all associated files. Note that an error is thrown if you attempt to attach a file that does not already exist in NetSuite. If you choose, you can later use the **detach** operation to remove a file from a Download Item record.

Note: You *can* create both the Download Item record and file record in one step using the `addList` operation. However, to attach any file to the download item record, you will still need to call the `attach` operation.

- In Web services there is a 10 MB file size limit for file upload. If you attempt to upload a file larger than 10 MB to attach to a Download Item record, an error is thrown.
- When you do a **get** on Download Item records, the details of the files attached to the record are not returned in your SOAP response. If you need to get the list of files attached to a Download Item, you must do an advanced search on the item record and specify the fields on the file record that you want returned. If you want to update the files attached to a Download Item record, you must update the file record directly.
- To search on the files associated with a Download Item record, use the file join on the [Item Search](#) record.

Sample Code

C#

```
NetSuiteService nss = new NetSuiteService();

ItemSearchAdvanced itemSearchAdv = new ItemSearchAdvanced();

// Create search criteria
ItemSearch itemSearch = new ItemSearch();
ItemSearchBasic itemSearchBasic = new ItemSearchBasic();

// Set item search type to downloadItem
SearchEnumMultiSelectField itemTypeFld = new SearchEnumMultiSelectField();
String [] itemTypes = new String[1];
itemTypes[0] = "_downloadItem";
itemTypeFld.searchValue = itemTypes;
itemTypeFld.operatorSpecified = true;
itemTypeFld.operator = SearchEnumMultiSelectFieldOperator.anyOf;
itemSearchBasic.type = itemTypeFld;

// Select search return columns
ItemSearchRow itemRow = new ItemSearchRow();
```

```

// Select to return file internal id
FileSearchRowBasic fileRowBasic = new FileSearchRowBasic();

SearchColumnSelectField [] selectColumns = new SearchColumnSelectField[1];
SearchColumnSelectField selectColumn = new SearchColumnSelectField();
selectColumns[0] = selectColumn;
fileRowBasic.internalId = selectColumns;

// Select to return file name
SearchColumnStringField [] stringColumns = new SearchColumnStringField[1];
SearchColumnStringField stringColumn = new SearchColumnStringField();
stringColumns[0] = stringColumn;
fileRowBasic.name = stringColumns;

// Set file join
itemRow.fileJoin = fileRowBasic;

// Select to return item internalId
ItemSearchRowBasic itemRowBasic = new ItemSearchRowBasic();
itemRowBasic.internalId = selectColumns; // column definition can be reused
itemRow.basic = itemRowBasic;

// Set item search criteria
itemSearch.basic = itemSearchBasic;
itemSearchAdv.criteria = itemSearch;

// Set item return columns
itemSearchAdv.columns = itemRow;

// Perform search
SearchResult searchResult = nss.search(itemSearchAdv);

```

SOAP Request

```

<search xmlns="urn:messages_2009_2.platform.webservices.netsuite.com">
  <searchRecord xmlns:q1="urn:accounting_2009_2.lists.webservices.netsuite.com"
xsi:type="q1:ItemSearchAdvanced">
    <q1:criteria>
      <q1:basic>
        <type operator="anyOf" xmlns="urn:common_2009_2.platform.webservices.netsuite.com">
          <searchValue xmlns="urn:core_2009_2.platform.webservices.netsuite.com">_downloadItem</
searchValue>
        </type>
      </q1:basic>
    </q1:criteria>
    <q1:columns>
      <q1:basic>
        <internalId xmlns="urn:common_2009_2.platform.webservices.netsuite.com" />
      </q1:basic>
      <q1:fileJoin>
        <internalId xmlns="urn:common_2009_2.platform.webservices.netsuite.com" />
        <name xmlns="urn:common_2009_2.platform.webservices.netsuite.com" />
      </q1:fileJoin>
    </q1:columns>
  </searchRecord>
</search>

```

SOAP Response

```
<platformCore:searchRowList>
  <platformCore:searchRow xsi:type="listAcct:ItemSearchRow"
xmlns:listAcct="urn:accounting_2009_2.lists.webservices.netsuite.com">
  <listAcct:basic
xmlns:platformCommon="urn:common_2009_2.platform.webservices.netsuite.com">
  <platformCommon:internalId>
  <platformCore:searchValue internalId="105"/>
  </platformCommon:internalId>
  </listAcct:basic>
  <listAcct:fileJoin
xmlns:platformCommon="urn:common_2009_2.platform.webservices.netsuite.com">
  <platformCommon:internalId>
  <platformCore:searchValue internalId="239"/>
  </platformCommon:internalId>
  <platformCommon:name>
  <platformCore:searchValue>IDreamedADreamLyrics.txt</platformCore:searchValue>
  </platformCommon:name>
  </listAcct:fileJoin>
  </platformCore:searchRow>
</platformCore:searchRowList>
```

Inventory Item

Inventory item records are used to track information about items which you maintain a stock of.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Working with Inventory Items Sublists

The SuiteTalk Schema Browser includes all sublists associated with the Inventory Item record. See the following information for usage notes regarding specific Inventory Item sublists. Usage notes are not provided for every sublist type.

InventoryItemLocations

In order to provide a locations list, the Multi-Location Inventory feature must be enabled at Setup > Company > Enable Features. Click the Items & Inventory subtab, and select the Multi-Location Inventory check box. Otherwise, single entries for each corresponding field can be entered in the regular record fields.

ItemVendorList

The Vendor sublist (ItemVendorList) on Inventory Items contains a **schedule** field that holds pricing schedule values. Note that when this field is set, it triggers a recalc on add and update operations.

Kit/Package Item

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 Web site. Whenever you sell a kit, inventory items are deducted from inventory.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Gift Certificate Item

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

This sample shows how to add a gift certificate item. Note that the following has been set in the NetSuite account: Setup > Accounting > Accounting Preferences > Items/Transactions tab > Other Items Preferences > Gift Certificate Auth Code Generation = System Generated.

SOAP Request

```
soapenv:Body>
  <add xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
    <record xsi:type="ns6:GiftCertificateItem"
xmlns:ns6="urn:accounting_2_6.lists.webservices.netsuite.com">
      <ns6:itemId xsi:type="xsd:string">BeyondMotorSports.com Gift Certificate</ns6:itemId>
      <ns6:isInactive xsi:type="xsd:boolean">>false</ns6:isInactive>
      <ns6:liabilityAccount internalId="28" type="account" xsi:type="ns7:RecordRef"
xmlns:ns7="urn:core_2_6.platform.webservices.netsuite.com" />
      <ns6:rate xsi:type="xsd:double">100.0</ns6:rate>
      <ns6:taxSchedule internalId="1" xsi:type="ns8:RecordRef"
xmlns:ns8="urn:core_2_6.platform.webservices.netsuite.com" />
    </record>
  </add>
</soapenv:Body>
```

SOAP Response

```
<soapenv:Body>
  <addResponse xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
    <writeResponse>
      <ns2:status isSuccess="true" xmlns:ns2="urn:core_2_6.platform.webservices.netsuite.com" />
      <baseRef internalId="93" type="giftCertificateItem" xsi:type="ns3:RecordRef"
xmlns:ns3="urn:core_2_6.platform.webservices.netsuite.com" />
    </writeResponse>
  </addResponse>
</soapenv:Body>
```

Java

```
GiftCertificateItemAuthCodes authCode1 = new GiftCertificateItemAuthCodes();
authCode1.setAuthCode("AUTHCODE003");

GiftCertificateItemAuthCodes authCode2 = new GiftCertificateItemAuthCodes();
authCode2.setAuthCode("AUTHCODE004");

GiftCertificateItemAuthCodes[] authCodes = new GiftCertificateItemAuthCodes[2];
authCodes[0] = authCode1;
authCodes[1] = authCode2;

GiftCertificateItemAuthCodesList authCodesList = new GiftCertificateItemAuthCodesList();
authCodesList.setAuthCodes(authCodes);
authCodesList.setReplaceAll(true);

RecordRef liabilityAccount = new RecordRef();
liabilityAccount.setInternalId("28");
liabilityAccount.setType(RecordType.account);

RecordRef taxSchedule = new RecordRef();
taxSchedule.setInternalId("1");
```

```
GiftCertificateItem gcltem = new GiftCertificateItem();
gcltem.setItemId("BeyondMotorSports.com Gift Certificate");
gcltem.setLiabilityAccount(liabilityAccount);
gcltem.setTaxSchedule(taxSchedule);
gcltem.setRate(100.00);
gcltem.setIsInactive(false);

WriteResponse response = _port.add(gcltem);
WriteResponse response = _port.add(gci);
```

Lot Numbered Assembly Item

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 in order to track it as it enters and leaves your inventory.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Lot Numbered Inventory Item

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.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Working with Lot Numbered Inventory Items Sublists

The SuiteTalk Schema Browser includes all sublists associated with the Lot Numbered Inventory Item record. See the following information for usage notes regarding specific Lot Numbered Inventory Item sublists. Usage notes are not provided for every sublist type.

LotNumberedInventoryItemLocations

In order to provide a locations list for Lot Numbered Inventory Items, the Multi-Location Inventory feature must be enabled at Setup > Company > Enable Features > Accounting. Otherwise, single entries for each corresponding field can be entered in the regular record fields.

LotNumberedInventoryItemNumbers

This list is read-only and displays the serial number and quantity on hand for the lot numbered inventory item.

Serialized Inventory Item

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.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Working with Serialized Inventory Items Sublists

The SuiteTalk Schema Browser includes all sublists associated with the Serialized Inventory Item record. See the following information for usage notes regarding specific Serialized Inventory Item sublists. Usage notes are not provided for every sublist type.

SerializedInventoryItemLocations

In order to provide a locations list, the Multi-Location Inventory feature must be enabled at Setup > Company > Enable Features > Accounting. Otherwise, single entries for each corresponding field can be entered in the regular record fields.

SerializedInventoryItemNumbers

This list is read-only and displays the serial number and quantity on hand for the lot numbered inventory item.

Description Item

Description item records are used to create a description line you can add to your transactions. Use the Description item to enter long descriptions, without amounts, as line items.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Discount Item

Discount item records are used to create discounts you can apply to your transactions.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Markup Item

Markup item records are used to create markups you can apply to your transactions.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Payment Item

Payment item records are used for partial payments, such as showing a down payment when you create an invoice. Payment items can be assigned to post to your un-deposited funds account.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Subtotal Item

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.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Noninventory Purchase Item

Noninventory Purchase item records are used to track something you buy but don't sell, like office supplies for your business.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Noninventory Sales Item

Noninventory Sales item records are used to track items you sell but don't buy, such as a restaurant entree.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Noninventory Resale Item

Noninventory Resale item records are used to track something you buy and then sell for a profit, such as a case of soda.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Other Charge Resale Item

Other Charge Resale item records are used to track charges that your business pays for initially and then bills to a customer, like freight charges.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Other Charge Purchase Item

Other Charge Purchase item records are used to track charges that your business pays for that are not billed to a customer.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Serialized Assembly Item

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.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Service Resale Item

Service Resale item records are used to track charges for a service your business pays for initially and then bills to a customer.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Service Purchase Item

Service Purchase item records are used to track charges for a service your business pays for that is not billed to a customer.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Service Sale Item

Service Sale item records are used to track charges for a service your business performs that is billed to a customer.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Other Charge Sale Item

Other Charge Sale item records are used to track charges for something a business creates or performs and charges a customer for, like a charge for gift wrapping.

Supported Operations

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Chapter 7 Support

The following support records are supported in SuiteTalk:

- Issue
- Support Case
- Support Case Status
- Support Case Type
- Support Case Origin
- Support Case Issue
- Support Case Priority
- Solution
- Topic

Issue

The Issue record is defined in the [listSupport \(support\)](#) XSD.

Supported Operations

The following operations can be used with Issue records.

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists support.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Support Case

Support Cases are support issues logged for a given company.

The SupportCase record is defined in the [listSupport \(support\)](#) XSD.

Supported Operations

The following operations can be used with the Support Case record.

add | addList | attach / detach | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Note: The getDeleted operation is NOT supported for Cases.

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists support.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Case Status and Priority Internal IDs

The following table lists the internal IDs for all standard Case Status and Priority values that can be used to populate the status and priority fields.

The status and priority values can be deleted or recreated so that the following internal ID values may differ for your organization. If the Show Internal IDs preference is enabled, you can confirm the internal ID values in the associated list. See [Setting the Internal ID Preference in the Web Services Platform Guide](#).

Case Statuses		Case Priorities	
ID	Status	ID	Priority
1	Not Started	1	High
2	In Progress	2	Medium
3	Escalated	3	Low
4	Re-Opened		
5	Closed		

Working with Support Case Sublists

The SuiteTalk Schema Browser includes all sublists associated with the Support Case record. See the following information for usage notes regarding specific Support Case sublists. Usage notes are not provided for every sublist type.

EmailEmployeesList

Use this list to provide a list of employees that should be cc'd when the record is updated. Values submitted in this list are not saved in the NetSuite database.

MessageList

The MessageList complex type has been deprecated as of SuiteTalk Version 2_0_0.

Support Case Status

Case Status records define available statuses for customer support cases. The SupportCaseStatus record is defined in the [listSupport \(support\) XSD](#).

Supported Operations

The following operations can be used with the Support Case Status record.

add | addList | delete | deleteList | get | getAll | getDeleted | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists support.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Support Case Type

Case Type records define available case types for customer support cases. The SupportCaseType record is defined in the [listSupport \(support\) XSD](#).

Supported Operations

The following operations can be used with the Support Case Type record.

add | addList | delete | deleteList | get | getAll | getDeleted | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists support.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Support Case Origin

Case Origin records define the various methods of origin for customer support cases. The SupportCaseOrigin record is defined in the [listSupport \(support\) XSD](#).

Supported Operations

The following operations can be used to manipulate the Support Case Origin record.

add | addList | delete | deleteList | get | getAll | getDeleted | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists support.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Support Case Issue

Support Case Issue records define available issue types for customer support cases. The SupportCaseIssue record is defined in the [listSupport \(support\) XSD](#).

Supported Operations

The following operations can be used with the Support Case Issue record.

add | addList | delete | deleteList | get | getAll | getDeleted | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists support.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Support Case Priority

Case Priority records define available case priorities for customer support cases. The SupportCasePriority record is defined in the [listSupport \(support\) XSD](#).

Supported Operations

The following operations can be used with the Support Case Priority record.

add | addList | delete | deleteList | get | getAll | getDeleted | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists support.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Solution

The Solution record is defined in the [listsSupport \(support\) XSD](#).

Supported Operations

The following operations can be used with the Solution record.

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists support.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Topic

The Topic record is defined in the [listsSupport \(support\) XSD](#).

Supported Operations

The following operations can be used with the Topic record.

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists support.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Chapter 8 Website

Site Category

The Site Category record is defined in the `listsWebsite (website)` XSD.

Note that when instantiating a Site Category record in PHP, you must use the name space prefix at the end of the directory - for example:

```
$newSiteCategory = new nsComplexObject('listswebsite:SiteCategory');
```

For more information on the NetSuite PHP toolkit, see [Using Web Services with PHP](#) in the NetSuite Help Center.

Supported Operations

The following operations can be used with the Site Category record:

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See `lists website.xsd` in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Root site categories (such as the Welcome Page) are not editable in Web services because they have negative internalIds. For example, the following update operation will fail when submitted:

```
<update xmlns="urn:messages_2009_1.platform.webservices.netsuite.com">
  <record internalId="-102" xsi:type="ns1:SiteCategory"
    xmlns:ns1="urn:website_2009_1.lists.webservices.netsuite.com">
    <ns1:pageTitle xsi:type="xsd:string">pageTitle</ns1:pageTitle>
  </record>
</update>
```

where:

internalId="-102" is the internal ID of a root category

If you attempt to edit/update root site categories, the following error message is returned:

Cannot update root level website categories through Web Services.

Chapter 9 Lists

The following list records are supported in SuiteTalk:

- Account
- Accounting Period
- Bin
- Budget Category
- Classification (Class)
- Currency
- Department
- Location
- Expense Category
- Gift Certificate
- Partner Category
- Sales Tax Item
- Subsidiary
- Tax Group
- Tax Type
- Unit Type
- Vendor Category

Account

The Account record is defined in the `listAcct (accounting)` XSD.

Supported Operations

The following operations can be used with the Account record.

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Currently there are no available search joins for this record.

Accounting Period

In NetSuite, transactions are posted in real time. If you choose to use accounting periods, you must enter a posting period on the transaction and select its posting period from a dropdown list on the transaction page. The posting period can be future or past, depending on the close status of the period.

To access the Accounting Periods record in the UI, go to Setup > Accounting > Manage Accounting Periods. For information on setting up Accounting Periods, see [Setting Up Accounting Periods](#) in the NetSuite Help Center.

In Web services, the Accounting Period record is defined in the [listAcct \(accounting\)](#) XSD.

Supported Operations

The following operations can be used with the Accounting Period record.

get | getList | getSavedSearch | search | searchMore | searchMoreWithId | searchNext

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Important: This record is available as a read-only record. To make changes to this record, you must do so through the UI.

Also note that to use the Accounting Periods record, a NetSuite administrator must first enable this feature in your account.

To enable Accounting Periods:

1. Go to Setup > Company > Enable Features.
2. Click the Accounting subtab.
3. Select the Accounting Periods check box.
4. Click Save.

Bin

To use Bins the Bin Management feature must be enabled at Setup > Company > Setup Tasks > Enable Features. On the Items & Inventory tab, click the Bin Management checkbox.

The Bin record is defined in the [listAcct \(accounting\)](#) XSD.

Supported Operations

The following operations can be used with the Bin record.

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Currently there are no available search joins for this record.

Budget Category

Budget Categories are used with the **Multiple Budgets** feature to create budgets for a variety of scenarios. Be aware that Budget Categories are available only if you have the Multiple Budgets feature enabled in your account. See [Usage Notes](#) for steps on enabling this feature.

The Budget Category record is defined in the `listAcct (accounting)` XSD.

Supported Operations

The following operations can be used with the Budget Category record.

add | addList | delete | deleteList | get | getAll | getList | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See `lists accounting.xsd` in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

To use the Budget Category record, a NetSuite administrator must first enable the Multiple Budgets feature in your account.

To enable Multiple Budgets:

1. Go to Setup > Company > Enable Features.
2. Click the Accounting tab.
3. Select the Multiple Budget check box.
4. Click Save.

Once this feature is enabled, any user can create budget categories in the UI. For steps on creating budget categories, see [Creating Budget Categories for Local Subsidiary Budgeting](#) in the NetSuite Help Center.

Classification (Class)

The Classification record is defined in the [listAcct \(accounting\)](#) XSD.

Supported Operations

The following operations can be used with the Classification (Class) record.

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists accounting.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Currency

The Currency record is defined in the [listAcct \(accounting\)](#) XSD.

Note that you cannot use Web services to get the exchange rates specified on Currency records. In both the UI and in Web services, the exchangeRate field is a create-time-only field that is used to specify the initial exchange rate with respect to the base currency. After the initial exchange rate value is set, this value does not get updated to reflect real-time exchange rates. Therefore, if you were to attempt to get “today’s” current exchange rate, the value returned could be inaccurate.

Supported Operations

The following operations can be used with the Currency record.

add | addList | delete | deleteList | get | getAll | getList | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists accounting.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Department

The Department record is defined in the [listAcct \(accounting\)](#) XSD.

Supported Operations

The following operations can be used with the Department record.

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Location

The Location record is defined in the [listAcct \(accounting\)](#) XSD.

Supported Operations

The following operations can be used with the Location record.

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Expense Category

The Expense Category record is defined in the [listAcct \(accounting\)](#) XSD.

Supported Operations

The following operations can be used with the Expense Category record.

add | addList | delete | deleteList | get | getAll | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Gift Certificate

In the UI you can add custom item number fields to gift certificate records. To add a custom field to a gift certificate record, go to Setup > Customization > Fields > Item Number Fields > New.

To add the field to specific gift certificates, select them in the Select Items to Apply Field. To add the field to all gift certificates, select All Items, and check the Gift Certificate box.

Note: The Gift Certificate and Gift Certificate Item records are two different record types.

The Gift Certificate record is defined in the [listAcct \(accounting\) XSD](#).

Supported Operations

The following operations can be used with the Gift Certificate record.

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchMoreWithId | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

The following code samples show how to add a gift certificate to a sales order.

SOAP Request

```
<soapenv:Body>
  <add xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
    <record xsi:type="ns6:SalesOrder" xmlns:ns6="urn:sales_2_6.transactions.webservices.netsuite.com">
      <ns6:entity internalId="25" type="customer" xsi:type="ns7:RecordRef"
        xmlns:ns7="urn:core_2_6.platform.webservices.netsuite.com" />
      <ns6:itemList replaceAll="false" xsi:type="ns6:SalesOrderItemList">
        <ns6:item xsi:type="ns6:SalesOrderItem">
          <ns6:item internalId="76" type="giftCertificate" xsi:type="ns8:RecordRef"
            xmlns:ns8="urn:core_2_6.platform.webservices.netsuite.com" />
        </ns6:item>
      </ns6:itemList>
    </record>
  </add>
</soapenv:Body>
```

```

<ns6:giftCertFrom xsi:type="xsd:string">Alex Santos</ns6:giftCertFrom>
<ns6:giftCertRecipientName xsi:type="xsd:string">Janina San Miguel</ns6:giftCertRecipientName>
<ns6:giftCertRecipientEmail xsi:type="xsd:string">janina@sanmiguel.com</ns6:giftCertRecipientEmail>
</ns6:item>
</ns6:itemList>
</record>
</add>
</soapenv:Body>

```

SOAP Response:

```

<soapenv:Body>
<addResponse xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
<writeResponse>
<ns2:status isSuccess="true" xmlns:ns2="urn:core_2_6.platform.webservices.netsuite.com" />
<baseRef internalId="750" type="salesOrder" xsi:type="ns3:RecordRef"
xmlns:ns3="urn:core_2_6.platform.webservices.netsuite.com" />
</writeResponse>
</addResponse>
</soapenv:Body>

```

Java

```

RecordRef entity = new RecordRef();
entity.setInternalId("25");
entity.setType(RecordType.customer);

RecordRef itemRef = new RecordRef();
itemRef.setInternalId("76");
itemRef.setType(RecordType.giftCertificate);

SalesOrderItem item = new SalesOrderItem();
item.setItem(itemRef);
item.setGiftCertFrom("Alex Santos");
item.setGiftCertRecipientName("Janina San Miguel");
item.setGiftCertRecipientEmail("janina@sanmiguel.com");

SalesOrderItem[] itemArray = new SalesOrderItem[1];
itemArray[0] = item;

SalesOrderItemList itemList = new SalesOrderItemList();
itemList.setItem(itemArray);

SalesOrder so = new SalesOrder();
so.setEntity(entity);
so.setItemList(itemList);

WriteResponse response = _port.add(so);

if (response.getStatus().isSuccess()) {
    _console.info("\n\nThe following Sales Order was added successfully: " + ((RecordRef)
response.getBaseRef()).getInternalId());
} else {
    _console.error("The Sales Order was not added:", true);
    _console.error(getStatusDetails(response.getStatus()));
}

```

Partner Category

The Partner Category record is defined in the [listAcct \(accounting\)](#) XSD.

Supported Operations

The following operations can be used with the Partner Category record.

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists accounting.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Sales Tax Item

The Sales Tax Item record is defined in the [listAcct \(accounting\)](#) XSD.

Supported Operations

The following operations can be used with the Sales Tax Item record.

add | addList | delete | deleteList | get | getAll | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists accounting.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Subsidiary

The Subsidiary record is defined in the [listAcct \(accounting\)](#) XSD.

Supported Operations

The following operations can be used with the Subsidiary record.

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Tax Group

The Tax Group record is defined in the [listAcct \(accounting\)](#) XSD.

Supported Operations

The following operations can be used with the Tax Group record.

add | addList | delete | deleteList | get | getAll | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Tax Type

The Tax Type record is defined in the [listAcct \(accounting\)](#) XSD.

Supported Operations

The following operations can be used with the Tax Type record.

add | addList | delete | deleteList | get | getAll | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Unit Type

The Unit Type record is defined in the [listAcct \(accounting\)](#) XSD.

Supported Operations

The following operations can be used with the Unit Type record.

add | addList | delete | deleteList | get | getAll | getList | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists accounting.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Vendor Category

The Vendor Category record is defined in the [listAcct \(accounting\)](#) XSD.

Supported Operations

The following operations can be used with the Vendor Category record.

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists accounting.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Chapter 10 Other Lists

For general information about other lists in SuiteTalk, see:

- [Usage Notes for Other List Record Types](#)

The following other list record types are supported in SuiteTalk:

- Contact Category
- Contact Role
- Customer Category
- Lead Source
- Note Type
- Payment Method
- Price Level
- Pricing Group
- Sales Role
- Term
- Win Loss Reason

Usage Notes for Other List Record Types

Other List records represent user-defined lists that define items that can be selected for a specific field.

For example, in the Contact record, you can set the type of Contact in the *category* field. The *category* field references the ContactCategory record, which contains a predefined list of different contact types.

All Other List record types are defined in the [listAcct \(accounting\)](#) XSD.

Permissions

In cases where the logged in user's role does NOT have permissions for the specified Other List record, the `get()` operation will return only the key/name information instead of the whole record. This is to ensure that these lists can be manipulated when they are included as select or multiselect fields within another record.

Contact Category

Contact Category defines a list of values that are used by the Contact record to set the type of contact (see CategoryList). In the UI, this is a user defined list at Setup > Sales > CRM Lists > Contact Category.

Supported Operations

The following operations can be used with the Contact Category record.

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Contact Role

Contact Roles can be defined for each contact listed on various records including Opportunity, Customer, Vendor, and Partner.

Note that when instantiating a Contact Role record in PHP, you must use the name space prefix at the end of the directory - for example:

```
$newContactRole = new nsComplexObject('listsaccounting:ContactRole');
```

For more information on the NetSuite PHP toolkit, see [Using Web Services with PHP](#) in the NetSuite Help Center.

Supported Operations

The following operations can be used with the Contact Role record.

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Customer Category

Customer Category defines a list of values that are used by the Customer record to set the type of customer (see [Customer](#)). In the UI, reference an existing Customer Category list by going to Setup > Accounting > Setup Tasks : Accounting Lists.

To create a new Customer Category List, set the Type field at the bottom of the page to Customer Category. Next, click the New button that appears on the right side of the page.

Supported Operations

The following operations can be used with the Customer Category record.

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists accounting.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Lead Source

LeadSource defines a list of values that are used by the Customer record to set the source of the lead for this customer — such as an ad or website referral (see [Customer](#)). In the UI, a user-defined lead source list is defined at Setup > Sales > Setup Tasks > CRM Lists > New > Lead Source. Note that the Marketing Automation feature must be turned off before you can create a user-defined lead source list.

Supported Operations

The following operations can be used with the Lead Source record.

add | addList | delete | deleteList | getAll | getList | update | updateList

Note: The getAll operation should be used to retrieve values of a list since the search operation does NOT exist for this record.

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists accounting.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Note Type

This type defines a list of values that are used by the Note record to set the type of note (see [Note](#)). In the UI, this is a user defined list at Setup > Sales > Setup Tasks > CRM Lists > New > Note Type.

Supported Operations

The following operations can be used with the Note Type record.

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists accounting.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Payment Method

Supported Operations

The following operations can be used with the Payment Method record.

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchMoreWithId | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists accounting.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Price Level

Price Level defines a list of values that are used by the Opportunity and Item records to set the price level for a given item. Items can be assigned different price levels — such as Employee Price or Corporate Discount Price (see [OpportunityItemList](#)).

Supported Operations

The following operations can be used with the Price Level record.

add | addList | delete | deleteList | get | getList | getSavedSearch | getSelectValue | search | searchMore | searchMoreWithId | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Pricing Group

Pricing groups allow you to assign customer-specific price levels for groups of items.

Supported Operations

The following operations can be used with the Pricing Group record.

add | addList | delete | deleteList | get | getList | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Sales Role

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.

Supported Operations

The following operations can be used with the Sales Role record.

add | addList | delete | deleteList | getList | getSavedSearch | search | searchMore | searchMoreWithId | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Term

Terms are used to calculate the due date on invoices and bills. In the UI, this is a user defined list at Setup > Accounting > Setup Tasks > Accounting Lists.

Supported Operations

The following operations can be used with the Term record.

add | addList | delete | deleteList | getDeleted | getList | getSavedSearch | search | searchMore | searchMoreWithId | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Win Loss Reason

WinLossReason defines a list of values that are used by the Opportunity record to set the reason for a win or loss (see [Opportunity](#)). In the UI, this is a user defined list at Setup > Sales > CRM Lists > New > Win/Loss Reason.

Supported Operations

The following operations can be used with the Win Loss Reason record.

add | addList | delete | deleteList | getDeleted | getList | getSavedSearch | search | searchMore | searchMoreWithId | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists accounting.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Chapter 11 Customization

The following customization records are supported in SuiteTalk:

- Custom Record
- Custom Record Type
- Custom Record Custom Field
- Custom List
- CRM Custom Field
- Entity Custom Field
- Item Custom Field
- Item Number Custom Field
- Item Option Custom Field
- Other Custom Field
- Transaction Body Custom Field
- Transaction Column Custom Field

Note: If you are unfamiliar with NetSuite custom fields, it is recommended that you see [Custom Fields](#) in the NetSuite Help Center. This section describes the purpose and general characteristics of each custom field type, which will help you when working with the SuiteTalk customization API. For general information on working with both standard and custom fields in Web services, see [Working with Fields in Web Services](#).

Custom Record

Custom Records are entry forms based on existing record types, but customized to include fields for gathering information specific to the needs of your business. Use the Custom Record object to define records to emulate existing custom records.

The Custom Record object is defined in the [setupCustom \(customization\)](#) XSD.

Supported Operations

The following operations can be used with the Custom Record record type.

add | addList | delete | deleteList | getDeleted | get | getList | getSavedSearch | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [setup customization.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

Currently NetSuite does not support the **store values** option on custom records.

Working with References

When referencing custom records, you may need a separate CustomRecordRef type and a RecordRef type.

RecordRef and CustomRecordRef both descend from BaseRef. They are most prominently used in `get()`, and that is why they have a type attribute. CustomRecordRef has a typeId to indicate which **kind** of CustomRecord it is. Therefore, you cannot get a CustomRecord by setting the RecordRef.type to customRecord on add — you must use CustomRecordRef to specify the kind of CustomRecord.

Note: For a list of the possible typelds (or record internalIds), refer to [ListOrRecordRef](#) in the NetSuite Help Center.

CustomRecord descends from Record, just like Customer or Opportunity. This is what you submit to `add()` and it is what is returned through `search()`. CustomRecord uses a RecordRef() to store its type information.

Java Example

To add a CustomRecord

```
CustomRecord myCR = new CustomRecord();
RecordRef rt = new RecordRef();
rt.setName(" Authors");
rt.setId("3"); // This indicates a typeId of 3 and corresponds to the Authors CustomRecord type
myCR.setRecType(rt);
myCR.setName("Ernest Hemmingway"); // This is what will show up in List->
CustomRecords->Authors under Name
myCR.setCustomFieldList(myCFL); // An already filled out CustomField List
WriteResponse wr = port.add(myCR);
```

Note that setTypeId is not submitted since this is the internal ID returned by the response, just like for normal records.

To get a CustomRecord:

```
CustomRecordRef customRec = new CustomRecordRef();
customRec.setTypeId("21"); // internal id of the custom record type
customRec.setInternalId("1"); // internal id of the custom record entry
sessMgr.getWrappedPort().get(customRec, this);
```

.Net Example

To add a CustomRecord

```

CustomRecord myCR = new CustomRecord();
RecordRef rt = new RecordRef();
rt.name = "Authors";
// This indicates a typeId of 3 and corresponds to the Authors CustomRecord type
rt.internalId = "3";
customRecord.recType = rt;
// This is what will show up in List->CustomRecords->Authors under Name
customRecord.name = "Ernest Hemingway";
customRecord.customFieldList = myCFL; // An already filled out CustomField List
WriteResponse wr = port.add(customRecord);

```

To get a CustomRecord:

```

CustomRecordRef myCRR = new CustomRecordRef();
myCRR.internalId = "1200"; // The key we got back
myCRR.typeId = "3"; // For Authors
ReadResponse rr = _service.get(myCRR);

```

Sample Code

The following code retrieves the entire custom list for a list with an internal ID of 1.

```

CustomRecordSearch customRecordSearch = new CustomRecordSearch();
RecordRef recordRef = new RecordRef();
recordRef.setInternalId("1");
customRecordSearch.setRecType(recordRef);
SearchResult result = port.search(customRecordSearch);

```

Custom Record Type

For Custom Record Types, fields on the body and subtabs (Numbering, Permissions, Links, and Managers) can be set on add or update of a new Custom Record Type. Fields on the Fields, Forms, Online Forms, Child Records, and Parent Records subtabs can only be set on update to an existing Custom Record.

The Custom Record Type record is defined in the [setupCustom \(customization\)](#) XSD.

Supported Operations

The following operations can be used with Custom Record Types.

add | addList | delete | deleteList | getDeleted | get | getCustomization | getCustomizationId | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [setup customization.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Working with Custom Record Sublists

The SuiteTalk Schema Browser includes all sublists associated with the Custom Record record type. See the following information for usage notes regarding specific Custom Record sublists. Usage notes are not provided for every sublist type.

- [CustomRecordTypeChildren](#)
- [CustomRecordTypeForms](#)
- [CustomRecordTypeOnlineForms](#)
- [CustomRecordTypeParents](#)

CustomRecordTypeChildren

This is a **read-only** list that returns all custom records that define the current record as it's parent.

CustomRecordTypeForms

This is a **read-only** list that returns the forms that have been created for use with the current record type. A single default form is automatically created for any new record type.

CustomRecordTypeOnlineForms

This is a **read-only** list that returns all online forms that have been created for the current custom record type.

CustomRecordTypeParents

This is a read-only list that returns the parent records this record is a child of.

To have the entries set here active, you must set the *usePermission* field to TRUE for the record type definition.

Custom Record Custom Field

Custom Record Custom Fields can only be set on a Custom Record after the Custom Record has been created.

The Custom Record Custom Field record is defined in the [setupCustom \(customization\)](#) XSD.

Supported Operations

The following operations can be used with Custom Record Custom Fields.

add | addList | delete | deleteList | get | getList | getSelectValue | update | updateList

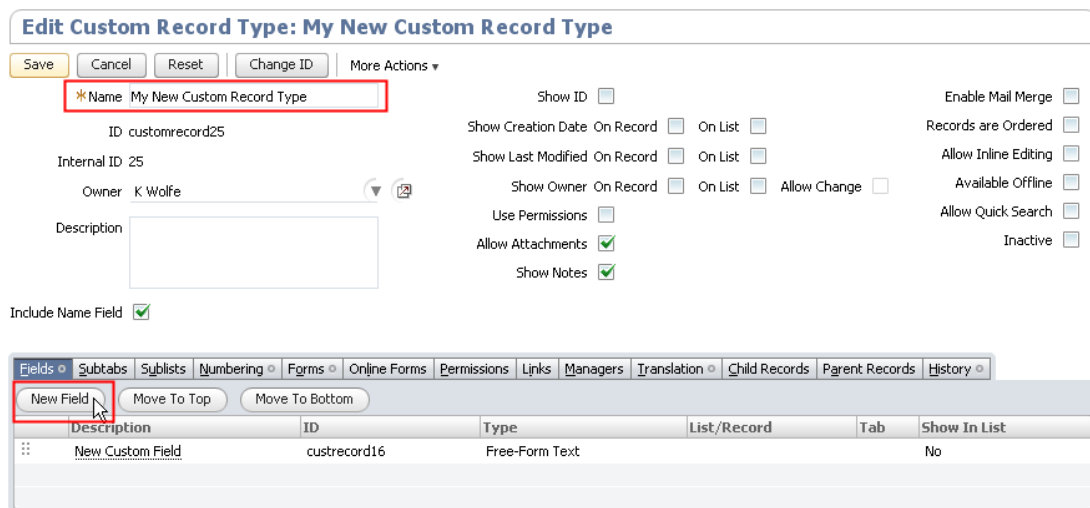
Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [setup customization.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

The following figure shows the UI equivalent of creating a custom record custom field. First you must create a new custom record type. Once the new type is created, click the New Field button on the Fields tab. This is the only way you can create a custom record custom field, which is directly associated with the custom record type. As a consequence, you cannot directly search for custom record custom fields using either the search() or getCustomization() operations. You must perform a record type search and locate your custom record custom field within that search.



Custom List

The Custom List record is defined in the `setupCustom (customization) XSD`.

Supported Operations

The following operations can be used with Custom List.

add | addList | delete | deleteList | get | getCustomization | getCustomizationId | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See `setup customization.xsd` in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

The following SOAP sample shows how to set a value from a custom list onto a record (in this case a Contact record) using Web services.

```
<soapenv:Body>
  <platformMsgs:add
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:s0="urn:relationships_2008_2.lists.webservices.netsuite.com"
    xmlns:platformMsgs="urn:messages_2008_2.platform.webservices.netsuite.com"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
    <platformMsgs:record xsi:type="s0:Contact">
      <s0:entityId>Clint Eastwood</s0:entityId>
      <s0:customFieldList>
        <platformCore:customField xsi:type="platformCore>SelectCustomFieldRef"
          internalId="custentity6">
          <platformCore:value typeId="7" internalId="2" />
        </platformCore:customField>
      </s0:customFieldList>
    </platformMsgs:record>
  </platformMsgs:add>
</soapenv:Body>
```

Where:

- `internalId="custentity6"` is the name of the custom list as it appears on the Contact record.
- `typeId="7"` is the custom list internal id (Setup > Customization > Lists - Internal ID)
- `internalId="2"` is the internal id of the second item on the custom list (Setup > Customization > Lists > List - ID)

Also note that when working with custom lists in Web services, you can update specific values in a custom list using **valueId** as the key field for the CustomListCustomValue sublist and setting `replaceAll` flag to FALSE.

CRM Custom Field

The CRM Custom Field record is defined in the [setupCustom \(customization\)](#) XSD.

Supported Operations

The following operations can be used with CRM Custom Field.

add | addList | delete | deleteList | get | getCustomization | getCustomizationId | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **setup customization.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Working with CRM Custom Fields Sublists

The SuiteTalk Schema Browser includes all sublists associated with the CRM Custom Fields record. See the following information for usage notes regarding specific CRM Custom Fields sublists. Usage notes are not provided for every sublist type.

CrmsCustomFieldFilter

This list can only be populated when the type of custom field is set to List/Record.

Entity Custom Field

The Entity Custom Field record is defined in the `setupCustom (customization) XSD`.

Supported Operations

The following operations can be used with Entity Custom Field.

add | addList | delete | deleteList | get | getCustomization | getCustomizationId | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **setup customization.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Item Custom Field

The Item Custom Field record is defined in the `setupCustom (customization) XSD`.

Supported Operations

The following operations can be used with Item Custom Field.

add | addList | delete | deleteList | get | getCustomization | getCustomizationId | getList | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **setup customization.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Item Number Custom Field

The Item Number Custom Field record is defined in the [setupCustom \(customization\) XSD](#).

Supported Operations

The following operations can be used with Item Number Custom Field.

add | addList | delete | deleteList | get | getCustomization | getCustomizationId | getList | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **setup customization.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Item Option Custom Field

The Item Option Custom Field record is defined in the [setupCustom \(customization\) XSD](#).

Supported Operations

The following operations can be used with Item Option Custom Field.

add | addList | delete | deleteList | get | getCustomization | getCustomizationId | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **setup customization.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Other Custom Field

The Other Custom Field record is defined in the `setupCustom (customization)` XSD.

Supported Operations

The following operations can be used with Other Custom Field.

add | addList | delete | deleteList | get | getCustomization | getCustomizationId | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See `setup customization.xsd` in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Transaction Body Custom Field

The Transaction Body Custom Field record is defined in the `setupCustom (customization)` XSD.

Supported Operations

The following operations can be used with Transaction Body Custom Field.

add | addList | delete | deleteList | get | getCustomization | getCustomizationId | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See `setup customization.xsd` in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Transaction Column Custom Field

The Transaction Column Custom Field record is defined in the `setupCustom (customization)` XSD.

Supported Operations

The following operations can be used with Transaction Column Custom Field.

add | addList | delete | deleteList | get | getCustomization | getCustomizationId | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **setup customization.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

The following sample shows how to set transaction column custom fields.

SOAP

```
<soapenv:Body>
<add xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
<record xsi:type="ns9:SalesOrder" xmlns:ns9="urn:sales_2_5.transactions.webservices.netsuite.com">
  <ns9:entity internalId="2" xsi:type="ns10:RecordRef"
xmlns:ns10="urn:core_2_5.platform.webservices.netsuite.com" />
<ns9:itemList replaceAll="false" xsi:type="ns9:SalesOrderItemList">
<ns9:item xsi:type="ns9:SalesOrderItem">
  <ns9:item internalId="34" xsi:type="ns11:RecordRef"
xmlns:ns11="urn:core_2_5.platform.webservices.netsuite.com" />
  <ns9:quantity xsi:type="xsd:double">3.0</ns9:quantity>
<ns9:customFieldList xsi:type="ns12:CustomFieldList"
xmlns:ns12="urn:core_2_5.platform.webservices.netsuite.com">
<ns12:customField internalId="custcol_my_bool" xsi:type="ns12:BooleanCustomFieldRef">
  <ns12:value xsi:type="xsd:boolean">true</ns12:value>
</ns12:customField>
<ns12:customField internalId="custcol_my_string" xsi:type="ns12:StringCustomFieldRef">
  <ns12:value xsi:type="xsd:string">hello world</ns12:value>
</ns12:customField>
<ns12:customField internalId="custcol_my_integer" xsi:type="ns12:LongCustomFieldRef">
  <ns12:value xsi:type="xsd:long">100</ns12:value>
</ns12:customField>
</ns9:customFieldList>
</ns9:item>
</ns9:itemList>
</record>
</add>
</soapenv:Body>
```

Java

```
SalesOrder so = new SalesOrder();

RecordRef entityRef = new RecordRef();
entityRef.setType(RecordType.customer);
entityRef.setInternalId("2");
so.setEntity(entityRef);

SalesOrderItem soi = new SalesOrderItem();
```

```
RecordRef itemRef = new RecordRef();
itemRef.setType(RecordType.inventoryItem);
itemRef.setInternalId("34");
soi.setItem(itemRef);
soi.setQuantity(new Double(3));

BooleanCustomFieldRef cf1 = new BooleanCustomFieldRef(true, "custcol_my_bool");
StringCustomFieldRef cf2 = new StringCustomFieldRef("hello world", "custcol_my_string");
LongCustomFieldRef cf3 = new LongCustomFieldRef(100, "custcol_my_integer");

soi.setCustomFieldList(new CustomFieldList(new CustomFieldRef[]{cf1, cf2, cf3}));
so.setItemList(new SalesOrderItemList(new SalesOrderItem[] {soi}, true));
```

Chapter 12 Marketing

The following marketing records are supported in SuiteTalk:

- Campaign
- Campaign Category
- Campaign Audience
- Campaign Family
- Campaign Search Engine
- Campaign Channel
- Campaign Offer
- Campaign Response
- Campaign Vertical
- Campaign Subscription
- Promotion Code

Campaign

The Campaign record is defined in the [listMkt \(marketing\)](#) XSD.

Supported Operations

The following operations can be used with Campaign records:

add | addList | attach / detach | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists marketing.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Campaign Category

The Campaign Category record is defined in the [listMkt \(marketing\)](#) XSD.

Supported Operations

The following operations can be used with Campaign Category records:

add | addList | delete | deleteList | get | getAll | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists marketing.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Campaign Audience

The Campaign Audience record is defined in the [listMkt \(marketing\) XSD](#).

Supported Operations

The following operations can be used with Campaign Audience records:

add | addList | delete | deleteList | get | getAll | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists marketing.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Campaign Family

The Campaign Family record is defined in the [listMkt \(marketing\) XSD](#).

Supported Operations

The following operations can be used with Campaign Family records:

add | addList | delete | deleteList | get | getAll | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists marketing.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Campaign Search Engine

The Campaign Search Engine record is defined in the [listMkt \(marketing\) XSD](#).

Supported Operations

The following operations can be used with Campaign Search Engine records:

add | addList | delete | deleteList | get | getAll | getList | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists marketing.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Campaign Channel

The Campaign Channel record is defined in the [listMkt \(marketing\) XSD](#).

Supported Operations

The following operations can be used with Campaign Channel records:

add | addList | delete | deleteList | get | getAll | getList | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists marketing.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Campaign Offer

The Campaign Offer record is defined in the [listMkt \(marketing\) XSD](#).

Supported Operations

The following operations can be used with Campaign Offer records:

add | addList | delete | deleteList | get | getAll | getList | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists marketing.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Campaign Response

The Campaign Response record is defined in [listMkt \(marketing\) XSD](#).

Note that when instantiating a Campaign Response record in PHP, you must use the name space prefix at the end of the directory - for example:

```
$newCampaignResponse = new nsComplexObject('listsmarketing:CampaignResponse');
```

For more information on the NetSuite PHP toolkit, see [Using Web Services with PHP](#) in the NetSuite Help Center.

Supported Operations

The following operations can be used with Campaign Response records:

add | addList | delete | deleteList | get | getAll | getList | getSelectValue | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **lists marketing.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

The following SOAP sample shows how to attach a campaign response to a contact. Be aware that you cannot attach campaign records to contacts, however you can attach campaign responses to contacts.

Note that you must set the **entity** field on CampaignResponse to reference the contact record.

```
<soapenv:Body>
  <platformMsgs:add
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:platformMsgs="urn:messages_2008_2.platform.webservices.netsuite.com"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com"
    xmlns:s0="urn:marketing_2008_2.lists.webservices.netsuite.com"
```

```
xmlns:listMktTyp="urn:types.marketing_2008_2.lists.webservices.netsuite.com"  
xmlns:s1="urn:sales_2008_2.transactions.webservices.netsuite.com">  
<platformMsgs:record xsi:type="s0:CampaignResponse">  
  <s0:entity internalId="8" type="contact" />  
  <s0:leadSource internalId="10202" />  
  <s0:response>_responded</s0:response>  
</platformMsgs:record>  
</platformMsgs:add>  
</soapenv:Body>  
</soapenv:Envelope>
```

Campaign Vertical

The Campaign Vertical record is defined in the [listMkt \(marketing\)](#) XSD.

Supported Operations

The following operations can be used with Campaign Vertical records:

add | addList | delete | deleteList | get | getAll | getList | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists marketing.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Campaign Subscription

The Campaign Subscription record is defined in the [listMkt \(marketing\)](#) XSD.

Supported Operations

The following operations can be used with Campaign Subscription records:

add | addList | delete | deleteList | get | getAll | getList | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists marketing.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Promotion Code

The Promotion Code record is defined in the [listMkt \(marketing\) XSD](#).

Supported Operations

The following operations can be used with Promotion Code records:

add | addList | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See [lists marketing.xsd](#) in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

The following sample shows how to add a promotion code record to NetSuite.

SOAP Request

```
<soapenv:Body>
  <platformMsgs:add
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:listMktTyp="urn:types.marketing_2008_2.lists.webservices.netsuite.com"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:platformMsgs="urn:messages_2008_2.platform.webservices.netsuite.com"
    xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:s0="urn:marketing_2008_2.lists.webservices.netsuite.com"
    xmlns:s1="urn:sales_2008_2.transactions.webservices.netsuite.com">
    <platformMsgs:record xsi:type="s0:PromotionCode">
      <s0:code>Any Promo Code</s0:code>
      <s0:isInactive>>false</s0:isInactive>
      <s0:discount internalId="28" />
      <s0:applyDiscountTo>_firstSaleOnly</s0:applyDiscountTo>
      <s0:startDate>2009-01-09T23:23:38.787Z</s0:startDate>
      <s0:endDate>2009-01-16T18:23:52.000Z</s0:endDate>
      <s0:isPublic>>true</s0:isPublic>
      <s0:excludeItems>>false</s0:excludeItems>
    </platformMsgs:record>
  </platformMsgs:add>
</soapenv:Body>
</soapenv:Envelope>
```

Java

```
public void addPromotionCode()throws RemoteException
{
    //This operation requires a valid session
```

```
this.login(true);

String readStr = "";
PromotionCode pc = new PromotionCode();

_console.writeln("\nPlease enter promotion name: ");
readStr = _console.readLine();
pc.setCode(readStr);

_console.writeln("\nPlease enter description : ");
readStr = _console.readLine();
pc.setDescription(readStr);

pc.setIsInactive(new Boolean(false));

_console.writeln("\nPlease enter the discount item internal ID: ");
readStr = _console.readLine();
RecordRef discount = new RecordRef();
discount.setType(RecordType.discountItem);
discount.setInternalId(readStr);
pc.setDiscount(discount);

//pc.setRate(String rate) //required if discount item not specified
//pc.setDiscountType(Boolean discountType) //required if discount item not specified
_console.writeln("Do you want to apply code to first sale only or all sales?");

while(true)
{
    _console.writeln("\nEnter 1 for First Sale Only, 2 for All Sales.");
    readStr = _console.readLine();
    PromotionCodeApplyDiscountTo applyDiscountTo = null;

    if(readStr.equalsIgnoreCase("1"))
    {
        applyDiscountTo = PromotionCodeApplyDiscountTo._firstSaleOnly;
        pc.setApplyDiscountTo(applyDiscountTo);
        break;
    }
    if(readStr.equalsIgnoreCase("2"))
    {
        applyDiscountTo = PromotionCodeApplyDiscountTo._allSales;
        pc.setApplyDiscountTo(applyDiscountTo);
        break;
    }
}

Calendar startDate = Calendar.getInstance();
Calendar endDate = Calendar.getInstance();

_console.writeln("Enter promotion code starting day.");
readStr = _console.readLine();
int day = Integer.parseInt(readStr);

_console.writeln("Enter promotion code starting month (1-12).");
readStr = _console.readLine();
```

```
int month = Integer.parseInt(readStr);

//January=0
startDate.set(2009, month-1, day);
pc.setStartDate(startDate);

_console.writeln("Enter promotion code ending day.");
readStr = _console.readLine();
day = Integer.parseInt(readStr);

_console.writeln("Enter promotion code ending month (1-12).");
readStr = _console.readLine();
month = Integer.parseInt(readStr);

endDate.set(2009, month-1, day);
pc.setEndDate(endDate);

pc.setIsPublic(new Boolean(true));
pc.setExcludeItems(new Boolean(false));

WriteResponse writeRes = _port.add(pc);
boolean success = writeRes.getStatus().isSuccess();
if (success)
{
    _console.writeln("\nPromotion Code created successfully.");
    RecordRef _ref = (RecordRef)(writeRes.getBaseRef());
    _console.writeln("Internal ID: " + _ref.getId());
}
else
{
    _console.error(getStatusDetails(writeRes.getStatus()));
}
}
```

Chapter 13 File Cabinet

The following file cabinet records are supported in SuiteTalk:

- File
- Folder

File

The File record is defined in the `docFileCab (fileCabinet)` XSD.

Supported Operations

The following operations can be used with the File record.

add | addList | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **documents fileCabinet.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Usage Notes

- You can return the file content only on a single **get** operation. When using either the **search** or **getList** operations, NetSuite returns only the file record (without the content).
- To add a file through Web services, you must always specify a file name, even if you are attaching from the Web. If you do not specify a file name, a `USER_ERROR` occurs. This behavior differs from file addition in the UI.
- The `bodyFieldsOnly` search preference has no bearing on file content in a search. For information on the `bodyFieldsOnly` search preference, see “Setting Search Preferences” in the *SuiteTalk (Web Services) Platform Guide* [Setting Search Preferences](#).
- To optimize storage, NetSuite does not log any `base64Binary` data in the Import log with the SOAP.

Sample 1

The following SOAP sample shows a simple request to get the contents of a particular file.

```

GET REQUEST:
<get xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
<baseRef internalId="207" type="file" xsi:type="ns1:RecordRef"
xmlns:ns1="urn:core_2_6.platform.webservices.netsuite.com"/>
</get>

```

Sample 2

The following SOAP samples show how to add different file types to the NetSuite file cabinet.

SOAP (adding a TEXT file)

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<platformMsgs:preferences
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:platformMsgs="urn:messages_2_6.platform.webservices.netsuite.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" />
</soapenv:Header>
<soapenv:Body>
<platformMsgs:add
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:platformMsgs="urn:messages_2_6.platform.webservices.netsuite.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:s0="urn:filecabinet_2_6.documents.webservices.netsuite.com">
<platformMsgs:record xsi:type="s0:File">
<s0:name>sample.txt</s0:name>
<s0:attachFrom>_computer</s0:attachFrom>
<s0:content>sample.txt</s0:content>
<s0:description>text</s0:description>
<s0:isOnline>>true</s0:isOnline>
<s0:isInactive>>false</s0:isInactive>
</platformMsgs:record>
</platformMsgs:add>
</soapenv:Body>
</soapenv:Envelope>

```

SOAP (adding an IMAGE file)

Be aware that after an image file is added to the NetSuite file cabinet, the file type will appear as “Other Binary File.”

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<platformMsgs:preferences
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:platformMsgs="urn:messages_2_6.platform.webservices.netsuite.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" />

```

```
</soapenv:Header>
<soapenv:Body>
  <platformMsgs:add
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:platformMsgs="urn:messages_2_6.platform.webservices.netsuite.com"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:s0="urn:filecabinet_2_6.documents.webservices.netsuite.com"
    xmlns:platformCore="urn:core_2_6.platform.webservices.netsuite.com">
    <platformMsgs:record xsi:type="s0:File">
      <platformCore:nullFieldList />
      <s0:name>wlogo2</s0:name>
      <s0:attachFrom>_computer</s0:attachFrom>
      <s0:content>C:\wlogo2.gif</s0:content>
      <s0:folder internalId="-4" type="folder" />
      <s0:description>wlogo2</s0:description>
      <s0:isInactive>>false</s0:isInactive>
    </platformMsgs:record>
  </platformMsgs:add>
</soapenv:Body>
</soapenv:Envelope>
```

Folder

The Folder record is defined in [docFileCab \(fileCabinet\) XSD](#).

Supported Operations

The following operations can be used to manipulate the Folder record.

add | addList | delete | deleteList | get | getDeleted | getList | getSavedSearch | getSelectValue | search | searchMore | searchNext | update | updateList

Field Definitions

The SuiteTalk Schema Browser includes definitions for all body fields, sublist fields, search filters, and search joins available to this record. See **documents fileCabinet.xsd** in the [SuiteTalk Schema Browser](#) for details.

Note: For information on using the SuiteTalk Schema Browser, see [Using the SuiteTalk Schema Browser](#).

Chapter 14 Country and State Enumerations

The following sections list all country enumerations defined in SuiteTalk, as well as explain how to set and validate state values in your Web services requests:

- [Country Enumerations](#)
- [Setting State Values in Web Services](#)

Important: Be aware that NetSuite uses country/state validation. This means that if the *Allow Free-Form States in Addresses* preference is set to `FALSE`, the state value is validated against the country. For example, you would not be able to set **CA** for **UK** since California is not a state that exists within the United Kingdom. For information on setting the *Allow Free-Form States in Addresses* preference, see [Setting the Allow Free-Form States in Addresses Preference](#).

As a general rule, enumerations are simply camel-cased concatenations of the country name as defined in the NetSuite UI.

Country Enumerations

The following table lists countries, as defined in NetSuite drop-down lists, and the corresponding enumeration to be used when populating these fields in Web services requests.

The Country Codes listed here are the ISO compliant country code string values returned when performing a search operation on entity records, whereas the Schema Enumeration is the corresponding value used during a request.

Country	Schema Enumeration	Country Code
Afghanistan	_afghanistan	AF
Aland Islands	_alandIslands	AX
Albania	_albania	AL
Algeria	_algeria	DZ
American Samoa	_americanSamoa	AS
Andorra	_andorra	AD
Angola	_angola	AO
Anguilla	_anguilla	AI
Antarctica	_antarctica	AQ
Antigua and Barbuda	_antiguaAndBarbuda	AG
Argentina	_argentina	AR

Country	Schema Enumeration	Country Code
Armenia	_armenia	AM
Aruba	_aruba	AW
Australia	_australia	AU
Austria	_austria	AT
Azerbaijan	_azerbaijan	AZ
Bahamas	_bahamas	BS
Bahrain	_bahrain	BH
Bangladesh	_bangladesh	BD
Barbados	_barbados	BB
Belarus	_belarus	BY
Belgium	_belgium	BE
Belize	_belize	BZ
Benin	_benin	BJ
Bermuda	_bermuda	BM
Bhutan	_bhutan	BT
Bolivia	_bolivia	BO
Bosnia and Herzegovina	_bosniaAndHerzegovina	BA
Botswana	_botswana	BW
Bouvet Island	_bouvetIsland	BV
Brazil	_brazil	BR
British Indian Ocean Territory	_britishIndianOceanTerritory	IO
Brunei Darussalam	_bruneiDarussalam	BN
Bulgaria	_bulgaria	BG
Burkina Faso	_burkinaFaso	BF
Burundi	_burundi	BI
Cambodia	_cambodia	KH
Cameroon	_cameroon	CM
Canada	_canada	CA
Canary Islands	_canaryIslands	IC
Cape Verde	_capeVerde	CV
Cayman Islands	_caymanIslands	KY
Central African Republic	_centralAfricanRepublic	CF
Ceuta and Melilla	_ceutaAndMelilla	EA
Chad	_chad	TD
Chile	_chile	CL
China	_china	CN

Country	Schema Enumeration	Country Code
Christmas Island	_christmasIsland	CX
Cocos (Keeling) Islands	_cocosKeelingsIslands	CC
Colombia	_colombia	CO
Comoros	_comoros	KM
Congo, Democratic People's Republic	_congoDemocraticPeoplesRepublic	CD
Congo, Republic of	_congoRepublicOf	CG
Cook Islands	_cookIslands	CK
Costa Rica	_costaRica	CR
Cote d'Ivoire	_coteDlvoire	CI
Croatia/Hrvatska	_croatiaHrvatska	HR
Cuba	_cuba	CU
Cyprus	_cyprus	CY
Czech Republic	_czechRepublic	CZ
Denmark	_denmark	DK
Djibouti	_djibouti	DJ
Domenica	_dominica	DM
Dominican Republic	_dominicanRepublic	DO
East Timor	_eastTimor	TP
Ecuador	_ecuador	EC
Egypt	_egypt	EG
El Salvador	_elSalvador	SV
Equatorial Guinea	_equatorialGuinea	GQ
Eritrea	_eritrea	ER
Estonia	_estonia	EE
Ethiopia	_ethiopia	ET
Falkland Islands (Malvina)	_falklandIslands	FK
Faroe Islands	_faroelslands	FO
Fiji	_fiji	FJ
Finland	_finland	FI
France	_france	FR
French Guiana	_frenchGuiana	GF
French Polynesia	_frenchPolynesia	PF
French Southern Territories	_frenchSouthernTerritories	TF
Gabon	_gabon	GA
Gambia	_gambia	GM
Georgia	_georgia	GE

Country	Schema Enumeration	Country Code
Germany	_germany	DE
Ghana	_ghana	GH
Gibraltar	_gibraltar	GI
Greece	_greece	GR
Greenland	_greenland	GL
Grenada	_grenada	GD
Guadeloupe	_guadeloupe	GP
Guam	_guam	GU
Guatemala	_guatemala	GT
Guernsey	_guernsey	GG
Guinea	_guinea	GN
Guinea-Bissau	_guineaBissau	GW
Guyana	_guyana	GY
Haiti	_haiti	HT
Heard and McDonald Is.	_heardAndMcDonaldIslands	HM
Holy See (City Vatican State)	_holySeeCityVaticanState	VA
Honduras	_honduras	HN
Hong Kong	_hongKong	HK
Hungary	_hungary	HU
Iceland	_iceland	IS
India	_india	IN
Indonesia	_indonesia	ID
Iran, Islamic Republic of	_iranIslamicRepublicOf	IR
Iraq	_iraq	IQ
Ireland	_ireland	IE
Isle of Man	_isleOfMan	IM
Israel	_israel	IL
Italy	_italy	IT
Jamaica	_jamaica	JM
Japan	_japan	JP
Jersey	_jersey	JE
Jordan	_jordan	JO
Kazakhstan	_kazakhstan	KZ
Kenya	_kenya	KE
Kiribati	_kiribati	KI
Korea, Democratic People's Republic	_koreaDemocraticPeoplesRepublic	KP

Country	Schema Enumeration	Country Code
Korea, Republic of	_koreaRepublicOf	KR
Kuwait	_kuwait	KW
Krgyzstan	_kyrgyzstan	KG
Lao, People's Democratic Republic	_laoPeoplesDemocraticRepublic	LA
Latvia	_latvia	LV
Lebanon	_lebanon	LB
Lesotho	_lesotho	LS
Liberia	_liberia	LR
Libyan Arab Jamahiriya	_libyanArabJamahiriya	LY
Liechtenstein	_liechtenstein	LI
Lithuania	_lithuania	LT
Luxembourg	_luxembourg	LU
Macau	_macau	MO
Macedonia	_macedonia	MK
Madagascar	_madagascar	MG
Malawi	_malawi	MW
Malaysia	_malaysia	MY
Maldives	_maldives	MV
Mali	_mali	ML
Malta	_malta	MT
Marshall Islands	_marshallIslands	MH
Martinique	_martinique	MQ
Mauritania	_mauritania	MR
Mauritius	_mauritius	MU
Mayotte	_mayotte	YT
Mexico	_mexico	MX
Micronesia, Federal State of	_micronesiaFederalStateOf	FM
Moldova, Republic of	_moldovaRepublicOf	MD
Monaco	_monaco	MC
Mongolia	_mongolia	MN
Montenegro	_montenegro	ME
Montserrat	_montserrat	MS
Morocco	_morocco	MA
Mozambique	_mozambique	MZ
Myanmar	_myanmar	MM
Namibia	_namibia	NA

Country	Schema Enumeration	Country Code
Nauru	_nauru	NR
Nepal	_nepal	NP
Netherlands	_netherlands	NL
Netherlands Antilles	_netherlandsAntilles	AN
New Caledonia	_newCaledonia	NC
New Zealand	_newZealand	NZ
Nicaragua	_nicaragua	NI
Niger	_niger	NE
Nigeria	_nigeria	NG
Niue	_niue	NU
Norfolk Island	_norfolkIsland	NF
Northern Mariana Islands	_northernMarianalands	MP
Norway	_norway	NO
Oman	_oman	OM
Pakistan	_pakistan	PK
Palau	_palau	PW
Palestinian Territories	_palestinianTerritories	PS
Panama	_panama	PA
Papua New Guinea	_papuaNewGuinea	PG
Paraguay	_paraguay	PY
Peru	_peru	PE
Philippines	_philippines	PH
Pitcairn Island	_pitcairnIsland	PN
Poland	_poland	PL
Portugal	_portugal	PT
Puerto Rico	_puertoRico	PR
Qatar	_qatar	QA
Reunion Island	_reunionIsland	RE
Romania	_romania	RO
Russian Federation	_russianFederation	RU
Rwanda	_rwanda	RW
Saint Barthelemy	_saintBarthelemy	BL
Saint Helena	_saintHelena	SH
Saint Kitts and Nevis	_saintKittsAndNevis	KN
Saint Lucia	_saintLucia	LC
Saint Martin	_saintMartin	MF

Country	Schema Enumeration	Country Code
Saint Vincent and the Grenadines	_saintVincentAndTheGrenadines	VC
Samoa (formerly Western Samoa)	_Samoa	WS
San Marino	_sanMarino	SM
Sao Tome and Principe	_saoTomeAndPrincipe	ST
Saudi Arabia	_saudiArabia	SA
Senegal	_senegal	SN
Serbia	_serbia	CS
Seychelles	_seychelles	SC
Sierra Leone	_sierraLeone	SL
Singapore	_singapore	SG
Slovak Republic	_slovakRepublic	SK
Slovenia	_slovenia	SI
Solomon Islands	_solomonIslands	SB
Somalia	_somalia	SO
South Africa	_southAfrica	ZA
South Georgia	_southGeorgia	GS
Spain	_spain	ES
Sri Lanka	_sriLanka	LK
St. Pierre and Miquelon	_stPierreAndMiquelon	PM
Sudan	_sudan	SD
Suriname	_suriname	SR
Svalbard and Jan Mayen Islands	_svalbardAndJanMayenIslands	SJ
Swaziland	_swaziland	SZ
Sweden	_sweden	SE
Switzerland	_switzerland	CH
Syrian Arab Republic	_syrianArabRepublic	SY
Taiwan	_taiwan	TW
Tajikistan	_tajikistan	TJ
Tanzania	_tanzania	TZ
Thailand	_thailand	TH
Togo	_togo	TG
Tokelau	_tokelau	TK
Tonga	_tonga	TO
Trinidad and Tobago	_trinidadAndTobago	TT
Tunisia	_tunisia	TN
Turkey	_turkey	TR

Country	Schema Enumeration	Country Code
Turkmenistan	_turkmenistan	TM
Turks and Caicos Islands	_turksAndCaicosIslands	TC
Tuvalu	_tuvalu	TV
Uganda	_uganda	UG
Ukraine	_ukraine	UA
United Arab Emirates	_unitedArabEmirates	AE
United Kingdom (GB)	_unitedKingdomGB	GB
United States	_unitedStates	US
Uruguay	_uruguay	UY
US Minor Outlying Islands	_uSMajorOutlyingIslands	UM
Uzbekistan	_uzbekistan	UZ
Vanuatu	_vanuatu	VU
Venezuela	_venezuela	VE
Vietnam	_vietnam	VN
Virgin Islands, British	_virginIslandsBritish	VG
Virgin Islands, USA	_virginIslandsUSA	VI
Wallis and Futuna Islands	_wallisAndFutunaIslands	WF
Western Sahara	_westernSahara	EH
Yemen	_yemen	YE
Zambia	_zambia	ZM
Zimbabwe	_zimbabwe	ZW

Setting State Values in Web Services

The WSDL does not include a State class or state enumeration values (previously defined in commonTypes.xsd in endpoints older than 2008.2). State fields now take any of the string values defined in the **Short Name** column on the State/Provinces/Countries page in the NetSuite UI (see figure).

- Navigate to this page by going to Setup > Company > States/Provinces/Countries.

Note that some short name values may contain characters that are not translated in some languages. You can retrieve a list of translated short names through Web services. However, in Web services, you cannot use the search operation to retrieve state values. You must use the **getAll** operation to retrieve all state values in the system. This operation will return **all** states, not just the legal ones for your default country. Also note that the country and state must match on the address.

Be aware that state values can be added/updated/deleted through the NetSuite UI and through Web services.

States/Provinces/Countries		
Edit	State	Short Name
	Alabama	AL
	Alaska	AK
	Arizona	AZ
	Arkansas	AR
	Armed Forces Europe	AE
	Armed Forces Pacific	AP
	California	CA
	Colorado	CO
	Connecticut	CT
	Delaware	DE
	District of Columbia	DC
	Florida	FL
	Georgia	GA
	Hawaii	HI
	Idaho	ID
	Illinois	IL
	Indiana	IN
	Iowa	IA
	Kansas	KS
	Kentucky	KY
	Louisiana	LA
	Maine	ME
	Maryland	MD
	Massachusetts	MA
	Michigan	MI
	Minnesota	MN
	Mississippi	MS

Country: United States

Print New

The Java and SOAP samples below show how to create a new customer and define all address properties on the customer Address (addressbook) sublist. Note that the state value is defined as the string "CA" — as it appears on the States/Provinces/Counties page in the UI.

Java

```
public void add_customer_with_address() throws Exception {
    Customer c = new Customer();
    c.setCompanyName("my company");
    CustomerAddressbook cab = new CustomerAddressbook();
    cab.setAddr1("123 Main St");
    cab.setCity("San Mateo");
    cab.setState("CA");
    cab.setZip("94403");
    cab.setCountry(Country._unitedStates);
    c.setAddressbookList(new CustomerAddressbookList(new
        CustomerAddressbook[]{cab}, false));
    sessMgr.getPort().add(c);
}
```

SOAP

```
<add xmlns="urn:messages_2008_2.platform.webservices.netsuite.com">
  <record externalId="Strawberry10150306" xsi:type="ns1:Customer"
    xmlns:ns1="urn:relationships_2008_2.lists.webservices.netsuite.com">
    <ns1:companyName xsi:type="xsd:string">my company</ns1:companyName>
    <ns1:addressbookList replaceAll="false" xsi:type="ns1:CustomerAddressbookList">
      <ns1:addressbook xsi:type="ns1:CustomerAddressbook">
        <ns1:addr1 xsi:type="xsd:string">123 Main St</ns1:addr1>
        <ns1:city xsi:type="xsd:string">San Mateo</ns1:city>
        <ns1:zip xsi:type="xsd:string">94403</ns1:zip>
        <ns1:country xsi:type="ns2:Country"
          xmlns:ns2="urn:types.common_2008_2.platform.
            webservices.netsuite.com">_unitedStates
          <ns1:country>
            <ns1:state xsi:type="xsd:string">CA</ns1:state>
          </ns1:country>
        </ns1:addressbook>
      </ns1:addressbookList>
    </record>
  </add>
```

Important: State enumeration changes made in the 2008_2 endpoint have no impact on WSDL versions 2008_1 and lower. However, users *will* have to change their code when they upgrade to the 2009_1 endpoint.

For example, when upgrading to the 2009_1 endpoint, existing code that reads something similar to:

```
address.state = State._california;
```

should be changed to:

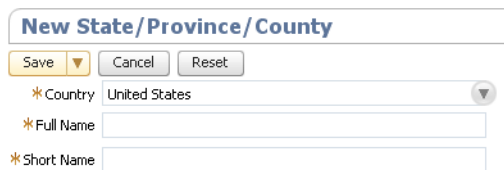
```
address.state("CA");
```

Creating Custom State Values

For countries that do not have country/state values built in to NetSuite (for example, Portugal or Argentina), you can create a list of custom state values and assign these values to a specific country. Doing so allows you to enforce country/state validation when the *Allow Free-Form States in Addresses* is set to FALSE.

Note: See [Setting the Allow Free-Form States in Addresses Preference](#) for more information on setting this preference and enabling or disabling country/state validation.

Create custom state values by going to Setup > Company > States/Provinces/Countries > New. In the Country drop-down, select your country. In the Full Name and Short Name fields, specify your state values. Note that you must use the **Short Name** value in your Web services requests.



Note: You can also click the New button at the bottom of the State/Province/Countries list page to create custom states.

Setting the Allow Free-Form States in Addresses Preference

Setting the *Allow Free-Form States in Addresses* to either TRUE or FALSE controls whether country/state validation is performed during a Web services request. In NetSuite, Navigate to this preference by going to Setup > Company > General Preferences.

To enable country/state validation:

1. Set the *Allow Free-Form States in Addresses* preference to FALSE (in other words, **do not select** the *Allow Free-Form States in Addresses* checkbox). When set to FALSE, the system validates the state short name (for example, CA or GA) against the country.
2. In your Web services request, you must then submit the Short Name state value that exists in the Setup > Company > States/Provinces/Countries list.
3. If the country/state values do not exist in the UI, extend this list by going to Setup > Company > State/Provinces/Countries > New to define custom states and associate them with a country. (See [Creating Custom State Values](#) for more information on creating custom state values.)

To disable country/state validation:

1. Set the *Allow Free-Form States in Addresses* preference to TRUE (in other words, **select** the *Allow Free-Form States in Addresses* checkbox).

2. In your Web services request, send any state value in the request. When this preference is set to TRUE, NetSuite stores your literal string.