SuiteTalk (Web Services) Platform Guide

November 12, 2014

Version 2014 Release 2

NETSUITE

Sample Code

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

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

Table of Contents

Chapter 1 SuiteTalk Platform Overview

The SuiteTalk Platform provides programmatic access to your NetSuite data and business processes through an XML-based application programming interface (API). Generally speaking, the SuiteTalk Platform has the following characteristics:

- **SOAP encoded Web services** – the SuiteTalk Platform uses SOAP-based Web services with document style, or Doc style, encoding.

  Doc style encoding consists of message-oriented exchanges where an XML schema specified within the SOAP message defines the structure of any messages sent between two applications. RPC exchanges are NOT supported.

- **HTTPS Transport** – currently the only transport supported is HTTPS as defined in the WSDL document.

For a list of NetSuite records that are supported in Web services development, see the help topic *Web Services Supported Records*. To see code samples of all SuiteTalk operations, see Web Services Operations.

Important: Some endpoints that pre-date 2014.2 are still supported. For documentation about these endpoints, see the help topic *SuiteTalk Archives*.

In this Guide

This manual contains the following sections:

SuiteTalk Platform Overview — provides a general overview of this guide, Web services and the NetSuite WSDL, and the NetSuite Web services governance model.

Getting Started — provides steps for getting your Web services development environment set up to create your first service. It also describes how to modify or set Web services preferences. This section also describes general concepts that pertains to SuiteTalk development.

Setting Web Services Preferences — provides information on setting company-wide preferences, request level preferences, search preferences, and setting the Internal ID preference.

Roles and Permissions in Web Services — defines the concept of a role within NetSuite and the necessity of providing a role ID during authentication. This section also describes how to assign a default role to a Web services user, as well as how to set a “Web Services Only” role for a user.

Records, Fields, Forms, and Sublists in Web Services — describes how to work with records, field, and sublist objects in Web services. This section also provides information on working with custom forms.

Web Services Processing — describes how to process requests synchronously versus asynchronously. Also provided are steps for monitoring your web services requests.

Web Services Security — describes all aspects of Web services security and session management.

Platform Features — describes the Web Services Concurrent License.

Types — describes the various types available in the SuiteTalk Platform.

Web Services Operations — describes each operation that can be performed through Web services and provides SOAP, C# and Java code samples.

Web Services Error Handling and Error Codes — provides tables of possible SOAP faults, fault status codes and error status codes that can be thrown for a given request.

Task IDs — provides a list of task IDs for NetSuite pages.

Support Has Ended for SuiteTalk 2009.2 and Earlier Endpoints

NetSuite no longer supports SuiteTalk (Web Services) 2009.2 or earlier endpoints. NetSuite strongly encourages customers and partners using 2009.2 and earlier versions of the SuiteTalk WSDL to begin upgrading to later endpoints.

The end of support for these endpoints is enabling NetSuite to focus maintenance on fewer versions, and will enable customers to benefit from more consistent record processing behavior in the current endpoints.

Determining Endpoints that are Currently in Use

Use the following guidelines to determine whether any of your SuiteTalk integrations use the 2009.2 or earlier endpoints:

- For each partner application, work with the partner directly to find out the endpoint in use.
- For your own integrations:
  - You can check SOAP requests to see which endpoints they reference. These are visible at Setup > Integration > Web Services Usage Log, in the Request column.
  - You can contact NetSuite Support to ask for help in determining the endpoints used in your account.

Note that different integrations in your account may use different endpoints.

Resources for Endpoint Upgrades

To begin the process of upgrading your endpoint, refer to the following resources.

- For the schema of the latest endpoint, see:

  https://webservices.netsuite.com/wsdl/v2014_2_0/netsuite.wsdl

- For details about the records supported in the latest endpoint, see the help topic *SuiteTalk Records Overview*.
- For general information about using SuiteTalk, see SuiteTalk Platform Overview.
- For upgrade tips, see Upgrading WSDL Versions.
- If you have issues during your transition process, please contact NetSuite Technical Support.

Note About Record Processing Behavior in 2010.1 and Later Endpoints

Be aware of the following change: When modifying records using the 2009.2 or an earlier endpoint, most, but not all, of NetSuite's core business logic is executed. For integrations that use an endpoint later than 2009.2, *all* core business logic executes when the record is submitted. So you may notice some changes in SuiteTalk behavior for integrations that are upgraded to use an endpoint later than 2009.2. Following your upgrade, the behavior will more closely match the behavior when records are modified through the NetSuite user interface.

Understanding Web Services

Web services are Extensible Markup Language (XML) applications mapped to programs, objects, databases or complex business functions. They utilize a standardized XML messaging system to send or receive requests to authorized parties over the Internet. Businesses can implement Web services to provide standards based processes that can be utilized by other organizations or integrated with business partner processes. Since the programming logic encapsulated by each Web service is independent of any one platform or technology set, Web services provide an independence and flexibility for integration across and between businesses.

The following protocols are used to publish, expose or access Web Services:

- WSDL (Web Services Description Language): a WSDL file exposes a Web service interface, interaction patterns and protocol mapping. WSDL can be readily interpreted by other applications, systems and platforms.
- UDDI (Universal Description, Discovery and Integration): A Web Service can be categorized and registered in a UDDI Registry so that applications can locate it and retrieve its WSDL. Currently NetSuite does NOT provide a UDDI registry.

- SOAP (Simple Object Access Protocol): A Web Service can use the SOAP messaging format to define an envelope for Web services communication via HTTP, HTTPs or other transport layers.

With the SuiteTalk Platform, you can integrate existing business functions within your organization to NetSuite through the use of Web services. Web services can be invoked in real-time to perform operations such as retrieving, adding, updating, and deleting data.

Understanding the NetSuite WSDL and XSD Structure

The SuiteTalk Platform has a single WSDL file that describes all supported operations and messages. You can access that file here —

https://webservices.netsuite.com/wsdl/v2014_2_0/netsuite.wsdl

— where v2014_2_0 reflects the WSDL version.

NetSuite defines WSDL versioning, the location of schemas, namespaces, and the endpoint as follows:

```
WSDL: https://webservices.netsuite.com/wsdl/v2014_2_0/netsuite.wsdl
   <xsd:import namespace="urn:core_2014_2_0.platform.webservices.netsuite.com"
  schemaLocation="https://webservices.netsuite.com/xsd/platform/v2014_2_0/core.xsd"/>
<port name="NetSuitePort" binding="tns:NetSuiteBinding">
 <soap:address location="https://webservices.netsuite.com/servicesNetSuitePort_2014_2_0" />
</port>
```

Note: For more information on WSDL versioning, upgrading, and testing, see NetSuite WSDL Versioning. For information on WSDLs that pre-date 2014.2, see the help topic *SuiteTalk Archives*.

The WSDL is composed of numerous NetSuite-specific types that are defined in related XSDs. Each XSD URL has an alias that can be used as a reference to the corresponding XSD file. The following tables show the organization of the XSD files.

Important: Since September 2012, when NetSuite began hosting new customer accounts in multiple data centers, the URL to be used for Web services access depends upon the data center hosting each NetSuite account. For example, it could be webservices.netsuite.com or webservices.na1.netsuite.com. In the 2012.2 and later endpoints, you can use the getDataCenterUrls operation to dynamically discover the correct URL for Web services access to a NetSuite account. A REST service is also available for this purpose. See the help topic *Using the REST roles Service to Get User Accounts, Roles, and Domains*.

NetSuite Messaging XSD Files

These files provide descriptions for the base Web services functions used by all operations. For API documentation on each operation, see Web Services Operations.

URL	Schema Alias	Notes
https://webservices.netsuite.com/xsd/platform/v2014_2_0/common.xsd	platformCommon	This file also includes descriptions for the following subrecords: • Address • InventoryDetail • LandedCost
https://webservices.netsuite.com/xsd/platform/v2014_2_0/core.xsd	platformCore	
https://webservices.netsuite.com/xsd/platform/v2014_2_0/faults.xsd	platformFaults	
https://webservices.netsuite.com/xsd/platform/v2014_2_0/messages.xsd	platformMsgs	

NetSuite Business Records XSD Files

These files provide descriptions for each record type in NetSuite. For field reference information on each record, see the help topic *SuiteTalk Records Overview*. For general information on working with records in SuiteTalk, see Working with Records in Web Services.

URL	Schema Alias	Record Types
https://webservices.netsuite.com/xsd/activities/v2014_2_0/scheduling.xsd	actSched	ContactCategory CustomerCategory SalesRole PriceLevel WinLossReason Term NoteType PaymentMethod CalendarEvent CalendarEventSearch CalendarEventSearchAdvanced Task TaskSearch TaskSearchAdvanced PhoneCall PhoneCallSearch PhoneCallSearchAdvanced ProjectTask Resource Allocation
https://webservices.netsuite.com/xsd/transactions/v2014_2_0/demandPlanning.xsd	demandPlanning	ItemDemandPlan ItemSupplyPlan
https://webservices.netsuite.com/xsd/documents/v2014_2_0/fileCabinet.xsd	docfileCab	File FileSearch FileSearchAdvanced Folder

URL	Schema Alias	Record Types
		FolderSearch FolderSearchAdvanced
https://webservices.netsuite.com/xsd/general/v2014_2_0/communication.xsd	generalComm	Note NoteSearch NoteSearchAdvanced Message MessageSearch MessageSearchAdvanced
https://webservices.netsuite.com/xsd/lists/v2014_2_0/accounting.xsd	listAcct	Billing Schedule ContactCategory CustomerCategory SalesRole PriceLevel WinLossReason Term NoteType PaymentMethod LeadSource Assembly Item (BOM Item) DescriptionItem DiscountItem Download Item GiftCertificateItem InventoryItem ItemSearch ItemSearchAdvanced Kit/Package Item Lot Numbered Assembly Item Lot Numbered Inventory Item MarkupItem NonInventoryPurchaseItem NonInventoryResaleItem NonInventorySaleItem OtherChargePurchaseItem OtherChargeResaleItem OtherChargeSaleItem PaymentItem SerializedAssemblyItem Serialized Inventory Item ServicePurchaseItem ServiceResaleItem ServiceSaleItem SubtotalItem Currency ExpenseCategory Account AccountSearch AccountSearchAdvanced AccountingPeriod Bin BinSearch

URL	Schema Alias	Record Types
		BinSearchAdvanced Classification ClassificationSearch ClassificationSearchAdvanced Department DepartmentSearch DepartmentSearchAdvanced Expense Category Gift Certificate
https://webservices.netsuite.com/xsd/lists/v2014_2_0/accounting.xsd	listAcct	GiftCertificateSearch GiftCertificateSearchAdvanced GlobalAccountMapping Inventory Number ItemAccountMapping Location LocationSearch LocationSearchAdvanced Nexus Revenue Recognition Schedule Revenue Recognition Template Sales Tax ItemItemSearchAdvanced ContactRole Bin SalesTaxItem TaxAcct TaxGroup TaxType Subsidiary SubsidiarySearch SubsidiarySearchAdvanced UnitsType PartnerCategory VendorCategory
https://webservices.netsuite.com/xsd/lists/v2014_2_0/employees.xsd	listEmp	Employee EmployeeSearch EmployeeSearchAdvanced
https://webservices.netsuite.com/xsd/lists/v2014_2_0/marketing.xsd	listMkt	Campaign CampaignSearch CampaignSearchAdvanced CampaignCategory CampaignAudience CampaignFamily CampaignSearchEngine CampaignChannel CampaignOffer CampaignResponse CampaignVertical CampaignSubscription PromotionCode PromotionCodeSearch

URL	Schema Alias	Record Types
		PromotionCodeSearchAdvanced
https://webservices.netsuite.com/xsd/lists/v2014_2_0/relationshipTypes.xsd	listRel	Contact ContactSearch ContactSearchAdvanced Customer CustomerSearch CustomerSearchAdvanced CustomerStatus Partner PartnerSearch PartnerSearchAdvanced Vendor VendorSearch VendorSearchAdvanced EntityGroup EntityGroupSearch EntityGroupSearchAdvanced Job JobSearch JobSearchAdvanced JobType JobStatus
https://webservices.netsuite.com/xsd/lists/v2014_2_0/supplyChain.xsd	listScm	ManufacturingCostTemplate ManufacturingOperationTask ManufacturingRouting
https://webservices.netsuite.com/xsd/lists/v2014_2_0/website.xsd	listSite	SiteCategory SiteCategorySearch SiteCategorySearchAdvanced
https://webservices.netsuite.com/xsd/lists/v2014_2_0/support.xsd	listSupport	SupportCase SupportCaseSearch SupportCaseSearchAdvanced SupportCaseStatus SupportCaseType SupportCaseOrigin SupportCaseIssue SupportCasePriority Solution SolutionSearch SolutionSearchAdvanced Topic TopicSearch TopicSearchAdvanced Issue IssueSearch IssueSearchAdvanced
https://webservices.netsuite.com/xsd/setup/v2014_2_0/customization.xsd	setupCustom	CustomRecord CustomRecordSearch CustomRecordSearchAdvanced CustomList

URL	Schema Alias	Record Types
		CustomRecordType EntityCustomField CrmCustomField OtherCustomField ItemCustomField TransactionBodyCustomField TransactionColumnCustomField ItemOptionCustomField CustomRecordCustomField
https://webservices.netsuite.com/xsd/transactions/v2014_2_0/bank.xsd	tranBank	Check Deposit
https://webservices.netsuite.com/xsd/transactions/v2014_2_0/customers.xsd	tranCust	CashRefund Charge CustomerPayment ReturnAuthorization CreditMemo CustomerRefund CustomerDeposit DepositApplication
https://webservices.netsuite.com/xsd/transactions/v2014_2_0/inventory.xsd	tranInvt	InventoryAdjustment InventoryCostRevaluation AssemblyBuild AssemblyUnbuild WorkOrder WorkOrderClose WorkOrderIssue WorkOrderCompletionOperation
https://webservices.netsuite.com/xsd/transactions/v2014_2_0/employees.xsd	tranEmp	TimeBill TimeBillSearch TimeBillSearchAdvanced ExpenseReport TimeEntry Timesheet
https://webservices.netsuite.com/xsd/transactions/v2014_2_0/financial.xsd	tranFin	Budget BudgetSearch BudgetSearchAdvanced
https://webservices.netsuite.com/xsd/transactions/v2014_2_0/general.xsd	tranGeneral	JournalEntry InterCompanyJournalEntry StatisticalJournalEntry
https://webservices.netsuite.com/xsd/transactions/v2014_2_0/purchases.xsd	tranPurch	VendorBill VendorPayment PurchaseOrder ItemReceipt

URL	Schema Alias	Record Types
https://webservices.netsuite.com/xsd/transactions/v2014_2_0/sales.xsd	tranSales	AccountingTransactionSearch AccountingTransactionSearchAdvanced Opportunity OpportunitySearch OpportunitySearchAdvanced SalesOrder TransactionSearch TransactionSearchAdvanced ItemFulfillment Invoice CashSale Estimate

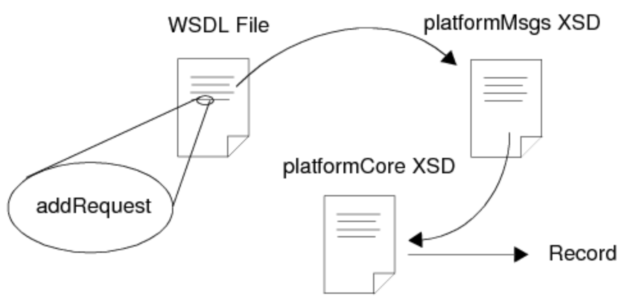
System Constants XSD Files

These files provide constant values for the corresponding types in the business records XSD files.

URL	Schema Alias
https://webservices.netsuite.com/xsd/activities/v2014_2_0/schedulingTypes.xsd	actSchedTyp
https://webservices.netsuite.com/xsd/transactions/v2014_2_0/demandPlanning.xsd	demandPlanningTyp
https://webservices.netsuite.com/xsd/documents/v2014_2_0/fileCabinetTypes.xsd	docFileCabTyp
https://webservices.netsuite.com/xsd/general/v2014_2_0/communicationTypes.xsd	generalCommTyp
https://webservices.netsuite.com/xsd/transactions/v2014_2_0/inventoryTypes.xsd	invtTyp
https://webservices.netsuite.com/xsd/lists/v2014_2_0/accountingTypes.xsd	listAcctTyp
https://webservices.netsuite.com/xsd/lists/v2014_2_0/employeeTypes.xsd	listEmpTyp
https://webservices.netsuite.com/xsd/lists/v2014_2_0/marketingTypes.xsd	listMktTyp
https://webservices.netsuite.com/xsd/lists/v2014_2_0/relationshipTypes.xsd	listRelTyp
https://webservices.netsuite.com/xsd/lists/v2014_2_0/supplyChainTypes.xsd	listScmTyp
https://webservices.netsuite.com/xsd/lists/v2014_2_0/supportTypes.xsd	listSupportTyp
https://webservices.netsuite.com/xsd/platform/v2014_2_0/coreTypes.xsd	platformCoreTyp
https://webservices.netsuite.com/xsd/platform/v2014_2_0/faultTypes.xsd	platformFaultsTyp
https://webservices.netsuite.com/xsd/setup/v2014_2_0/customizationTypes.xsd	setupCustomTyp
https://webservices.netsuite.com/xsd/transactions/v2014_2_0/saleTypes.xsd	tranSalesTyp

Example

For example, the addRequest message type has three levels of referencing.



In the WSDL file, the **addRequest** message is defined as:

```
<message name="addRequest">
 <part name="parameters" element="platformMsgs:add"/>
</message>
```

The element called **platformMsgs:add** is defined in the platformMsgs XSD file. In this case, the platformMsgs alias refers to the xsd file at:

https://webservices.netsuite.com/xsd/platform/v2014_2_0/messages.xsd

In this file, the addRequest element is defined again as:

```
<complexType name="AddRequest">
 <sequence>
   <element name="record" type="platformCore:Record" />
  </sequence>
</complexType>
```

Again there is a reference that is not contained in this XSD file called **platformCore:Record**. The platformCore alias refers to the XSD file at:

https://webservices.netsuite.com/xsd/platform/v2014_2_0/core.xsd

The abstract type Record is defined as:

```
<complexType name="Record" abstract="true">
 <sequence>
  <element name="nullFieldList" type="platformCore:NullField" minOccurs="0" maxOccurs="1" />
 </sequence>
</complexType>
```

Note: In the SuiteTalk Platform, Record is the base for all other NetSuite record types.

NetSuite WSDL Versioning

The following topics provide information on NetSuite WSDL versioning, upgrading to a new WSDL version, and things to consider when your NetSuite account is upgraded. If you are new to SuiteTalk development, it is recommended that you read these topics in order:

- Understanding NetSuite Versioning and WSDL Versioning
- Upgrading WSDL Versions
- Testing After a NetSuite Upgrade
- Supporting Existing WSDL Versions

Understanding NetSuite Versioning and WSDL Versioning

The naming convention for NetSuite versions and WSDL versions are the same. For example, for Version 2014 Release 2, there is an accompanying 2014_2 version of the WSDL, as shown in the following WSDL URL:

https://webservices.netsuite.com/wsdl/v2014_2_0/netsuite.wsdl

NetSuite WSDL versioning denotes:

- the major version: **v2014_2**
- the minor or patch version: **_0**

Generally speaking, the WSDL version that accompanies a NetSuite release incorporates much of the new functionality offered in the NetSuite upgrade. For example, if new workflows or new records are introduced in a NetSuite upgrade, the accompanying WSDL often includes new operations or records to support the NetSuite enhancements. In some releases, a new WSDL also includes enhancements to the SuiteTalk platform itself. These enhancements are independent of NetSuite product enhancements.

Be aware that when your NetSuite account is upgraded, **you do not need to upgrade** your WSDL to the supporting NetSuite version. For example, you may still use version 2014.1 of the WSDL in conjunction with a NetSuite account that has been upgraded to Version 2014 Release 2. For more information, see Upgrading WSDL Versions.

With each new release of NetSuite (and new WSDL version), it is recommended that you read the SuiteTalk portion of the Sneak Peeks and the Release Notes that come with the release. These documents list all of the new functionality and any schema bug fixes offered in the latest WSDL.

Upgrading WSDL Versions

NetSuite customers typically upgrade their WSDL when a newer version includes functionality that enables them to meet a particular business need. They will also upgrade their WSDL when the one they are currently using is going to be retired by NetSuite (see Supporting Existing WSDL Versions for details). If you do plan to upgrade the WSDL you are using, NetSuite strongly recommends that you upgrade to the latest available version to prolong the life cycle for your integrated application.

Important: You are not required to upgrade your WSDL when your NetSuite account is upgraded. However, when your account is upgraded, you should re-test all existing integrations to ensure they work against the latest version of NetSuite (see Testing After a NetSuite Upgrade for details).

After upgrading your WSDL version, any API incompatibilities between the new and old versions will be displayed in your development environment. You will be unable to proceed to unit testing until you have fixed any IDE compilation errors. Consequently, most WSDL upgrade issues are usually addressed during development.

Upgrading your WSDL version may require you to re-code parts of your integrated application to be compatible with the latest version of the NetSuite product. Re-coding may be due to any of the following reasons:

- a new required field has been added to NetSuite
- a field that was optional in previous versions of NetSuite becomes required in the latest version
- a field that is referenced in your integration has been removed from NetSuite
- the data type for a field has changed
- the method signature changes for a SuiteTalk operation that is used in your integration

Note: See Working with the Araxis Merge Diff File for information on how to use Araxis to see the changes made from one WSDL version to the next. You can also get an idea of the types of changes you need to make by reading the release notes from the endpoint you are currently using onwards to the last release of the WSDL.

Testing After a WSDL Upgrade

If you are upgrading your WSDL to make use of a particular NetSuite enhancement, it is recommended that you use the UI to learn how the feature works. In most cases, NetSuite Web services is the programmatic equivalent of the NetSuite UI. Therefore, if you know how a feature works in the UI, your learning curve for understanding how to leverage the feature in Web services will be minimized.

Be aware that WSDL upgrades may take considerable planning and testing. Consequently, you may decide to upgrade only when you know your version is going to be retired. (See Supporting Existing WSDL Versions for more information.)

Testing After a NetSuite Upgrade

NetSuite strives to maintain backward compatibility from one NetSuite version to the next. However, your integration may rely on unique workflows that might break when your NetSuite account is upgraded. Therefore, NetSuite recommends that you test all existing integrations during the **Release Preview** phase of a NetSuite upgrade and report any issues you may find to ensure a smooth upgrade of your production environment.

During the beta phase, you will need to modify the WSDL references you are using in your application to point to one of the following endpoints (based on the endpoint you are currently using):

- https://webservices.na1.beta.netsuite.com/wsdl/v2014_1_0/netsuite.wsdl
- https://webservices.na1.beta.netsuite.com/wsdl/v2013_2_0/netsuite.wsdl
- https://webservices.na1.beta.netsuite.com/wsdl/v2013_1_0/netsuite.wsdl
- https://webservices.na1.beta.netsuite.com/wsdl/v2012_2_0/netsuite.wsdl
- https://webservices.na1.beta.netsuite.com/wsdl/v2012_1_0/netsuite.wsdl
- https://webservices.na1.beta.netsuite.com/wsdl/v2011_2_0/netsuite.wsdl
- https://webservices.na1.beta.netsuite.com/wsdl/v2011_1_0/netsuite.wsdl
- https://webservices.na1.beta.netsuite.com/wsdl/v2010_2_0/netsuite.wsdl
- https://webservices.na1.beta.netsuite.com/wsdl/v2010_1_0/netsuite.wsdl

Note: If you are engaged with a partner for any of your Web services integrations, it is recommended that you contact them and inform them of when you are going on beta. They can then set up a time with you and a test plan for the integration.

Working with the Araxis Merge Diff File

To help you see the differences between the latest WSDL and the version that preceded it, refer to the Araxis diff file provided in the release notes for each release. When reviewing the diff file, start by investigating the records you are currently using. For example, if your Web services application interacts with customer and vendor records, use the diff file to review any changes made to these records. These changes may include added or removed fields or field (element) data type changes.

If you want to compare the latest WSDL with the version you are currently using, you can download Araxis and compare your own version to any other version listed on the Developer Portal.

For more information on using the Araxis diff file to review NetSuite WSDL changes, see the help topic *Using the Araxis Diff File* in the Version 2014 Release 1 Release Notes.

Supporting Existing WSDL Versions

NetSuite WSDLs are supported for up to three years from the date of first release. After this time, NetSuite recommends that users upgrade to the latest WSDL version. To find the latest version, see the help topic *SuiteTalk Archives*.

Note that if you do not upgrade when your WSDL is retired, your integrations may cease to run as intended and the endpoint for that version may be shut down.

Understanding Web Services Governance

In order to optimize NetSuite application and database servers, we have implemented a number of mechanisms to control the consumption of Web services. These mechanisms ensure the following:

- Requests are monitored and controlled to ensure that the user experience is not excessively impacted.
- The burden of heavy Web services users is not *shared* among all users.

NetSuite Web services governance includes:

- Record Limiting (see Understanding Record Limiting)
- Request Limiting (see Understanding Request Limiting)

Understanding Record Limiting

An individual request is rejected if it exceeds a certain size. The allowed request size varies depending on several factors, including the time of day the request is submitted, and whether it is sent asynchronously or synchronously.

Important: The record limits provided in the following tables are on a **per request** basis. There is NO limit to the number of requests that can be sent within a given time period, only on the number of records sent in an individual request.

Synchronous Operations

Note: Peak hours are considered 6am to 6pm PST, Monday through Friday.

Peak Hours

Operation (on a per request basis)	Record Count
Add	100
Update	50
Delete	100
Search Page Size	500

Off-Peak Hours

Operation (on a per request basis)	Record Count
Add	200
Update	100
Delete	200
Search Page Size	1000

Asynchronous Operations

Record limits for asynchronous requests are the same regardless of whether they are sent during peak or off-peak hours.

Operation (on a per request basis)	Record Count
Add	400
Update	200
Delete	400
Search Page Size	2000

Understanding Request Limiting

Request limiting pertains to the size of your SOAP request. The size of a request cannot exceed 100MB.

Understanding Governance Errors

The following faults are thrown as a result of other governance violations.

- **ExceededRecordCountFault** – thrown if a request exceeds the allowed record count (see Understanding Record Limiting).
- **ExceededRequestLimitFault** – thrown if the allowed number of concurrent requests is exceeded (see Understanding Request Limiting).
- **ExceededRequestSizeFault** – thrown if a request exceeds 100M (see Understanding Request Limiting).

For more information on exceptions, refer to Web Services Error Handling and Error Codes.

Vocabulary

You should be familiar with the following terms before implementing SuiteTalk integration technology.

- **Type** – A type is an element defined in the XML Schema. See Types for more details.
- **Record Type** – A NetSuite business record type, such as customer, event, or custom record, that has an entry form in the user interface. See Working with Records in Web Services for more details.
- **Search Record Type –** A type that encapsulates the available search criteria for a given NetSuite business record type. See Working with Records in Web Services for more details.
- **System** – The NetSuite application.
- **Client** – A company with a NetSuite account that is using Web services.
- **Requester** – The sender of a Web service request.
- **Write operations** – A Web services operation that changes data within a NetSuite account. These include add, addList, update, updateList, delete, and deleteList operations. See Web Services Operations Web Services Operations for information on each operation.
- **Read operations** – A Web services operation that retrieves data from a NetSuite account. These include get, getList, getAll, search, searchNext and searchMore operations. See Web Services Operations Web Services Operations for information on each operation.

Chapter 2 Getting Started

These sections provide the basic information you need to get started with SuiteTalk.

- Quick Start — provides step-by-step instructions on how to set up your Web services environment and start building applications to interface with SuiteTalk. Also provides a path for Downloading Sample Applications provided by NetSuite.
- SuiteTalk Development Considerations — describes development considerations such as account configurations and the enabling of NetSuite features to ensure your Web services calls execute successfully.
- Using the Web Services PHP Toolkit — provides setup information for those wanting to use NetSuite's PHP toolkit to develop Web services applications.

Before building a Web services application to interface with SuiteTalk, the NetSuite Web services feature must be enabled in your NetSuite account. See Enabling the Web Services Feature.

Once you have enabled Web services in your NetSuite account, you must set your Web services preferences. See Setting Web Services Preferences.

Quick Start

This section provides details on using Microsoft .NET or Java to build an application that uses NetSuite Web services. This section provides steps for the following tasks:

- Enabling the Web Services Feature
- Enabling the Show Internal IDs Preference
- Building an Application with Microsoft .NET
- Building an Application with Java using Apache Axis
- Downloading Sample Applications

Note the following:

- Before developing your own SuiteTalk applications, NetSuite strongly recommends you see the topic SuiteTalk Development Considerations.
- For information on working with SuiteTalk and PHP, see Using the Web Services PHP Toolkit.
- Although some customers have built SuiteTalk applications using Perl, currently Perl is not offically supported by NetSuite.

Enabling the Web Services Feature

The Web services feature must be enabled prior to submitting Web services requests.

To enable the Web services feature:

1. As administrator, click Setup > Company > Enable Features.
2. Click the **SuiteCloud** subtab.
3. Check the **Web Services** box.
4. Click **Save**.

Note that once you have enabled Web services in your NetSuite account, you should set your Web services preferences. See Setting Web Services Preferences for more details.

Enabling the Show Internal IDs Preference

NetSuite recommends that you enable the **Show Internal IDs** preference when working with Web services or SuiteScript. Enabling this preference lets you see the internal IDs for all fields, records, lists, and custom forms in NetSuite. In both Web services (and in SuiteScript) you will reference many of these IDs in your code.

To enable this preference, go to Home > Set Preferences, and on the General subtab, in the Defaults section, check the Show Internal IDs box.

When this preference is enabled:

- You can view the internal ID for a field by clicking on that field's label to open the field level help popup window. The internal ID is displayed in the lower right corner of this window.
- You can view the internal ID for a record or a custom field in an Internal ID column that displays on a list or search results page for that type of record or custom field.

For more details, see the help topic *Showing Record and Field IDs in Your Account*.

Building an Application with Microsoft .NET

This section provides details on how to use the Microsoft .NET platform to build a SuiteTalk application. You can also see Downloading Sample Applications for a .NET sample currently available on the NetSuite Developer Portal.

Important: When building an application with Microsoft .NET, NetSuite recommends that you use Visual Studio .NET since it provides an integrated development environment to build and debug your Web service applications. Alternatively, you can download and install the Microsoft .NET Framework SDK. However, this only provides you with the SDK and does NOT provide an integrated IDE.

All code samples in this section use the C# language, however these steps are similar for all other languages supported by Microsoft .NET. These languages include Visual Basic, Visual J#, and C++.

To use Microsoft .NET with NetSuite Web services:

1. Install the Microsoft .NET framework.

   Install Microsoft Visual Studio .NET, version 2003 or higher, which includes the .NET framework SDK (RECOMMENDED), or the Microsoft .NET Framework SDK version 1.1 (NOT recommended).

2. Use the .NET framework SDK to automatically generate the client proxy code

   If using Visual Studio .NET (RECOMMENDED):

   1. Start Microsoft Visual Studio .NET.
   2. Create a new project and choose a template, for example, Console Application.
   3. Once the project has been created, add a Web reference.

      The Add Web Reference option may be available in the **Project** menu. If not, select **Add Service Reference**, click the **Advanced** button, and click the **Add Web Reference** button.

   4. In the **Add Web Reference** dialog, enter the SuiteTalk WSDL URL and click the **Go** icon.

      A URL for the latest SuiteTalk WSDL is: https://webservices.netsuite.com/wsdl/v2014_2_0/netsuite.wsdl

      Important: The URL for the WSDL may vary from the example shown above. See Understanding the Effects of Accounts' Host Data Center on URLs for an important note about these URLs.

      Note: Although the material in this guide pertains to NetSuite's latest WSDL, NetSuite continues to support endpoints released prior to this version.

   5. Visual Studio inspects the WSDL and displays a summary of the available operations. If security warnings are displayed, click **Yes** as many times as necessary to continue this process.

6. Once the summary is displayed, click the **Add Reference** button to generate the classes.

   When this process is complete, com.netsuite.webservices is listed under **Web References** in the Solution Explorer.

7. To view all generated proxy classes, do one of the following:

   - Enable the **Show All Files** option in the Project menu
   - (If the above option is not available), right-click the com.netsuite.webservices listing in the Solution Explorer and choose **View in Object Browser**.

If using only the Microsoft .NET Framework SDK (NOT recommended):

1. Locate the wsdl.exe file under the Microsoft .NET Framework SDK installation and add it to your system path.

2. Open a command prompt, and type the following to generate the proxy classes:

```
wsdl /language:<language> <url>
```

   Where <language> is your preferred language and <url> is the URL for the NetSuite WSDL.

   For example, generate the proxy classes in C# as follows:

```
C:\project>wsdl /language:cs https://webservices.netsuite.com/wsdl/
v2012_1_0/netsuite.wsdl
```

   A C# file called NetSuiteService.cs is generated.

3. Compile the source code for your proxy into a .NET Assembly using the .NET Framework C# compiler or any other supported compiler.

4. Locate the csc.exe file under the Microsoft .NET Framework SDK installation and add it to your system path.

```
csc.exe /out: NetSuiteService.dll /target:library
reference:system.xml.serialization.dll /
reference:system.web.services.dll
NetSuiteService.cs
```

3. Implement your application by writing your business logic using the generated .NET proxy classes.

   Note that the following code snippets are shown in C#.

1. Allow the use of objects from the newly created namespace without having to qualify them.

```
using <namespace>.com.netsuite.webservices;
```

2. Instantiate the NetSuite service.

```
NetSuiteService service = new NetSuiteService();
```

3. Enable support for multiple cookie management.

```
service.CookieContainer = new CookieContainer();
```

4. Create a valid session by populating the Passport object.

```
//invoke the login operation
Passport passport = new Passport();
passport.account = "TSTDRV96";
passport.email = "username@netsuite.com";
RecordRef role = new RecordRef();
role.id = "3";
passport.role = role;
passport.password = "mypassword";
Status status = service.login( passport ).status;
```

5. Implement your business logic. For example, create a new customer in NetSuite.

```
Customer cust = new Customer();
cust.entityID( “XYZ Inc” );
WriteResponse response = service.add( cust );
```

6. Logout to invalidate the current session.

```
service.logout();
```

Building an Application with Java using Apache Axis

This section provides details on how to use the Apache Axis framework (versions 1.3 and 1.4) to build a SuiteTalk application.

Note: The Java sample application available on the Developer portal provides an in-depth look at how to build an application using Apache Axis and NetSuite Web services. See Downloading Sample Applications for more information.

To use the Apache Axis framework with NetSuite Web services:

1. Install the Java 2 Platform.

Download and install the Java 2 Platform, Standard Edition, version 1.4 or higher from http://java.sun.com/j2se/1.4.2/download.html. Ensure that the executables are available through the system path.

2. Install Apache Axis.

   1. Download and install Apache Axis from http://ws.apache.org/axis/.

      Important: Only Axis versions 1.3 and 1.4 are supported in NetSuite Web services.

   2. Download and install the Apache Axis patch for cookie management from NetSuite: http://www.netsuite.com/portal/developers/resources/suitetalk-sample-applications.shtml

      The NetSuite Web service implementation requires the client application to support multiple cookies on one line, as is the standard for cookies. There is a bug in Apache Axis that puts each cookie on its own line in the HTTP Headers. The patch version of the axis.jar fixes this problem. Once downloaded, replace the existing axis.jar file in the lib directory of your Axis root directory with this version.

   3. Once installed, set an environment variable called AXIS_HOME to point to the Axis installation directory.

3. Install Apache Ant (Optional).

   Download and install Apache Ant, version 1.5 or higher, from http://ant.apache.org/. Apache Ant is a Java-based build tool that facilitates automation of the build process, including generating the proxy classes from the NetSuite WSDL.

   See the build.xml file in the Java sample application for a complete Ant build script.

4. Configure Java to generate unreferenced types.

   Set the **all** parameter in your axis-wsdl2java ant task to **true**. For example:

```
<axis-wsdl2java timeout="120000" output="${generated.src.dir}" verbose="true"
url="${wsdl-1.3.url}" all="true" wrapArrays="true">
```

5. Use Apache Ant to automatically generate the client proxy code.

   Using Apache Axis from the command line

   Use the WSDL2Java utility to generate the proxy classes as follows:

```
java -cp <classpath> org.apache.axis.wsdl.WSDL2Java <url>
```

Where the <classpath> points to the appropriate Apache Axis JAR files and <url> is the URL for the NetSuite WSDL.

For example, the following commands will set the class path and generate the proxy classes:

```
>set
CP=%AXIS_HOME%\lib\axis.jar;%AXIS_HOME%\lib\jaxrpc.jar;%
AXIS_HOME%\lib\commons-discovery.jar;%AXIS_HOME%\lib\wsdl4j.jar;
%AXIS_HOME%\lib\saaj.jar;>java -cp %CP% org.apache.axis.wsdl.WSDL2Java
https://webservices.netsuite.com/wsdl/v2012_1_0/netsuite.wsdl
```

Important: The URL for the WSDL may vary from the example shown above. See Understanding the Effects of Accounts' Host Data Center on URLs for an important note about these URLs.

Using the Apache Axis task in Apache Ant

Create a build target that uses the WSDL2Java Ant task as follows where the ${generated.src.dir} variable is the directory where the source code is generated and the ${wsdl.url}variable points to the NetSuite WSDL.

```
<target name="generate.interfaces" description="Generates
client interfaces using Axis">
<echo>Generating client interfaces using Apache Axis</echo>
<axis-wsdl2java output="${generated.src.dir}" verbose="true"
url="${wsdl.url}">
<mapping namespace="http://axis.apache.org/ns/interop"
package="interop"></mapping>
</axis-wsdl2java>
</target>
```

6. Ensure that the Ant executables are available on the system path. Run the Ant task as follows:

```
ant generate.interfaces
```

7. Implement your application by writing your business logic using the generated axis proxy classes.

    1. Locate the NetSuite service.

    ```
    NetSuiteServiceLocator service = new NetSuiteServiceLocator();
    ```

    2. Enable support for multiple cookie management.

    ```
    service.setMaintainSession( true );
    ```

    3. Get the NetSuite port.

    ```
    NetSuitePortType port = service.getNetSuitePort();
    ```

4. Create a valid session by populating the Passport object and then invoking the login operation.

```
passport.setEmail( "username@netsuite.com" );
passport.setPassword( "mypassword" );
role.setid( "3" );
passport.setRole( role );
passport.setAccount( "TSTDRV96" );
Status status = port.login( passport ).getStatus();
```

5. Implement your business logic. For example, create a new customer in NetSuite.

```
Customer cust = new Customer();
cust.setEntityID( “XYZ Inc” );
WriteResponse response = port.add( cust );
```

6. Logout to invalidate the current session.

```
port.logout();
```

Downloading Sample Applications

Web services developers can download sample applications from the NetSuite Developer Portal, at http://www.netsuite.com/portal/developers/resources/suitetalk-sample-applications.shtml. CRM and ERP sample applications are provided in both Java and .NET.

Capturing SOAP

When writing Web services you will want to capture and analyze SOAP request and response messages associated with an operation call. You can use such products as Actional Diagnostics or SOAPScope. You can also use the Web Services Usage Log within NetSuite. In most cases the usage logs will be sufficient for capturing and inspecting your SOAP. See Using the Web Services Usage Log for more details.

SuiteTalk Development Considerations

This section covers the following topics:

- Development Considerations Overview
- Understanding NetSuite Features in Web Services
- Understanding the Effects of Account Configuration in Web Services
- Understanding the Effects of Accounts' Host Data Center on URLs
- Working with Enumerations and Special Characters

Development Considerations Overview

Before you begin writing SuiteTalk applications, you should be aware that SuiteTalk adheres to the same role-based permission structure enforced in the NetSuite UI. Because a SuiteTalk application needs a pair of sign-in credentials to login, its permission to various operations and records is subjected to the role it uses, just like it would when it is used for a browser session. For example, a SuiteTalk application that logs in with the Accountant role will receive the same permissions as it will logging in to the browser interface using the same role.

Note: See Roles and Permissions in Web Services for additional details.

Another characteristic of SuiteTalk is that its behavior is very similar to that of the NetSuite UI. The workflow of a SuiteTalk application and its underlying SOAP exchange with NetSuite tightly mimics the browser interface. For example:

1. A successful login operation must be performed to obtain a session in order to perform any subsequent operations. (See the documentation on the login operation and on Web services Authentication for more details.)
2. Some add operations are done in a tandem, such as loading an existing sales order to gain the context and then adding a new item fulfillment record. The loading of the sales order (using the initialize API) is the first operation; adding the item fulfillment record is the second.
3. Restrictions and requirements on custom forms are honored in SuiteTalk. Therefore, a SuiteTalk application's attempt to set a field that is hidden on a form results in a permission error. Likewise, as in the UI, a SuiteTalk application must set fields that are required on the form it is using.

The combination of permission adherence and similar behavioral patterns between SuiteTalk and the NetSuite UI provides a consistent and predictable platform for developers.

Important: When you are unsure of how to achieve something with SuiteTalk, try observing how it is done in the UI, then replicate it programmatically.

Note: A SuiteTalk application should use the response object to handle any errors, if any, generated by a Web service operation.

Understanding NetSuite Features in Web Services

The NetSuite UI allows you to enable or disable certain features. When designing your Web services, it is important to know which features must be enabled in order for the service to execute properly. If the service calls for a function that is NOT available because the associated feature is disabled, a SOAP **InsufficientPermissionFault** occurs causing the entire service to fail. For example, in the NetSuite UI you can enable or disable the Opportunities feature for CRM. If disabled, a Web service call to add an OpportunityItem will fail.

Web services is itself a feature that **must** be enabled prior to submitting Web services requests. To enable the NetSuite Web Services feature, see Enabling the Web Services Feature.

Understanding the Effects of Account Configuration in Web Services

Form customization in an account or enabling/disabling features may result in required fields being added to various forms used on records. For each Web service request, fields available for the specified record are validated against the data being submitted, and errors are returned where field validation fails. However, for get, add, addList and search requests, it is possible that the field requirements change mid-request, resulting in errors for a **subset** of the request.

For example, suppose you want to submit 20 new customer records using the addList operation. Upon submission, field validation passes for the request. However, after the first 15 customers are added, the required fields may be changed within your NetSuite account, causing an error to be returned for the 5 remaining items.

Note: For more specific information on working with NetSuite records, fields, and forms in Web services, see Records, Fields, Forms, and Sublists in Web Services.

Understanding the Effects of Accounts' Host Data Center on URLs

Since September 2012, when NetSuite began hosting new customer accounts in multiple data centers, the URL to be used for Web services access depends upon the data center hosting each NetSuite account. For example, it could be either of the following:

- webservices.netsuite.com
- webservices.na1.netsuite.com

Your code needs to dynamically discover the correct URL for Web services access to a NetSuite account. There are two different methods you can use for this discovery.

- In the 2012.2 and later endpoints, you can use the getDataCenterUrls operation. This operation is not supported in earlier endpoints.
- You can use a REST service provided by NetSuite.
  - For general information about this service, see the help topic *Using the REST roles Service to Get User Accounts, Roles, and Domains*.
  - For example code that uses this REST service, see:
    - Java REST Example for Web Services Domain Discovery
    - .NET REST Example for Web Services Domain Discovery

  Note that NetSuite Developer Resources sample applications also incorporate RESTful code for dynamic discovery of Web services domain. These samples are available at: http://www.netsuite.com/portal/developers/resources/suitetalk-sample-applications.shtml.

Java REST Example for Web Services Domain Discovery

This example includes a sample call to the REST roles service to dynamically discover the correct Web services domain and a sample client.

Sample Java Call to the REST roles Service

```
package com.netsuite.prototypes;
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;
import org.json.simple.parser.JSONParser;
import org.json.simple.parser.ParseException;
import javax.net.ssl.HttpsURLConnection;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.URL;
public class GetRESTDomain
{
 static class URLTriplet
 {
  String restDomain;
  String systemDomain;
  String webservicesDomain;
 }
 private BufferedReader _br = null;
 public GetRESTDomain()
 {
  _br = new BufferedReader(new InputStreamReader(System.in));
 }
 /** * Retrieve domains of a NetSuite account through RESTful WS call.***/

 public URLTriplet getDataCenterUrls(String nsAccount, String nsEmail, String nsPassword) throw
s IOException, ParseException
 {
  String sysURL        = "https://rest.netsuite.com/rest/roles";
  URLTriplet urls      = new URLTriplet();
  // 'nlauth_account' should NOT be specified for this type of request otherwise an error is re
turned.//
  String nlAuth = "NLAuth nlauth_email=" + nsEmail + ", nlauth_signature=" + nsPassword;
  URL u = new URL(sysURL);
  HttpsURLConnection uc = (HttpsURLConnection) u.openConnection();
  uc.setAllowUserInteraction(Boolean.FALSE);
  uc.setInstanceFollowRedirects(Boolean.FALSE);
  uc.setRequestMethod("GET");
  uc.setRequestProperty("Authorization", nlAuth);
  BufferedReader in = new BufferedReader(new InputStreamReader(uc.getInputStream()));
  String inputLine = in.readLine();
  in.close();
  if (inputLine.length() > 0)
  {
   JSONArray jsonArr = (JSONArray) new JSONParser().parse(inputLine);

   for (int i=0 ; i<jsonArr.size() ; i++)
```

```
    {
     JSONObject json = (JSONObject) jsonArr.get(i);
     // Validate JSON object if (json.containsKey("account") && json.containsKey("dataCenterURLs
"))
     {
      JSONObject jsonAccount     = (JSONObject) json.get("account");
      JSONObject jsonDataCenters = (JSONObject) json.get("dataCenterURLs");
      // Validate account and retrieve domains if (jsonAccount.get("internalId").toString().equa
lsIgnoreCase(nsAccount))
      {
       urls.restDomain = jsonDataCenters.get("restDomain").toString();
       urls.webservicesDomain = jsonDataCenters.get("webservicesDomain").toString();
       urls.systemDomain = jsonDataCenters.get("systemDomain").toString();
       break;
      }
     }
    }
   }

  return urls;
 }

 public static void main(String[] args) throws IOException
{
  GetRESTDomain ns = new GetRESTDomain();
  while (true)
 {
   System.out.println("GET REST DOMAIN URL");
   System.out.print("Account      : ");
   String account = ns._br.readLine().trim();

   if (account.isEmpty())
   {
    break;
   }

   System.out.print("Email address: ");
   String email = ns._br.readLine().trim();
   System.out.print("Password     : ");
   String password = ns._br.readLine().trim();
   try
   {
    URLTriplet urls = ns.getDataCenterUrls(account, email, password);
    System.out.println("\nREST domain URL        : " + urls.restDomain + "\n");
    System.out.println("\nWeb Services domain URL: " + urls.webservicesDomain + "\n");
    System.out.println("\nSystem domain URL      : " + urls.systemDomain + "\n");
   }
catch (IOException e)
   {
    e.printStackTrace();
   }
catch (ParseException e)
   {
    e.printStackTrace();
   }
  }
  System.out.print("\nPress any key to exit...");
  ns._br.readLine().trim();
 }
}
```

Sample AXIS Client

```
package com.netsuite.prototypes;
import com.netsuite.devtools.lists.employees.Employee;
import com.netsuite.devtools.platform.NetSuitePortType;
import com.netsuite.devtools.platform.NetSuiteServiceLocator;
import com.netsuite.devtools.platform.core.RecordRef;
import com.netsuite.devtools.platform.core.types.RecordType;
/*** Example showing how to get Web Services domain pointing to the account's datacenter throug
h RESTful WS call for WS endpoints older than 12.2 (as the method getDataCenterUrls() has been
introduced in 12.2 WS endpoint). ***/
public class AXISClient
{
 private static final String ENDPOINT_VERSION = "NetSuitePort_2012_1";
 private static final String ACCOUNT = "123456";
 private static final String PASSWORD = "123456@netsuite.com";
 private static final String EMAIL = "passwd";
 NetSuitePortType _port;
 public AXISClient(String acct, String email, String passwd) throws Exception
 {
  NetSuiteServiceLocator service = new NetSuiteServiceLocator();
  GetRESTDomain restHelper = new GetRESTDomain();
  String wsDomain = restHelper.getDataCenterUrls(acct, email, passwd).webservicesDomain;
  service.setNetSuitePortEndpointAddress(wsDomain + "/services/" + ENDPOINT_VERSION);
  _port = service.getNetSuitePort();
  // initialize _port - authentication// ...
 }
 public static void main(String[] args) throws Exception
 {
  AXISClient client = new AXISClient(ACCOUNT, EMAIL, PASSWORD);
  // _port points to the correct datacenter. Perform operations as usual.// ...
  RecordRef rr = new RecordRef();
  rr.setInternalId("25");
  rr.setType(RecordType.employee);
  Employee e = (Employee)client._port.get(rr).getRecord();
  System.out.println(e.getFirstName() + " " + e.getLastName());
 }
}
```

.NET REST Example for Web Services Domain Discovery

This example includes a sample call to the REST roles service to dynamically discover the correct Web services domain and a sample client.

Sample .NET Call to the REST roles Service

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Collections.Generic;
using System.Xml.Linq;
```

```
using System.Linq;
using System.Runtime.Serialization.Json;
using System.Net;
using System.IO;
using System.Xml;

namespace NSClientCRM
{
    class GetRESTDomain
    {
        public URLTriplet getDataCenterUrls(String nsAccount, String nsEmail, String nsPassword
)
        {
            // Preset Values
            string sysURL = "https://rest.netsuite.com/rest/roles";
            URLTriplet urls = new URLTriplet();

            HttpWebRequest httpWebRequest = (HttpWebRequest)HttpWebRequest.Create(sysURL);

            // 'nlauth_account' should NOT be specified for this type of request otherwise an e
rror is returned.
            // Issue 238527
            httpWebRequest.Headers.Add("Authorization",
                                       "NLAuth nlauth_email=" + nsEmail + "," +
                                       "nlauth_signature=" + nsPassword);

            // set content type
            httpWebRequest.ContentType = "application/json";

            // execute http GET
            WebResponse webResponse = httpWebRequest.GetResponse();
            Stream httpBody = webResponse.GetResponseStream();

            string jsonContent = "";
            int i = 0;

            StreamReader streamReader = new StreamReader(httpBody);

            while (jsonContent != null)
            {
                i++;
                jsonContent = streamReader.ReadLine();
                Stream mainContent = streamReader.BaseStream;

                if (jsonContent != null)
                {

                    // convert json content to bytes
                    byte[] bytesJsonContent = Encoding.Unicode.GetBytes(jsonContent);

                    // convert the json to an xml format so that xpath can be used to locate sp
ecific
                    // nodes & attributes
                    XmlDictionaryReader xmlReader =
                        JsonReaderWriterFactory.CreateJsonReader
                        (bytesJsonContent, new XmlDictionaryReaderQuotas());

                    // this returns the top-level element of the xml document;
                    // all of the following xpath statements are executed off of this element
                    XElement root = XElement.Load(xmlReader);
```

```
                    //Linq Query
                    IEnumerable<XElement> restDomains =
                    from c in root.Elements("item")
                    where (string)c.Element("account").Element("internalId").Value == nsAccount

                    select c;


                    // Outputs Data Center URLs for the specified account
                    if (restDomains != null)
                    {
                        foreach (XElement elementFromLinq in restDomains)
                        {
                            string accountIntId = elementFromLinq.Element("account").Element("i
nternalId").Value;
                            string accountName = elementFromLinq.Element("account").Element("na
me").Value;
                            Console.WriteLine("Data Center URLs for " + accountIntId + " " + ac
countName);
                            Console.WriteLine("-----------------------");
                            Console.WriteLine("Role: " + elementFromLinq.Element("role").Elemen
t("name").Value);
                            Console.WriteLine("Rest Domain: " + elementFromLinq.Element("dataCe
nterURLs").Element("restDomain").Value);
                            Console.WriteLine("System Domain: " + elementFromLinq.Element("data
CenterURLs").Element("systemDomain").Value);
                            Console.WriteLine("Web Services Domain: " + elementFromLinq.Element
("dataCenterURLs").Element("webservicesDomain").Value);
                            Console.WriteLine(" ");
                            urls.webservicesDomain = elementFromLinq.Element("dataCenterURLs").
Element("webservicesDomain").Value;
                            urls.systemDomain = elementFromLinq.Element("dataCenterURLs").Eleme
nt("systemDomain").Value;
                            urls.restDomain = elementFromLinq.Element("dataCenterURLs").Element
("restDomain").Value;
                            break;
                        }
                    }

                    Console.Write("Number of Roles for this Account: " + restDomains.Count());
                    if (restDomains != null)
                    {
                        Console.WriteLine("Roles");
                        Console.WriteLine("-----------------------");
                        foreach (XElement elementFromLinq in restDomains)
                        {
                            Console.WriteLine(elementFromLinq.Element("role").Element("name").V
alue);
                        }

                    }
                }
            }
            return urls;
        }
    }
}
```

Sample .NET Client

```
//...
NSClientCRM ns = new NSClientCRM();

GetRESTDomain restHelper = new GetRESTDomain();
URLTriplet urls = restHelper.getDataCenterUrls("123456", "123456@netsuite.com", "passwd");

ns._service.Url = urls.webservicesDomain + "/services/NetSuitePort_2012_1";
//...
```

Working with Enumerations and Special Characters

When enumerations contain either special characters ("(") or reserved keywords ("private" or "public" for example), both .NET and Axis may generate less usable code on the client side. To alleviate this problem, all enumerated values in NetSuite Web services are prefixed with an underscore "_", except for enumerated values from the platformCore, platformCoreTyp, platformFaults, platformFaultsTyp, and platformMsgs XSDs. For example, without the "_", .NET prepends an "@" symbol to the variable, as in "@private".

Working with Images in Web Services

To reference an image, the image must first be uploaded to the NetSuite file cabinet, and then referenced using the image name specified in the file cabinet. For more information on working with images (and other file types) in the file cabinet, see the help topic *File*.

Chapter 3 Setting Web Services Preferences

NetSuite preferences control how certain services are executed. Therefore, before using Web services with your NetSuite account, ensure that the appropriate preferences are enabled, as defined in the following sections:

- Setting Company-wide Preferences
- Resetting Default Behavior
- Setting Request Level Preferences
- Setting Search Preferences
- Setting the Internal ID Preference

Setting Company-wide Preferences

You can set the following company-wide preferences at Setup > Integration > Web Services Preferences. Changes to these preferences are propagated to **every** user within the company.

- **Search Page Size** — Determines the number of records returned for a given search. The default page size is 1,000. This value must be greater than 5 and less than or equal to 1,000. The page size entered here can be overridden at the individual request level. However, the following exception can be thrown:

  ExceededMaxRecordsFault — the page size is set to greater than the record size limit and the actual request includes more than the maximum allowed records. The request fails.

- **Use Conditional Defaults on Add** — See Resetting Default Behavior for more information.

Note: This preference is only available to endpoints **2009.2** and lower.

- **Treat Warnings as Errors** — If enabled, warning messages generated by Netsuite are treated as errors causing an exception to be thrown that results in rejection of the request. For more information on the difference between errors and warnings, refer to Understanding Web Services Warnings, Errors, and Faults.
- **Use Conditional Defaults on Update —** See Resetting Default Behavior for more information.

Note: This preference is only available to endpoints **2009.2** and lower.

- **Disable Mandatory Custom Field Validation** — If enabled, when data for a custom field is required for NetSuite UI submissions, it is NOT required when submitting a Web services request. If not enabled, an error is thrown whenever the data for a required custom field is not provided.

It is recommended that you enable this setting for cases where values for a required custom field may not be provided, such as when integrating with an application that does not include equivalent fields. If this setting is not enabled, and a request does not include data for a mandatory custom field, a **CSTM_FIELD_VALUE_REQD** error is returned. The error does not provide details on the data required.

- **disableSystemNotesForCustomFields** — When importing data from Custom Fields, you have the option to disable the creation of system notes during the import for those fields. Depending on the size of your import, this may significantly increase performance. This preference is available at the request level only. To enable the preference, submit the following in your SOAP Header (see Setting Request Level Preferences for more details on setting request level preferences):

```
<platformMsgs:disableSystemNotesForCustomFields>true
</platformMsgs:disableSystemNotesForCustomFields>
```

Important: System generated notes are used in NetSuite to track changes to a record including what action was taken, when the record was modified and the user that was responsible for the change. This is important for maintaining a complete audit trail. If you turn off system generated notes for custom fields, specific changes related to custom fields within the imported record are NOT recorded in NetSuite. All changes for standard fields are logged as usual. Therefore, if a custom field contains sensitive information that is critical for audit purposes, you should NOT disable system generated notes.

- **Disable Client SuiteScript** — Note that you must select at least one of the following preferences: Disable Client SuiteScript or Disable Server SuiteScript and Workflow Triggers. That is, either client or server SuiteScript must be disabled. You can also disable both (which means you can select both options). But at least one of these options must be selected.

Important: If you have enabled this preference and are experiencing unexpected errors during a Web services operation on a form that has Client SuiteScript associated with it, disable Client SuiteScript and then run the operation again to verify if Client SuiteScript is the cause of the problem.

- **Disable Server SuiteScript and Workflow Triggers** — Note that you must select at least one of the following preferences: Disable Client SuiteScript or Disable Server SuiteScript and Workflow Triggers. That is, either client or server SuiteScript must be disabled. You can also disable both (which means you can select both options). But at least one of these options must be selected.

  If you are doing a historical import, it is recommended that you disable Server SuiteScript. If you are syncing live data or running a partner application (for example, Outlook Sync) it is recommended that you enable Server SuiteScript to ensure your business logic is run for your integrated application. Note that running Server SuiteScript will have a negative performance impact.

Be aware that if this option is enabled, workflows do not run when Web services calls create or update records. Do not enable this option if you want Web services calls to trigger workflows based on record creation or update.

Important: To ensure that certain business logic is always executed for your integrated processes, use Server SuiteScript instead of Client SuiteScript for a more robust implementation.

Resetting Default Behavior

Within the NetSuite UI, there are three types of default behaviors that may be associated with any given record.

1. Record fields can be automatically populated with default values.
2. Records can have related fields that are automatically populated with default values when an initial value is entered. These fields are populated depending on the *condition* of the initial field.
3. Records can be populated with a *calculated* value depending on the values set in a particular field.

When using Web services, you may want to change the default behavior assigned to records since there is no visual confirmation of the default values being submitted. To specify your Web services behavior, select one of the following options.

Important: The following preferences are only available to endpoints **2009.2** and lower.

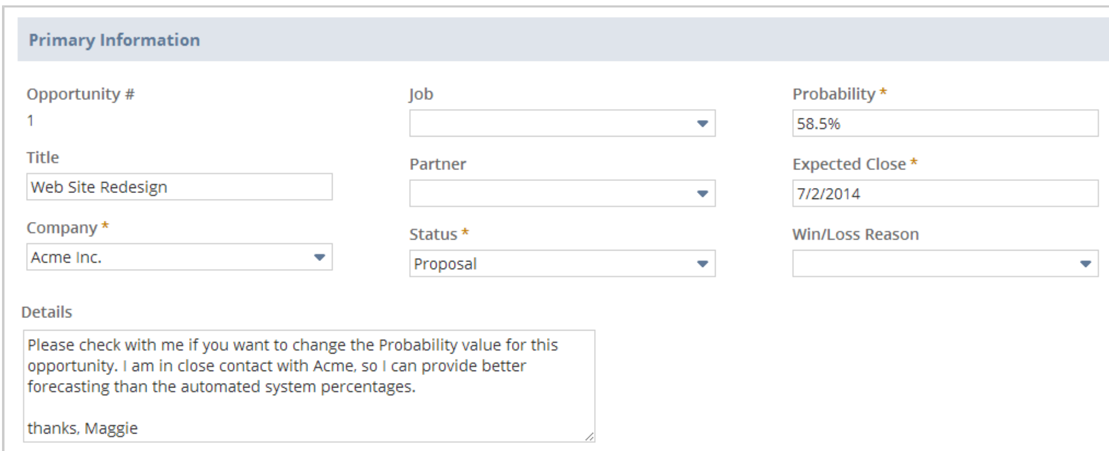
- **Use Conditional Defaults on Add** – Similar to the UI, if enabled, related fields are automatically populated with default values when a related value is entered in another field while *creating* a new record. If not enabled, no default values for conditional fields are submitted.
- **Use Conditional Defaults on Update** – If enabled, related fields can be automatically populated with default values when a value is entered while *updating* an existing record. If not enabled, no default values for conditional fields are submitted. This prevents overriding existing values that the user may not want to change.

Be aware that you **cannot** change the *default* behavior of calculated fields. Calculated fields are always reset when related fields are changed. However, you can override the value of the calculated field by submitting a value for that field in the request. Also, some fields within NetSuite are set as having slaving performed regardless of any default settings — the slaving values are mandatory. For these fields, unless a value is explicitly set, the field value is set as defined in the slaving definition regardless of default settings.

Example

When updating an Opportunity transaction, a change to the Status field causes the Probability field to automatically default to a new *conditional default* value. However, the Probability field can also be overridden by the user.

For example, when you set the status to Proposal, the Probability value automatically changes to 50.0%. However, a user could manually change that to some other number, as shown in the following screenshot.



Therefore, it may be undesirable to have the field automatically populated with a conditional default value as part of a Web services update operation. In that case, you would choose a value of false for the Use Conditional Defaults on Update preference.

Note: Endpoints prior to version 2.0.0 also include a Use Defaults preference. If enabled, this preference allowed the use of default values for records. If not enabled, default values were not set, forcing the Web services application to provide all required values. When upgrading to version 2.0.0 or later, you must adjust your code accordingly, because this preference will not be available.

Setting Request Level Preferences

In addition to setting preferences to be used for all requests submitted to NetSuite, you can also set preferences at the request level.

To set preferences at the request level, your code must do the following:

- Clear the headers (Java only)
- Create a new SOAPHeaderElement: the same command is used for all preference elements. You must use camelCase style capitalization for the preference type name. Although no errors are thrown if this is not used correctly, your settings will be ignored.

- Create the Preference object. This object contains the elements you are allowed to set. These elements include:
  - warningAsError
  - useConditionalDefaultsOnAdd
  - useConditionalDefaultsOnUpdate
  - disableMandatoryCustomFieldValidation
  - disableSystemNotesForCustomFields
  - ignoreReadOnlyFields (see Setting the ignoreReadOnlyFields Preference for details)
- Set the header (Java Only)
- If executing a search, set your search preferences. To set search preferences, create a SearchPreferences object and set available search preference elements. See Setting Search Preferences.

Sample Code

SOAP Request

```
<soap:Header>
<platformMsgs:preferences>
 <platformMsgs:warningAsError>true</platformMsgs:warningAsError>
 <platformMsgs:useConditionalDefaultsOnAdd>true</platformMsgs:useConditionalDefaultsOnAdd>
 <platformMsgs:useConditionalDefaultsOnUpdate>true
 </platformMsgs:useConditionalDefaultsOnUpdate>
 <platformMsgs:disableMandatoryCustomFieldValidation>true
 </platformMsgs:disableMandatoryCustomFieldValidation>
 <platformMsgs:disableSystemNotesForCustomFields>true
 </platformMsgs:disableSystemNotesForCustomFields>
</platformMsgs:preferences>
</soap:Header>
```

C#

In this example, a SearchPreference object is set as well as a warningAsError object. See Setting Search Preferences for more information on the SearchPreference object.

```
// Set up request level preferences as a SOAP header
Preferences prefs = new Preferences();
_service.preferences = prefs;
// Preference to ask NS to treat all warnings as errors
prefs.warningAsErrorSpecified = true;
prefs.warningAsError = false;
// Invoke search() web services operation
_service.searchPreferences.pageSize = 20;
_service.searchPreferences.pageSizeSpecified = true;
SearchResult response = _service.search( custSearch );
```

Java

Note that the following code snippet assumes that the 2013.1 endpoint is being used.

```
private void setPreferences() throws SOAPException
{
   stub = (NetSuiteBindingStub) _port;
   stub.clearHeaders();
   SOAPHeaderElement prefHeader =
  new SOAPHeaderElement("urn:messages_2013_1.platform.webservices.netsuite.com", "preferences")
;
  Preferences prefs = new Preferences();
   prefs.setWarningAsError(new Boolean(false));
   prefs.setUseConditionalDefaultsOnAdd(new Boolean(false));
   prefs.setUseConditionalDefaultsOnUpdate(new Boolean(false));
   prefs.setDisableMandatoryCustomFieldValidation(new Boolean(true));
   prefs.setDisableSystemNotesForCustomFields(new Boolean(true));
   prefs.setIgnoreReadOnlyFields(new Boolean(true));
   prefHeader.setObjectValue(prefs);

      stub.setHeader(prefHeader);
}
```

Setting the ignoreReadOnlyFields Preference

After getting or initializing a record, it is recommended that you set the ignoreReadOnlyFields preference to *true* when submitting the record using write operations such as add/addList or update/updateList. Setting this preference to *true* reduces the possibility of receiving an INSUFFICIENT_PERMISSION error because a read-only field was mistakenly set and then submitted.

It is also recommended that you set this preference to *true* when using the initialize/ initializeList operations. To submit an initialized record without having to remove read-only fields populated during the initialization, set the ignoreReadOnlyFields preference header preference to *true*. When this preference is set to *true*, read-only fields are simply ignored during the Web services request.

Note: For details on initialize operations, see initialize / initializeList.

Setting Search Preferences

The SearchPreferences type is used to set preferences for the search. This must be set in the SOAP message header. The SearchPreferences type contains the following fields.

Element Name	XSD Type	Notes
bodyFieldsOnly	boolean	Defaults to TRUE and indicates that the information in the body fields of the record are returned — significantly improving performance. Any fields in associated lists or sublists are not returned. If the bodyFieldsOnly field is set to FALSE, all fields

Element Name	XSD Type	Notes
		associated with the record are returned. See Returning Body Fields Only below.
pageSize	int	See Pagination below.
returnSearchColumns	boolean	Defaults to TRUE, meaning that only search columns will be returned in your search. See Returning Search Columns.
passport	platformCore: Passport	References the Passport object. This is an optional argument depending on whether you are authenticating through the *login* operation or you authenticating through request-level credentials.

Note: For information and code samples regarding setting additional request-level preferences not related to search, see Setting Request Level Preferences.

Returning Body Fields Only

When performing searches on records, by default only the body fields are returned only. To return field (line item) values for all sublists on a record, you must set the bodyFieldsOnly element to FALSE.

It is recommended that the default settings are used wherever sublist values are not necessary since this significantly improves performance.

Pagination

Pagination breaks up a large number of records that are part of a search result set into smaller pages. You then have the ability to retrieve these pages of records one at a time. Pagination is used mainly to make processing large result sets earier and more manageable.

There are two different ways for a page size to be determined:

- **System-defined maximum page size**: This is set to 1000 records for synchronous Web service requests by default
- **User-defined page size parameter**: The pageSize element in the searchPreference type is used to specify a value for the page size. The value must be greater than 10 and less than the system-defined maximum of 1000.

If the number of records for the search results exceeds the page size, the remaining results must be retrieved in a subsequent operation using the searchMore operation.

Returning Search Columns

When executing an advanced search, you can set the SearchPreferences. *returnSearchColumns* preference to TRUE to ensure that only search return columns are returned in a search. An error is thrown if *returnSearchColumns* is set to TRUE and you have not specified search return columns in your request. Note that in an advanced search, the *bodyFieldsOnly* preference is ignored.

The default value for *returnSearchColumns* is TRUE.

For information on advanced search in Web services, see Advanced Searches in Web Services.

Sample Code

Java

In this example, a SearchPreference object is instantiated and search preferences are set.

```
NetSuiteBindingStub stub = (NetSuiteBindingStub)aPort;
stub.clearHeaders();
SOAPHeaderElement searchPrefHeader = new
SOAPHeaderElement("urn:messages_2_5.platform.webservices.netsuite.com",
"searchPreferences");
SearchPreferences searchPrefs = new SearchPreferences();
searchPrefs.setPageSize(new Integer(nPageSize));
searchPrefs.setBodyFieldsOnly(isBodyFieldsOnly);
searchPrefHeader.setObjectValue(searchPrefs);
stub.setHeader(searchPrefHeader);
```

Setting the Internal ID Preference

You can configure NetSuite to display internal ID values on forms in the UI. This is useful during development as a quick reference to verify that the internal ID values submitted on Requests match the records expected as shown in the UI.

To display internal ID values on forms, go to Home > Set Preferences. Click the General tab, and in the Defaults section, click **Show internal IDs**. When this preference is enabled, a given list displays the IDs as one of the columns. For example, List > Relationships > Customers displays the internal ID as the second column.

Note: Changes to these preferences affect the current user only. Also, another way to quickly determine the internal ID value for an item is by hovering over the item in the UI and noting the id value in the URL. For example, when hovering over a specific customer at Lists > Relationships > Customers, you may see something like https://webservices.netsuite.com/app/common/entity/custjob.nl?id=272 (where 272 is the internal ID).

Chapter 4 Roles and Permissions in Web Services

NetSuite provides many standard roles with predefined permissions. A role is a set of permissions that allows customers, vendors, partners and employees access to specific aspects of your data. Each role grants access at a certain level for each permission.

When logging in using Web services you may provide a **role id** along with your **credentials**. The role id that you provide must have Web services permissions, otherwise an **INSUFFICIENT_PERMISSION** error is returned. If no role id is provided, then the user's default role is used. If the default role does NOT have Web services permissions, then a **ROLE_REQUIRED** fault is returned.

The following topics are provided in this section. They do not need to be read in order. However if you are new to NetSuite Web services, it is recommended that you read each topic to understand how NetSuite roles and permissions apply in a Web services context.

- Role and Permission Considerations When Developing in SuiteTalk
- Assigning the Web Services Permission to a Role
- Setting a Default Role for a Web Services User
- Setting a Web Services Only Role for a User
- Customer Center, Vendor Center, and Partner Center Roles
- Internal IDs Associated with Roles

Role and Permission Considerations When Developing in SuiteTalk

Due to SuiteTalk's reliance on NetSuite's role-based permissions, it is important for SuiteTalk developers to put that into considerations during the design phase to ensure smooth deployments.

It is common for developers to use the administrator role during development time because it gives them full permissions and access to all the records and operations. However, the target end users are likely to have less powerful roles, which may not have access to the data the SuiteTalk application requires.

Another role-related consideration is the preferred custom forms of some roles may not have access to certain fields or sublists that a SuiteTalk application requires. Hence the application's attempts to set those fields will result in permission errors.

The solution to these problems is to define a custom role and custom forms for the SuiteTalk application. The custom role should have the correct access permissions and operations permissions that the SuiteTalk application needs. The custom forms should give access to fields and sublists that are relevant to the SuiteTalk application. All SuiteTalk supported records have a `customForm` field for the application to reference specific custom forms.

Important: When testing SuiteTalk applications, you should do so using the role(s) of your intended users(s), in addition to the administrator role, to catch permission-related defects.

Assigning the Web Services Permission to a Role

All standard NetSuite roles have Web services permissions by default. For security reasons, it is recommended that you restrict permissions levels and access allowing only the most restricted permissions necessary to perform a given set of operations.

For non-standard or custom roles, use these steps to assign the Web services permission to the role.

To assign the Web Services permission to a role:

1. Go to Setup > Users/Roles > Manage Roles.
2. Locate the role you want to modify. Click the corresponding **Edit** or **Customize** link.
3. Click the **Setup** subtab.
4. In the **Permissions** dropdown list, select **Web Services**.
5. In the Level dropdown list, select **Full**.

   Note: Users with a Web Services permission level other than Full (View, Create, Edit) cannot log in to Web services. The Full level is required. Also note that the Web Services permission does not provide access to the Web Services Usage Log; only administrators can access this log. For details on the Web Services Usage Log, see Using the Web Services Usage Log.

6. Click **Add**.
7. Click **Save**.

Note also:

- If you are building an integrated application, it is best to create a new role or customize an existing role and grant the minimum set of permissions that are necessary for the client to carry out its functions. It is not recommended that users are granted the Full Access role or that a user should be assigned administrator privileges in your Web services.

- If your role has permission to view credit card data on the user interface, you can also retrieve this information through Web services calls. This is beneficial to integrated applications that use an external credit card processor. Based on your role, you may be able to retrieve the credit card on file for your customers.
- In order for a role to use the getDeleted operation, the role has to have both the Deleted Records and the Web Services permissions. Note that users who have the Deleted Records permission can access results about any type of record that was deleted, even if they do not have permission to create or modify that record type. They also have permission to use the Deleted Record search type, unless the Web Services Only option has been selected for the role.

Setting a Default Role for a Web Services User

You can specify a default role for any user making Web services requests. The permissions for the default role are determined as follows:

- First, any role specified in the Passport object of the request is used. The role defined here must be a valid role contained in the Employee record of the given user. (For information on the Passport object, see login. The Passport object is defined in the platformCore XSD.)
- If a role preference is not set at the request level, then any default Web services role as defined in the Web Services Preference page for the given user is used.

  Only one default Web services role can be assigned per user and only roles that contain the Web services permission can be specified as a default Web services role. Note that the user may be assigned a different role than those specified in their Employee record. In other words, a user may have greater or lesser permissions using Web services as compared to the UI.
- If neither the request nor the Web services default role is set, then the user's default UI role is used, provided it has the Web services permission.

Note: All standard roles have the Web services permission by default when the Web services feature is enabled. Custom roles, however, must be explicitly set to have Web services permissions.

To set a specific default role for a Web services user:

1. Click Setup > Integration > Web Services Preferences.
2. Select the desired user from the **Name** dropdown list.
3. Select the default role to use for Web services requests for this user.

   The internal ID for the selected role automatically populates the **ID** field.
4. Click **Done**.

5. Click **Save**.

Setting a Web Services Only Role for a User

In NetSuite you can designate a user's role as **Web Services Only**. When a user logs in with a role that has been designated as Web Services Only, validation is performed to ensure that the user is logging in through Web services and not through the UI.

Note: Your account must have the Web services feature enabled for the Web Services Only check box to appear. See Enabling the Web Services Feature for steps on enabling this feature.

The Web Services Only role increases the security of an integrated application by prohibiting a UI user from accessing the system with permissions and privileges that are specifically created for a Web services applications. For example, you may have a Web services application that requires certain employees to have write access to several records. However, you want to prohibit the employees from being able to edit these records directly from within the NetSuite UI. If you assign the Web Services Only role to specified employees, the employees can log in to NetSuite and access the application through Web services, however, the employees cannot switch to their other roles within the system and write/edit/delete these any data-sensitive records.

Important: The Web Services Only role does not appear in the Change Role drop-down list. Therefore, users cannot change their roles from their original UI login role (A/P clerk, for example) to their Web Services Only role from within the UI.

To designate a role as Web Services Only:

1. Click Setup > Users/Roles > Manage Roles.
2. On the Manage Roles list page, locate the role you want to set as Web Services Only.
3. Click the corresponding **Edit** or **Customize** link.
4. Check the **Web Services Only Role** box.
5. Click **Save**.

When to Set the Web Services Only Role

A role should not be designated as **Web Services Only** until the developers building and testing the integrated application have completed the application. Waiting to designate a role as **Web Services Only** allows developers to go back and forth during design and development time to test the permissions for the role that is designed specifically for an integrated application. Once the development and testing is complete, the developer can set the **Web Services Only** role to TRUE for a specified role to prevent users with this role access to the UI with this set of permissions and privileges.

Note: External roles such as Customer Center, Partner Center, Advanced Partner Center, Vendor Center, and Employee Center should not be customized to have Web Services Only permissions.

Customer Center, Vendor Center, and Partner Center Roles

The Customer Center, Vendor Center, and Partner Center roles have implicit Web services permissions. This allows integration with an externally hosted Web site where a client can execute any task available under the center-specific role through Web services. For example, the client could login and submit an order on behalf of the customer.

Note: It is not recommended that you customize Customer Center, Partner Center, or Partner Center Roles to have only Web Services Only permissions. For information on the Web Services Only role, see Setting a Web Services Only Role for a User.

If you choose you can remove Web services permissions from the Customer Center, Vendor Center, and Partner Center roles.

To remove Web services permissions:

1. Go to Setup > Users/Roles > Manage Roles.
2. Locate the role you want to modify.
3. Click the corresponding **Edit** or **Customize** link.
4. Click the **Setup** subtab.
5. In the **Permissions** dropdown list, click the **Web Services** line to make it editable. The page updates to include several buttons below the line.
6. Click the **Remove** button.
7. Click **Save**.

Internal IDs Associated with Roles

The table in this section lists the standard NetSuite roles and the associated internal ID values. You can use these internal ID values in the Passport object, which is used for the login operation. For information on the Passport object and the login operation, see login.

If you have the Show Internal IDs preference on, you can look up the internal ID of a role by going to Setup > Users/Roles > Manage Roles. The role ID appears in the Internal ID column.

For instructions on setting the Show Internal IDs preference, see Setting the Internal ID Preference.

Note: In addition to the NetSuite standard roles, there may exist custom roles that have been created by your organization. Custom roles are assigned internal IDs sequentially (starting with 1001).

ID	Is Settable	Role
1	Y	Accountant
2	Y	Accountant (Reviewer)
3	Y	Administrator
4	Y	A/P Clerk
5	Y	A/R Clerk
6	Y	Bookkeeper
7	Y	CEO (Hands Off)
8	Y	CEO
9	Y	Sales Manager
10	Y	Sales Person
11	Y	Store Manager
12	Y	Support Manager
13	Y	Support Person
14	Y	Customer Center
15	Y	Employee Center
16	Y	Vendor Center
17	**N**	Shopper
18	Y	Full Access
19	Y	Warehouse Manager
20	Y	Payroll Manager
21	**N**	Partner Center
22	Y	Intranet Manager
23	Y	Marketing Manager
24	Y	Marketing Assistant
25	Y	System Administrator
26	Y	Sales Administrator
27	Y	Support Administrator
28	Y	Marketing Administrator
29	**N**	Advanced Partner Center
30	Y	NetSuite Support Center

ID	Is Settable	Role
31	**N**	Online Form User

Chapter 5 Records, Fields, Forms, and Sublists in Web Services

The following topics are covered in this section. These topics do not need to be read in order, however, if you are new to SuiteTalk development, you will need to review each section to fully understand the SuiteTalk API.

- Working with Records in Web Services
- Working with Fields in Web Services
- Working with Forms in Web Services
- Working with Sublists in Web Services

Working with Records in Web Services

Most standard NetSuite records are supported by SuiteTalk. The list of supported records spans all areas of the NetSuite application from ERP to CRM to customization. For a list of records that are supported in SuiteTalk, see the help topic *Web Services Supported Records*.

In the SuiteTalk API, the Record class is the abstract super-class of all supported records. A supported record is always a concrete sub-class of Record. Due to the neutrality required to be language agnostic, the SuiteTalk classes inheritance chain remains simplistic and does not implement language-specific object-oriented concepts such as multiple inheritance and interfaces.

NetSuite records are divided into the following broad categories:

- Business Records
- Search Records
- Subrecords

A record's standard body fields are its attributes, for example: `Customer.email`, `SalesOrder.salesRep`. Composite attributes such as line items or sublists are structured as complex objects that contain arrays, for example: `SalesOrder.itemList`, `CalendarEvent.attendeeList`. Custom fields within a record (if available) are also structured as composite attributes, for example: `Contact.customFieldList`.

Note: A record element that ends with **List** is generally a sublist. (The exception is the customFieldList element, which represents custom fields on the record.) For information on working with sublists in Web services, see Working with Sublists in Web Services. For more general information on sublists in NetSuite, see the help topic *What is a Sublist?*

In addition to standard records, SuiteTalk also supports custom objects and their metadata (see the help topic *Customization* for more details). Using the getCustomizationId SuiteTalk API, an application external to NetSuite can query a NetSuite account to obtain metadata about the custom objects that have been implemented in the account. This allows you to build and ship off-the-shelf, generic applications that will work with any account. For example, a SuiteTalk point-of-sale application can be designed to determine (during runtime) all the custom fields applied to a NetSuite CashSale record, so that it can then import CashSale records with the necessary custom fields set.

Business Records

A NetSuite *business* record is a top-level record used in all operations other than the login and logout operations. These records represent specific business functions such as creating a customer or updating an opportunity. For a list of all supported business records, see the help topic *Web Services Supported Records*.

Business record fields are populated and sent through SOAP during a Web service request or response. Business records are also returned in get and search operations.

Search Records

Search record types encapsulate the available search criteria for a given NetSuite business record type. A NetSuite *search* record is defined as a top-level record that is used in the request portion of a search operation. Any field defined within a search record must be of one of the following logical types.

Type	Description
String	Corresponds to the SearchStringField type
Int	Corresponds to the SearchTextNumberField type
Double	Corresponds to the SearchDoubleField type
Boolean	Corresponds to the SearchBooleanField type
Datetime	Corresponds to the SearchDateField type
MultiSelectRef	Corresponds to the SearchMultiSelectField type
MultiSelectEnum	Corresponds to the SearchEnumMultiSelectField type

For complete details on using the search operation, see search. Also see the following topics, which describe all search types that can be used when constructing Web services searches.

- Which SuiteTalk objects are used in a basic search?
- Which SuiteTalk objects are used in a joined search?
- Which SuiteTalk objects are used in advanced search?

Subrecords

As of the 2011.2 and later endpoints, a new category of records, called subrecords, is supported. Note that in the SuiteTalk schema, a subrecord is defined simply as a record, and is used in the same manner as a business record. However, a subrecord is different from other SuiteTalk records, because a subrecord does not have a record type, and a subrecord is always contained within a parent record.This subrecord is always contained within a parent transaction or parent item record.

Any add, get, update, delete, or search operation on a subrecord must be performed within the context of an operation on its parent record. For example, if you want to update inventory detail data for a purchase order transaction, you must update the purchase order record itself; you cannot do an independent update of the inventory detail object. And this same limitation applies to all operations on a subrecord.

The following preferences apply to subrecords:

- For get operations, a SOAP level preference called returnSubRecords will be available, to indicate whether subrecord data should be returned. By default, this preference is set to true.
- The returnSubRecords preference is respected by search operations, but if the bodyFieldsOnly preference is enabled for search, subrecord data is not returned.
- For an add operation, you can put the subrecord in the nullFieldList to null out the subrecord.
- If a subrecord is not in the nullFieldList, its replaceAll attribute determines whether the subrecord is added or updated during an add operation on its parent record. If this attribute is set to true, the subrecord is deleted and a new one is added. If this preference is set to false, an update of the subrecord is attempted.

Using Internal IDs, External IDs, and References

Each record in NetSuite can be uniquely identified by its record type in combination with either an external ID or a system-generated NetSuite internal ID.

Internal and external IDs are NOT unique across all record types. Therefore, to perform an operation on an existing record, you need two pieces of data: the record type and either the internal ID or the external ID.

References are implemented through the RecordRef type, which is defined in the core XSD.

The RecordRef type is described by three attributes — the internal ID, external ID, and the type:

```
<complexType name="RecordRef">
<complexContent>
 <extension base="platformCore:BaseRef">
  <attribute name="internalId" type="xsd:string"/>
  <attribute name="externalId" type="xsd:string"/>
  <attribute name="type" type="platformCoreTyp:RecordType"/>
 </extension>
</complexType>
```

Important: When referencing records in any given update operation, you must provide the internalId or the externalId, but not both. If both are provided, the internalId is used to locate the record and the externalId is ignored.

Uniqueness of Internal and External IDs

Internal and external IDs are unique only within certain record groups. For details, see Shared Internal and External IDs.

Further, internal IDs are not reused in the system. Once an internal ID has been assigned, if the associated record is subsequently deleted, the ID is not reused.

Note also that an external ID exists only if you provided it, either at the time of record creation or during an update. For more on external IDs, see Understanding External IDs.

Internal IDs Do Not Indicate When Record Was Created

NetSuite's assignment of internal IDs to new records is sequential, but not based on chronology or when the record was added. Therefore, users should not conclude that records with lower internal IDs were added first, while those with higher internal IDs were added last.

For custom applications that rely on the chronological aspect of when data was entered, NetSuite suggests using the Date Modified or Date Created fields, as they are guaranteed to be chronological. Additionally, for transactions with auto-generated numbers, the sequence number (for example, “tranid”) is guaranteed to be sequential based on the date transactions were entered. The internal ID of a record should never be used to determine when data was entered.

Internal IDs for NetSuite Record Types

Like individual records, each record *type* also has an internal ID. In certain situations, you must use this ID to specify which type of record you are referencing. By contrast, in some cases, you can use a string to reference the record type (for example, when using RecordRef). But in other cases you must use the internal ID, or typeId (for example, when using ListOrRecordRef or CustomRecordRef). The table below lists the NetSuite record types and their associated typeIds.

Note that custom fields of type MultiSelectCustomFieldRef on custom records also reference these values, because they contain an array of ListOrRecordRef types. For details on working with custom records, see the help topic *Custom Record*.

Record Type	typeId
Account	-112
Accounting Period	-105
Bin	-242
Call	-22
Campaign	-24
Case	-23
Class	-101
Competitor	-108
Contact	-6
Customer	-2
Customer Category	-109
Department	-102
Email Template	-120
Employee	-4
Employee Type	-111
Entity Status	-104
Event	-20
Issue	-26
Item	-10
Item Type	-106
Job (Project)	-7
Location	-103
Module	-116
Opportunity	-31
Partner	-5
Product	-115
Product Build	-114
Product Version	-113
Project (Job)	-7
Role	-118
Saved Search	-119
Subsidiary	-117
Task	-21

Record Type	typeId
Transaction	-30
Transaction Type	-100
Vendor	-3
Vendor Category	-110

Understanding External IDs

The externalId attribute of a RecordRef provides a means to reference an object by its foreign key in an external database.

The externalId attribute can be set during an add or update operation. External IDs are useful in the following situations:

- **Maintaining client ID relationships**

  In cases where a client application already maintains references between records, set the externalId attribute for each record during imports. In subsequent API calls, you can then reference associated records by the known external ID.

- **Establishing relationships during a single import operation**

  For example, suppose you want to import customer records with references to sales reps into NetSuite. If no external ID is used, you would need to import the customer records, determine the IDs of the related sales reps' employee records, and then re-import the customer records with the sales reps ID references. By providing an external ID you can import the customer records in a single API call using the external ID references to the sales reps.

Important: External IDs can be updated through CSV or Web services. Therefore, it is recommended that your organization use a single approach for maintaining externalIds so that externalIds are not unknowingly updated using two separate methods.

Although records of a particular type may be used in multiple integration scenarios, each record instance can only have a single external ID value. In order to maintain data integrity, only a single integrated application can set and update external ID values for each record type. External ID values for all records of a particular type must all come from the same external application.

Most NetSuite record types support the use of external ID, but not all do. The following list identifies the record types that do not support external ID. For all other records, you should assume that external ID is supported. When in doubt, please refer to the appropriate page in the SuiteTalk Schema Browser. On each page describing a record type, view the Attributes table to see whether external ID is supported.

The record types that do not support external ID are:

- Accounting Period
- Budget Category
- CRM Custom Field
- Currency Rate
- Custom List
- Custom Record Custom Field
- Custom Record Type
- Entity Custom Field
- Gift Certificate
- Item Custom Field
- Item Number Custom Field
- Item Option Custom Field
- Landed Cost
- Other Custom Field
- State
- Transaction Body Custom Field
- Transaction Column Custom Field

Shared Internal and External IDs

System-generated internal IDs and custom external IDs can be shared among records belonging to the same high-level group. Therefore, when referencing a record using RecordRef, providing the system internal ID or custom external ID without specifying the record type is sufficient to uniquely identify the record within a given group.

The following list provides examples of these high-level groups and some of the records belonging to each group:

- Entities: contact, customer, employee, group, partner, vendor
- Transactions: invoice, journal entry, customer deposit, check
- Items: inventory item, markup item, payment item, serialized inventory item
- Activities: task, event, phone call

- Support: campaign, case, event, solution, task

Working with Fields in Web Services

NetSuite records contain standard body fields and custom fields. Standard fields are those that come with all NetSuite accounts. In Web services, a record's standard body fields are its attributes, for example: `Customer.email, SalesOrder.salesRep`. Standard fields must be of one of the following logical types.

Type	Description
String	Corresponds to the xsd:string type in the XML Schema
Int	Corresponds to the xsd:int type in the XML Schema
Double	Corresponds to the xsd:double in the XML Schema
Boolean	Corresponds to the xsd:boolean type in the XML Schema and has valid values of true or false. If not explicitly set to either true or false, then set as false.
Datetime	Corresponds to the xsd:dateTime type in the XML Schema which conforms to the ISO 8601 standard.
RecordRef	Corresponds to the RecordRef type in the XML Schema. References an nsKey value for any other record in the system including system defined constants that are controlled by the system.
Enum	Corresponds to a specific type defined as an enum in the XSD that represents system constants that are also available in the UI.
WsEnum	Corresponds to a specific type defined as an enum in the XSD that represents system constants that are NOT available in the UI.
List	A List references a type that is a list and should be explicitly defined in the in the XML Schema as a type. A list can either be null (if it's an optional field), or it must contain at least one entry unless otherwise noted.

Custom fields are those that have been created by NetSuite users to customize their accounts. Custom fields can be added to an account using SuiteBuilder point-and-click customization tools. They can also be added through the SuiteTalk add operation. Custom fields must be one of the types defined in Custom Field Types. Note that on business records, custom fields are contained in the customFieldList property. For more details, see CustomFieldList.

Also note that if you are getting unusual results in your Web services queries, it may be that a standard field has been customized. Additionally, a record can have any number of custom fields that may differ in their behavior when manipulated through Web services, versus when they are manipulated through the NetSuite UI.

The following are some general guidelines to consider when working with both standard and custom fields.

- Regardless of whether a **standard** field is set as mandatory in a custom form, the “requiredness” of the field is the same as it was originally (entry forms only — transaction forms honor custom requiredness settings). For more information on NetSuite form types, see the help topic *Custom Forms*.

- Customizations made to **standard** fields are not honored in Web services. The field behaves as it would with no customization done. For example, even if a standard field is set as disabled, it is still settable via Web services.
- If a **standard** field is set to NOT show in the UI, it is still settable via Web services. In the UI, these fields are not settable although the values are returned in search results.
- **Custom** display only (inline) and disabled fields are settable even though these are NOT settable through the UI. For information on field display types, see the help topic *Setting Display Options for Custom Fields*.
- The behavior of **custom** hidden fields varies depending on the endpoint being used. If you are the current endpoint, or any endpoint since 2011.2 endpoint (including 2011.2), these fields are not gettable. If you are using an older endpoint, they are gettable. By comparison, in the UI, these values cannot be seen.

  If you are using the 2011.1 endpoint or earlier, be aware that the behavior of the get operation is affected by the ignoreReadOnlyFields preference. If the preference is set to true, custom hidden fields and other read-only fields are not returned.

  Note also that custom hidden fields cannot be set (either in the user interface or using Web services, regardless of the endpoint).
- Defaulted fields that are set to blank on a Web services update will stay blank on update.

Note: If you are unfamiliar with NetSuite custom fields, it is recommended that you see the help topic *Custom Fields*. This section describes the purpose and general characteristics of each custom field type, which will help you when working with the SuiteTalk customization API (described in *Customization*).

CustomFieldList

Most business records that are exposed through Web services have one or more custom fields attached to it. These custom fields are contained in the customFieldList property on each record. The customFieldList property is an array of CustomFieldRef.

Note: The Disable Mandatory Custom Field Validation preference determines whether a required custom field with no data provided throws an error, or is accepted as a null value. For more information on this preferences, see Setting Company-wide Preferences.

Field Name	Type	Req.	Default	Notes
customField	varies	Yes		Value of the custom field. Points to a type of CustomFieldRef in the XML Schema which is an abstract type.

Field Name	Type	Req.	Default	Notes
internalID	string	Yes		The field instance internal ID
recType	string	No		The record type id
xsi:type	xsi:type	Yes		This is a field that is automatically implemented by the XML Schema. The value should represent the concrete custom field type.

The following is an example that contains an excerpt of the SOAP body for a list of custom fields. It shows that the record associated with the SOAP contains two custom fields, both of which are referenced in the customFieldList property on the record.

```
<customFieldList>
 <customField internalId="CUSTEVENT1" xsi:type="IntCustomFieldRef">
  <value>12</value>
 </customField>
 <customField internalId="CUSTEVENT2" xsi:type="MultiSelectCustomFieldRef">
  <value>peter</value>
  <value>paul</value>
  <value>mary</value>
 </customField>
</customFieldList>
```

Setting Custom Fields to NULL

Custom fields can only be set to NULL by submitting the field in nullFieldList. For example, to set a custom field on a customer record to null, submit the following SOAP request, where custEntity9 is the custom field ID and 373 is the specific instance of the customer record:

```
<soap:Body>
 <platformMsgs:update>
  <platformMsgs:record internalId="373" xsi:type="listRel:Customer">
   <platformCore:nullFieldList xsi:type="platformCore:NullField">
    <platformCore:name>custEntity9</platformCore:name>
   </platformCore:nullFieldList>
  </platformMsgs:record>
 </platformMsgs:update>
</soap:Body>
```

Note: You cannot set the Custom Form field to NULL. (This field is available on transaction and entry forms to indicate the form that should be used.) Any request to set it this field to NULL with nullFieldList will be ignored.

Sample .NET Code

```
Customer cust = new Customer();
cust.internalId = "373";
cust.nullFieldList = new String[]{"custentity9"};
_service.update(cust);
```

Sample Java Code

```
Customer cust = new Customer();
cust.setInternalId("373");
NullField nfl = new NullField(new String[]{"custentity9"});
cust.setNullFieldList(nfl);
port.update(cust);
```

Custom Fields and Joined Searches

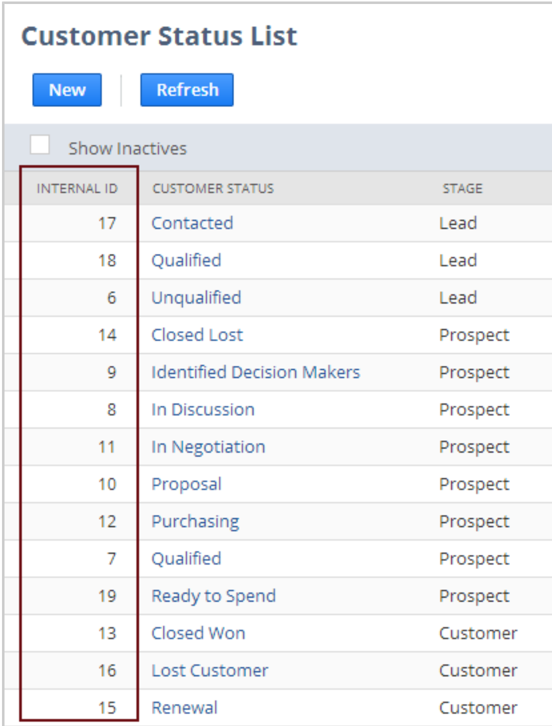
When a custom field is used to create a parent/child relationship between two existing records, the resultant joined searches available through the NetSuite UI are NOT supported in Web services.

User Defined Lists

In SuiteTalk, many fields require internal ID values that correspond to an item in a user-defined list. To locate the internal ID of a specific value in a user-defined list, you must have the **Show Internal IDs** preference enabled in your account. For details, see Enabling the Show Internal IDs Preference.

After enabling the **Show Internal IDs** preference, you can then navigate to the appropriate list within NetSuite. The internal ID values (also referred to as nsKeys) are displayed for each list item.

For example, the Status (*entityStatus*) field on the Customer record takes an internal ID value from the Customer Status list, which is a user-defined list located at Setup > Sales > Customer Statuses. If you navigate to this page in NetSuite, you can see the internal ID values for each item in the column called **Internal ID** (see figure).



INTERNAL ID	CUSTOMER STATUS	STAGE
17	Contacted	Lead
18	Qualified	Lead
6	Unqualified	Lead
14	Closed Lost	Prospect
9	Identified Decision Makers	Prospect
8	In Discussion	Prospect
11	In Negotiation	Prospect
10	Proposal	Prospect
12	Purchasing	Prospect
7	Qualified	Prospect
19	Ready to Spend	Prospect
13	Closed Won	Customer
16	Lost Customer	Customer
15	Renewal	Customer

Note: Some user-defined lists can also be edited through Web services to modify the values for the list. For details, see the help topic *Other Lists*.

Enumerated Lists

If values being returned for WSEnum and Enum fields in a get or search operation do NOT match the values enumerated in the schema, the following warning is returned:

```
Code = Invalid_data
Message = "Error in record number <id>: Invalid field value <field>.
Please refer to the XSD for enumerated list of valid field values."
```

In these cases, the existing data is corrupt and must be corrected before it can be retrieved using Web services.

Field Lengths

Most string fields have a maximum character limit. If the limit is exceeded, an error is returned indicating which field exceeded the limit and the maximum number of characters allowed for that field.

Field Level Errors

If a required field is missing for a given record within a request, an error is generated. The associated record is not processed and an appropriate error status code is returned.

Only the records without errors are processed. If multiple records are submitted within the same request, records without errors are processed; records with errors are not processed.

Required Fields

Required fields in the UI do not necessarily correspond to required fields in Web services. This is because there can be standard defaults that are applied if the field is not populated. For example, in the CalendarEvent record type, the *eventAccess* field is required in the UI. However, it is optional in Web services because it has a default value of Public.

Important: The **Required** column in the SuiteTalk Schema Browser lists T or F (for true or false) to specify if a field is required by the form in the user interface. Refer to the **Cardinality** column to see whether the XSD imposes a requirement for the field. For information on using the SuiteTalk Schema Browser, see the help topic *Using the SuiteTalk Schema Browser*.

Fields and Operations

For simplicity, the SuiteTalk WSDL is designed with all available fields listed for each record in each corresponding XSD. There is no differentiation as to what field is available for each individual operation. For example, add operations take a separate set of field values than the corresponding update operation for some records. If your Web services request includes a field value for a field that is unavailable, an error is thrown for that submission.

If you are using Web services for data migration where there may be fields that need to be populated that are unavailable during an add operation, you should perform two consecutive requests. Submit an initial add or addList request, with values for all fields available for an add operation, followed by an update or updateList request, with values for the fields available only during an update operation.

Default Field Values

The system provides default values only for fields that are not required. When applying a default value, the system first tries to use a specified value. If none is given, the system uses the default. If no default is given, the system enters null.

User-defined defaults through the UI do not apply to Web services.

Hidden Fields

The SuiteTalk API includes fields that are not visible in the NetSuite UI. In some cases, these fields have been hidden through SuiteBuilder point-and-click customization. In other cases, hidden fields are just hidden by NetSuite. These types of hidden fields are used primarily for storing system-generated information such as dates.

For example, every SuiteTalk-support record contains either a *dateCreated* or *createdDate* field. These fields are not writeable. By default, these fields are populated with a system-generated timestamp. Note that system-generated dates and times are preserved for audit trail purposes.

Note: The Use Defaults Web service preference does not affect the behavior of hidden fields. Even if the Use Defaults preference is not enabled, hidden fields are populated when a value is not supplied in the request. Also, for audit purposes, the true system date is always recorded for each record and cannot be overridden.

Important: Calls to get or search (when the full record is returned) return the record in **edit** mode.

Working with Forms in Web Services

If you are building a generic SuiteTalk application, in order to ensure that your applications are account independent, NetSuite recommends that you create **custom "Web-services-only" forms**. Use custom forms for Web services requests by specifying the form in the *customForm* element of a record. This will alleviate problems associated with customer-specific customizations to forms used in the UI that may break your Web services application code.

Note: If you are unfamiliar with NetSuite custom forms, see the help topic *Custom Forms*. For information on creating custom forms that you can designate as "web-services-specific," see the help topic *Creating Custom Entry and Transaction Forms*.

For all record types that support custom forms, if a form is specified in the Web services request, that form will be used for validation. Note that it is generally best practice to specify the custom form in all Web services update operations. Also be aware that no record type is needed when specifying a custom form. You only need to specify the internal ID of the custom form.

To get the internal ID of a custom form, in the UI, go to Setup > Customization > [*form type*], where form type is either an entry or transaction form. The internal ID appears in the Internal ID column if the Show Internal IDs preference is enabled. (For steps on enabling this preference, see Enabling the Show Internal IDs Preference.)

To specify a form in Web services, see the following C# snippet, which shows how to associate a particular form with a Customer record.

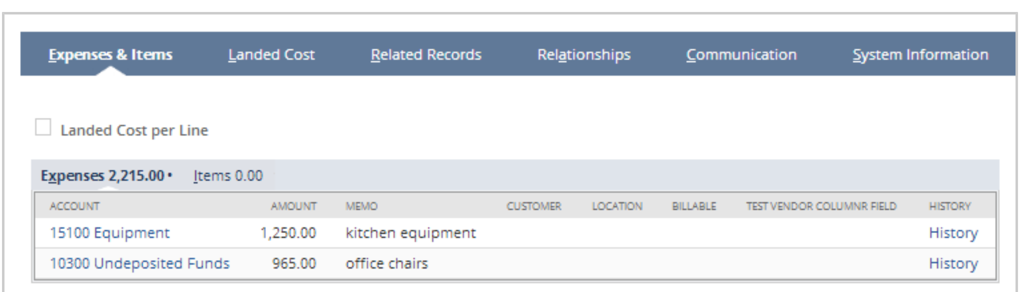
```
// create a customer object

Customer customer = new Customer();

customer.internalId = "555"

// create a custom form object

RecordRef customFormRef = new RecordRef();

// set the internal ID of the custom form. Get the internal ID from
the UI

customFormRef.internalId = "-100";

// set the customForm field on the Customer record to reference the
form

customer.customForm = customFormRef;
```

If a form is NOT specified, then the default preferred form for that record is used. If a custom form is saved with a record in the UI, that form is not used in the Web services request unless it is also the default form for that record type or is explicitly set as the form in the Web services request.

Working with Sublists in Web Services

Most records in NetSuite include sublists, which generally contain a list of references to other records. This figure shows the Expenses sublist on the Check record. Notice that this sublist includes two lines in the Expenses sublist.

Note: For general information on sublists in NetSuite, see the help topic *What is a Sublist?*



In Web services, sublists are represented in a record's schema definition by its **List** elements.

Note: Except for the customFieldList element (which contains a record's custom fields), all other elements that end in **List** represent a record's sublists.

The following is a portion of the schema for the Check record. (Although the XSD shown below describes the Items sublist on the Check record, the same pattern for defining sublists applies to all sublists on all records.)

Notice that the sublists on the Check record include the Expenses sublist, the Items sublist, and the Landed Costs sublist (not shown in the figure above because the Landed Costs feature is not enabled in the account).

```
 <complexType name="Check">
        <complexContent>
            <extension base="platformCore:Record">
                <sequence>
                    <element name="createdDate" type="xsd:dateTime" minOccurs="0"/>
                    <element name="address" type="xsd:string" minOccurs="0"/>
                    <element name="subsidiary" type="platformCore:RecordRef" minOccurs="0"/>
     ........
                    <element name=" expenseList " type="tranBank:CheckExpenseList" minOccurs="0
"/>
                    <element name=" itemList " type="tranBank:CheckItemList" minOccurs="0"/>
                    <element name=" landedCostsList " type="tranBank:CheckLandedCostList" minOc
curs="0"/>
                    <element name="billPay" type="xsd:boolean" minOccurs="0"/>
                    <element name="customFieldList" type="platformCore:CustomFieldList" minOccu
rs="0"/>
                </sequence>
              </extension>
        </complexContent>
    </complexType>
```

The **itemList** element (listed above) is of type **CheckItemList**, which is defined in the **CheckItem** complex type (defined in transactions.bank.xsd). The **CheckItem** type (shown below) includes all the properties that can be set for the Item sublist on the Check record. Also notice the replaceAll attribute in CheckItemList. You will set the value of replaceAll to true or false depending on whether you are working with keyed sublists or non-keyed sublists. See Updating Sublists in Web Services for details.

```
<complexType name=" CheckItem ">
        <sequence>
            <element name="item" type="platformCore:RecordRef" minOccurs="0"/>
            <element name="vendorName" type="xsd:string" minOccurs="0"/>
  ....

            <element name="line" type="xsd:long" minOccurs="0"/>
            <element name="location" type="platformCore:RecordRef" minOccurs="0"/>
            <element name="isBillable" type="xsd:boolean" minOccurs="0"/>
            <element name="customFieldList" type="platformCore:CustomFieldList" minOccurs="0"/>

        </sequence>
    </complexType>
    <complexType name=" CheckItemList ">
        <sequence>
            <element name="item" type="tranBank:CheckItem" minOccurs="0" maxOccurs="unbounded"/
```

```
>
        </sequence>
        <attribute name=" replaceAll " type="xsd:boolean" default="true"/>
    </complexType>
```

Updating Sublists in Web Services

When working with sublists, the approach for updating the lines in a sublist will vary depending on whether you are working with Keyed Sublists or Non-Keyed Sublists.

Keyed Sublists

Keyed sublists have an indexing line element or internal ID that can be used as a key. Keyed sublists allow you to set the **replaceAll** attribute to FALSE to update only the lines you are submitting in your SOAP request. When working with keyed sublists, the value of the replaceAll attribute has the following effects:

- replaceAll = TRUE: The existing sublist is **replaced** with the sublist submitted in the Web services request. The newly submitted sublist should include all values.
  - Lines that do not match the newly submitted lines are removed.
  - Currently existing lines that match lines in the new sublist submission are updated.
  - Newly submitted lines with no matches are added.

  The default value for the replaceAll attribute is TRUE.
- replaceAll = FALSE: Lines in the existing sublist are **selectively updated** with lines in the sublist submitted in the Web services request. The newly submitted sublist need only include values to be added and updated.
  - Lines that do not match the newly submitted lines are preserved.
  - Currently existing lines that match lines in the new sublist submission are updated.
  - Newly submitted lines with no matches are added.

The following table lists sublists that are categorized as keyed sublists in Web services. Sublists that do not appear in this table are considered Non-Keyed Sublists. The sublist key listed for each keyed sublist below is used to link the sublist data with the appropriate record.

Important: PricingMatrixList and demandPlanDetailList are matrix sublists. Updates to matrix sublists may require additional handling related to the replaceAll attribute. See Matrix Sublists and replaceAll.

Sublist	Sublist Key	Appears on Record(s)
accruedTimeList	payrollitem	Employee
addressbookList	internalId	Contact, Customer, Employee, Lead, Partner, Project, Prospect, Vendor

Sublist	Sublist Key	Appears on Record(s)
		Note: When you work with the addressbookList on customer records, be sure to use **internalId** as the key. Do not use **label**.
applyList	doc	Customer Payment, Credit Memo, Customer Refund, Customer Deposit, Deposit Application, Vendor Credit, Vendor Payment
assigneeList	resource	Project Task
attendeeList	attendee	Calendar Event
binNumberList	binNumber	Assembly Item (BOM Item), Inventory Item, Lot Numbered Assemby Item, Lot Numbered Inventory Item, Serialized Assembly Item, Serialized Inventory Item
campaignDirectMailList	internalId	Campaign
campaignEmailList	internalId	Campaign
campaignEventList	internalId	Campaign
companyContribution List	payrollitem	Employee
componentList	item	Assembly Unbuild, Assembly Build
contactList	company	Customer, Partner, Vendor
creditList	doc	Customer Payment, Vendor Payment
creditCardsList	internalId	Customer, Job
currencyList	currency	Customer (when Multi-Currency Customer enabled)
deductionList	payrollitem	Employee
depositList	doc	Customer Payment, Customer Refund
deptAccessList	dept	Custom Entity Field, CRM Custom Field, Other Custom Field, Item Custom Field, Transaction Body Custom Field, Transaction Column Custom Field, Item Option Custom Field, Custom Record Custom Field, Item Number Custom Field
directDepositList	internalid	Employee
earningList	payrollitem	Employee
expCostList	doc	Invoice, Cash Sale
expenseList	orderLine or line	Check, Expense Report, Item Receipt, Purchase Order, Vendor Bill, Vendor Credit, Vendor Return Authorization
itemList	orderLine or line	Bin Putaway Worksheet, Bin Transfer, Cash Sale, Check, Estimate, Inventory Transfer, Invoice, Item Fulfillment, Item Receipt, Credit Memo, Return Authorization, Cash Refund, Opportunity, Purchase Order, Sales Order, Vendor Bill, Vendor Credit, Vendor Return Authorization, Work Order *Notes:* On Item sublists, the value **End of Group** is not returned in Web services. This behavior mimics the behavior of the UI. For Cash Sale records, the orderLine field establishes the relationship with an existing Sales Order, if any. You must set a value for orderLine in order to populate the createdFrom field.

Sublist	Sublist Key	Appears on Record(s)
		For Item Receipt records, orderLine is used for transforms, because it implies a link between the previous transaction and the current one. For example, to add an item receipt from a purchase order, the purchase order lines would be "orderLines", because the receipt has not been saved. Once the item receipt is saved, lines should be accessed via the "line" field. For Opportunity records, line is used.
itemCostList	doc	Invoice, Cash Sale
itemsList	item	Promotion Code, Item Option Custom Field, Item Number Custom Field.
locationsList	locationId	Inventory Item, Serialized Inventory Item, Lot Numbered Inventory Item
milestoneList	milestoneId	Billing Schedule **Note:** This sublist is available only when scheduleType is set to Fixed Bid Milestone.
nexusesTaxList	nexus	Tax Type
partnersList	partner	Estimate, Cash Sale, Invoice, Sales Order, Opportunity, Credit Memo, Return Authorization, Cash Refund, Customer
partnersList	partner	Promotion Code
paymentList	The internal ID of the record being referenced in the sublist line.	Deposit
periodDemandPlanList	startDate	Item Demand Plan **Note:** This sublist includes additional handling for replaceAll. See Matrix Sublists and replaceAll.
predecessorList	task	Project Task
pricingMatrixList	Varies according to enabled features. See the help topic *Pricing Matrix Keys*.	Item **Note:** This sublist includes additional handling for replaceAll. See Matrix Sublists and replaceAll.
ratesList	subsidiary	Expense Category **Note:** Sublist available only to OneWorld accounts.
recurrence	recurrenceId	Billing Schedule **Note:** This sublist is available only when scheduleType is set to Standard and frequency is set to Custom.
resourceList	resource	Calendar Event
roleAccessList	role	Entity Custom Field, CRM Custom Field, Other Custom Field, Item Custom Field, Transaction Body Custom Field, Transaction Column

Sublist	Sublist Key	Appears on Record(s)
		Custom Field, Item Option Custom Field, Custom Record Custom Field, Item Number Custom Field
salesTeam	employee	Customer, Invoice, Cash Sale
timeList	doc	Invoice, Cash Sale

Example

The following sample shows how to add an item to an opportunity record and modify an existing item in that record. To do this, set replaceAll to FALSE, add the new item, and modify the existing item by referencing the desired item in the line property. The following shows that the existing line item is modified to change the quantity.

```
<soapenv:Body>
<platformMsgs:update xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:s0="urn:sales_2011_2_0.transactions.webservices.netsuite.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:platformMsgs="urn:messages_2_5.platform.webservices.netsuite.com">
<platformMsgs:record xsi:type="s0:Opportunity" internalId="1638">
 <s0:itemList replaceAll="false">
  <s0:item>
   <s0:item internalId="380" type="inventoryItem" />
   <s0:quantity>1.0</s0:quantity>
   <s0:amount>20.0</s0:amount>
  </s0:item>
  <s0:item>
   <s0:line>2</s0:line>
   <s0:quantity>10.0</s0:quantity>
  </s0:item>
 </s0:itemList>
</platformMsgs:record>
</platformMsgs:update>
</soapenv:Body>
```

Matrix Sublists and replaceAll

Matrix sublists respect the replaceAll attribute for updates, as other Keyed Sublists do, but in some cases, updates to matrix sublists require additional handling.

Matrix sublists require additional handling for replaceAll because body-level fields slave values into them. For example, setting the pricing schedule for an item dictates values in the pricing matrix. In the UI, when a pricing schedule is selected, the pricing schedule logic is run first, and values specified in the pricing matrix sublist are taken into account to arrive at what the update should do. However, the UI does not have the notion of replaceAll because the user interacting with the form can delete values they don't want to keep in individual fields.

Web services requests cannot delete individual field values in a sublist. The only way to get rid of old values is to set replaceAll to TRUE and send all values in the update request. Some body fields can override values in a matrix sublist, but a Web services request can then change some or all of these sublist values, depending on the replaceAll setting.

Review the following cases to get an understanding of how the slaving of values from body fields into a matrix sublist interacts with the replaceAll setting on updates:

- No sublist values from body fields and replaceAll=TRUE

  The Web services request does not include any body field values that drive values in the sublist, so the update is handled like those in other keyed sublists where replaceAll is set to TRUE.

  The existing sublist is **replaced** with the sublist submitted in the Web services request. The newly submitted sublist should include all values.

- No sublist values from body fields and replaceAll = FALSE

  The Web services request does not include any body field values that drive values in the sublist, so the update is handled like those in other keyed sublists where replaceAll is set to FALSE.

  Lines in the existing sublist are **selectively updated** with lines in the sublist submitted in the Web services request. The newly submitted sublist need only include values to be added and updated.

- Sublist values from body fields and replaceAll=TRUE

  The Web services request sets body field value(s) that drive values in the sublist, so specialized handling is required.

  First, all sublist values are removed. Next, sublist values driven by body field values are set. Then, the sublist is updated with all lines specified in the newly submitted sublist. The newly submitted sublist lines thus override the values driven by body fields.

Important: When sublist values from a Web services request override values driven by body fields, errors may occur. If you do not want these overrides, you must set replaceAll to FALSE. For example, if you are setting a quantity pricing schedule, you should set ReplaceAll to FALSE.

- Sublist values from bodyfields and replaceAll=FALSE

  The Web services request sets body field value(s) that drive values in the sublist, so specialized handling is required.

  First, sublist values driven by body field values are set. Then, the sublist is updated with any updated or added lines from the newly submitted sublist.

Note: By default, replaceAll is set to TRUE for matrix sublists, just like other sublists.

Currently Supported Matrix Sublists

The pricingMatrixList on Items and periodDemandPlanList on Item Demand Plans are currently supported matrix sublists. For details about these sublists, see the help topics *Pricing Matrix List* and *Item Demand Plan*.

Important: In endpoints prior to 2012.1, updates to pricingMatrixList ignore the replaceAll attribute. In these endpoints, this sublist is treated like , and all requests behave as if replaceAll is set to TRUE. For the 2012.1 and later endpoints, replaceAll is respected for this sublist.

Non-Keyed Sublists

Non-keyed sublists have no indexing line element or internal ID that can be used as a key. Each line in a non-keyed sublist is recorded in a *flat* list.

In the UI, non-keyed sublists and keyed sublists may look similar; however, technically, they are quite different. Because non-keyed sublists contain no referencing keys (or handles), you cannot update a specific line in a non-keyed sublist. Instead, you must interact with the **sublist as a whole.** In non-keyed sublists, the **replaceAll** attribute is ignored and behaves as if it were set to TRUE for all requests. Consequently, an update operation is similar to the add operation with respect to non-keyed sublists.

In the context of Web services, all sublists not listed as Keyed Sublists are considered to be non-keyed sublists. The following bullets outline the behaviors of non-keyed sublists.

- To update a single line in the sublist, retrieve the entire sublist, change a single line as desired, and then resubmit the entire sublist.
- To replace the sublist with a subset of lines, retrieve the entire sublist, and then resubmit the subset of lines. The original sublist is purged and replaced by the new sublist containing the subset of lines.
- To delete the sublist, submit an empty sublist. See Deleting All Lines on a Sublist for more information.

Referencing Sublist Line Numbers

The index number for each sublist line is assigned by the system. Web services developers cannot change these values. When lines are added or removed from a sublist (using either the UI or Web services), the system does not re-indexed the line numbers. Therefore, it is best practice to do a get on the sublist data before trying to update individual lines, otherwise you may be updating the wrong line in the sublist.

For example, upon doing a get on an Items sublist on a transaction record, the index values for the first three line items might actually appear be:

<ns3:line>1</ns3:line>
<ns3:line>8</ns3:line>
<ns3:line>9</ns3:line>

Deleting All Lines on a Sublist

This sample shows how to delete all addresses from the Address sublist on a customer record (CustomerAddressbookList).

Java

```
Customer update = new Customer();
update.setInternalId(c.getInternalId());
cabl = new CustomerAddressbookList();
cabl.setAddressbook(new CustomerAddressbook[0]);
cabl.setReplaceAll(true);
update.setAddressbookList(cabl);
```

SOAP

```
<update xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
            <record internalId="724" xsi:type="ns1:Customer" xmlns:ns1="urn:relationships_2_5.l
ists.webservices.netsuite.com">
               <ns1:addressbookList replaceAll="true" xsi:type="ns1:CustomerAddressbookList"/>
            </record>
         </update>
```

The next sample shows how to delete all line items from the Item sublist (OpportunityItemList) on an Opportunity record.

Java

```
Opportunity update = new Opportunity();
update.setInternalId(opp.getInternalId());
update.setItemList(new OpportunityItemList(new OpportunityItem[0], true));
```

SOAP

```
<update xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
            <record internalId="6147" xsi:type="ns1:Opportunity" xmlns:ns1="urn:sales_2_5.trans
actions.webservices.netsuite.com">
               <ns1:itemList replaceAll="true" xsi:type="ns1:OpportunityItemList"/>
            </record>
         </update>
```

Searching a Sublist

Keyed sublists have a **line** field (typically on transaction sublists). Individual lines in the list can be searched for by referencing the line value.

Also note that when searching records, by default sublists are NOT returned in the result set, which increases the search response time. However, if you want to return all sublist data in your search, set the SearchPreferences. *bodyFieldsOnly* preferences to FALSE in your search request. For more details, see Setting Search Preferences.

Chapter 6 Web Services Processing

This section describes synchronous and asynchronous processing. Also provided are steps for monitoring your web services requests. Refer to the following topics:

- Synchronous Versus Asynchronous Request Processing
- Asynchronous Request Processing
- Synchronous Request Processing

Synchronous Versus Asynchronous Request Processing

Web services requests can be processed synchronously or asynchronously.

With synchronous requests, your client application sends a request to NetSuite, and the client waits until the request is processed and a response is returned. That is, the application does not proceed with other work until receiving the response.

With asynchronous requests, your client application sends a request to SuiteTalk, where it is placed in a processing queue and handled asynchronously with other requests. Your client application does not wait for a response but goes on to other work. Once a job is submitted, a job Id is returned in the Web services response. Your client application can then check on the status and result of the request by referencing the job Id.

Asynchronous processing may be advantageous in the following situations:

- If you expect your connection to NetSuite to be slow or unstable.
- If your job is large, and its processing can be postponed until off-peak hours.

Be aware that asynchronous responses may not be returned immediately. Before committing to using asynchronous operations, you should consider this factor and decide whether it fits in with your business logic.

Asynchronous Request Processing

This section describes asynchronous request processing in more detail.

Available Operations

Asynchronous equivalents are available for several synchronous operations, as shown in the table below.

Asynchronous Operation	Synchronous Equivalent
asyncAddList	addList

Asynchronous Operation	Synchronous Equivalent
asyncUpdateList	updateList
asyncUpsertList	upsertList
asyncDeleteList	deleteList
asyncGetList	getList
asyncSearch	search
asyncInitializeList	initializeList

Understanding Asynchronous Requests and Responses

When submitting an asynchronous request, the request is similar to the equivalent synchronous call. The following code illustrates an asynchronous request for a list of customer records.

```
<soap:Body>
 <platformMsgs:asyncGetList>
  <platformMsgs:baseRef internalId="87" type="customer" xsi:type="platformCore:RecordRef"></pla
tformMsgs:baseRef>
  <platformMsgs:baseRef internalId="176" type="customer" xsi:type="platformCore:RecordRef"></pl
atformMsgs:baseRef>
 </platformMsgs:asyncGetList>
</soap:Body>
```

The response, however differs in that only information about the request is returned. This information includes a unique identifier for the request, which is enclosed in the jobId tags.

```
<soapenv:Body>
 <asyncGetListResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
  <asyncStatusResult xmlns="urn:core_2_5.platform.webservices.netsuite.com">
   <jobId>ASYNCWEBSERVICES_563214_053120061943428686160042948_4bee0685</jobId>
   <status>pending</status>
   <percentCompleted>0.0</percentCompleted>
   <estRemainingDuration>0.0</estRemainingDuration>
  </asyncStatusResult>
 </asyncGetListResponse>
</soapenv:Body>
```

Later, you can use the checkAsyncStatus or getAsyncResult operations to track the asynchronous request. To identify the specific request about which you need data, refer to it by its job Id.

Usage Notes

Be aware of the following characteristics of asynchronous processing:

- All available jobs are processed contiguously. That is, processing priority is based on when the job is submitted, and jobs can't pass one another in the queue.
- The system does not enforce waiting periods for jobs. That is, as soon as soon as a job is completed, the next job from the queue is immediately selected and the processing starts.

- Asynchronous request job Ids are valid for 30 days. After 30 days, you can no longer look up the request.

Checking the Status of an Asynchronous Job

To check the status of an asynchronous Web services job, you use the checkAsyncStatus operation. When you use this operation, you reference a job Id. In response, the system returns the job's status, among other details.

Specific status values that can be returned include the following:

- failed
- finishedWithErrors — indicates that at least one record was processed, but at least one failed.
- pending
- processing
- finished

If the status is failed, finishedWithErrors, or finished, you can in most cases use the getAsyncResult operation to obtain more details. These details may include the record results, or error and fault messages. The getAsyncResult operation is described in Checking for Detailed Results of an Asynchronous Job.

checkAsyncStatus also returns details about the percentage of the job that is complete, and the estimated time remaining.

checkAsyncStatus Request and Response

The following is a typical checkAsyncStatus request.

```
<soap:Body>
 <platformMsgs:checkAsyncStatus>
  <platformMsgs:jobId>ASYNCWEBSERVICES_563214_053120061943428686160042948_4bee0685
  </platformMsgs:jobId>
 </platformMsgs:checkAsyncStatus>
</soap:Body>=
```

The following is a typical checkAsyncStatus response.

```
<soapenv:Body>
<checkAsyncStatusResponse
 xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
 <asyncStatusResult xmlns="urn:core_2_0.platform.webservices.netsuite.com">
  <jobId>ASYNCWEBSERVICES_563214_053120061943428686160042948_4bee0685</jobId>
  <status>pending</status>
  <percentCompleted>0.0</percentCompleted>
```

```
  <estRemainingDuration>0.0</estRemainingDuration>
 </asyncStatusResult>
</checkAsyncStatusResponse>
</soapenv:Body>
```

checkAsyncStatus Faults

The checkAsyncStatus operation can throw any of the following faults:

- InvalidSessionFault
- AsyncFault
- UnexpectedErrorFault

For more information on faults, see SOAP Fault Status Codes.

Checking for Detailed Results of an Asynchronous Job

To retrieve details about specific records submitted as part of an asynchronous request, you use the getAsyncResult operation.

getAsyncResult Time Limits

Be aware of the following limits:

- You can use the getAsyncResult operation for a particular job Id up to 20 times within a 30-day time period. If this limit is exceeded, the system returns an INVALID_JOBID user error.
- Async job IDs are purged from the system every 30 days. If you attempt to download the results 31 days after your initial request, the system returns an INVALID_JOBID user error.

getAsyncResult Example

You can use code like the following to retrieve the getAsyncResult response. This example shows how to look for records that failed with a status detail code of “unexpected error” and then take some action with those records.

```
AsyncAddListResult result = (AsyncAddListResult)port.getAsyncResult("job ID", 1);
WriteResponseList responses = result.getWriteResponseList();
for (WriteResponse response : responses.getWriteResponse())
{
 if (!response.getStatus().isIsSuccess())
 {
  if (response.getStatus().getStatusDetail()[0].getCode() == StatusDetailCodeType.UNEXPECTED_ER
ROR)
  // add for resubmit
 }
```

```
}
```

Note: In order to use getAsyncResult with asyncInitializeList, you must be using the 2013.2 endpoint or later. You can monitor asyncInitializeList requests using the UI, as described in Monitoring Asynchronous Jobs from the UI.

getAsyncResult Faults

This operation can throw one of the following faults.

- InvalidSessionFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- AsyncFault
- UnexpectedErrorFault

For more information on faults, see SOAP Fault Status Codes.

Note: You can see some details about asynchronous job results by navigating to Setup > Integration > Web Services Process Status and viewing the Response.xml file for the job. For details, see Monitoring Asynchronous Jobs from the UI.

Monitoring Asynchronous Jobs from the UI

Web services jobs submitted asynchronously can be monitored at Setup > Integration > Web Services Process Status.

On this page, you can view the following information about each job:

- Date — the date the job was created
- Sequence — the order of the job in relation to other asynchronous jobs in process (jobs are processed on a first in, first out basis)
- Job Number — order number of the job in the work queue
- Job Id — the Job Id, which is also returned in the asynch SOAP response, and can be used to programmatically retrieve job status information
- Status — the status of the job: failed, finishedWithErrors, pending, processing, or finished (for more on status, see Checking the Status of an Asynchronous Job)

- Percent Complete — the percentage of job processing that has been completed
- Est. Remaining Time — the estimated amount of time remaining before the job is completed
- Request — the SOAP request associated with the job (see additional details below)
- Response — the SOAP response associated with the job (see additional details below)
- Cancel — you can check the box in this column to cancel a job that has not yet successfully completed

Limits on Requests and Responses

Note the following limitations:

- A response for a job is available only after the job has successfully completed.
- Lengthy requests and responses are not saved by the system and therefore are not available for your review. The system imposes the following limits:
  - The system does not save requests that are larger than 100MB.
  - The system does not save responses that are large than 1MB.

Refreshing the Job Status Page

The Job Status page does not automatically refresh. To get the current status of a job, click the Refresh button at the top of the page.

Synchronous Request Processing

In addition to using the responses NetSuite sends for each request, you can monitorn synchronous Web services processing in the NetSuite user interface. You have the following options:

- Using the Web Services Usage Log
- Creating Integration Reports

Note: Of these two options, the Web Services Usage Log is the generally recommended one, because it requires no setup.

See also Handling of Lengthy Requests, which describes limitations that exist for sending responses.

For information on monitoring asynchronous requests, see Monitoring Asynchronous Jobs from the UI.

Using the Web Services Usage Log

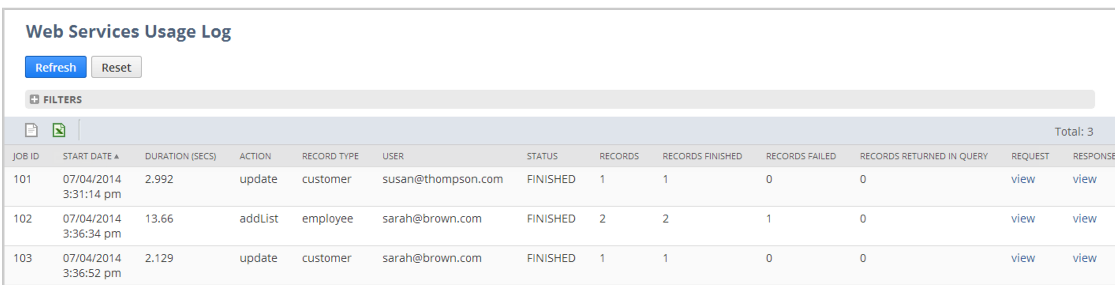
NetSuite administrators can use the Web Services Usage Log to monitor synchronous requests. This page can be accessed at Setup > Integration > Web Services Usage Log.

The Web Services Usage Log allows administrators to easily filter their Web services requests by specified date and time periods, record type, and action (operation). The SOAP requests and responses for each job are accessible in the Request and Response columns. Requests and responses are accessible for **30 days**.

Results can be filtered down to the minute. The results returned provide second-level granularity.

Note that lengthy requests and responses are not saved by the system and therefore are not available for your review. The system imposes the following limits:

- The system does not save requests that are larger than 100MB.
- The system does not save responses that are large than 1MB.



Only administrators can access the Web Services Usage Log. Assigning the Web Services permission to a role does not provide access to this log. Access is provided to administrators only because access to this log could allow users to see data they do not have permission to view.

Note: The Web Services Usage Log does not show asynchronous requests. For details, see Monitoring Asynchronous Jobs from the UI.

Creating Integration Reports

In addition to the Web Services Usage Log, you can also monitor Web services processing by creating the following types of integration reports:

- Integration and Automation Usage Summary by Job

- Integration and Automation Usage Summary by Record Type

Important: The **Integration** permission is required to view these reports. To enable this permission, administrators must go to Setup > Users/Roles > Manage Roles. Click Customize next to the appropriate role. On the Permissions tab, click Reports. In the Permissions drop-down list, select **Integration** and click Add.

Note: You might receive a timeout error when creating integration reports for accounts with large volumes of Web services traffic. Also, although integration reports do offer *some* filtering capabilities, users must first take the added step of customizing the report. The Web Services Usage Log requires no such customization to enable filtering. Additionally, no timeout issues will occur when using the Web Services Usage Log. Therefore, it is generally recommended that administrators use the Web Services Usage Log to monitor Web services activities in their account.

Both of these reports provide details on type of operations used (for example, add, addList, delete), who performed an operation, and the time an operation was performed.

Web Services Integration reports are also useful for administrators who need to diagnose and troubleshoot issues. Note that each report can be customized to display only the desired information. To customize a report, click Customize next to the desired report and then modify the report as desired. For details about how to customize a report, see the help topic *Report Customization*.

If an integration report is timing out, it is recommended that you customize the report by adding filters so the report returns fewer results. See Filtering Integration Report Data.

Integration and Automation Usage Summary by Job

This report can be used for performance statistics. It shows the duration of each Web services processing job, the number of records modified in each job, the number of successful versus failed record modifications for each job, and the number of records queried for each job.

Any related SOAP files are stored in View links in the reports Request Doc and Response Doc columns. **Be aware that NetSuite purges SOAP files every 30 days.**

You can view this report at Reports > Integration > Integration and Automation Usage Summary by Job.

Integration and Automation Usage Summary by Record Type

This report lists the record types modified by Web services processing, and the number of records added, updated, deleted, and queried for each record type.

You can view this report at Reports > Integration > Integration and Automation Usage Summary by Record Type.

Filtering Integration Report Data

You can customize integration reports to run faster, by adding filters that limit the returned results. For example, you could limit a report to return only jobs with start dates that fall into a specified date range.

To filter an integration report by start date:

1. Go to Reports > Integration > Integration and Automation Usage Summary by Job > Customize or Reports > Integration > Integration and Automation Usage Summary by Record Type > Customize.
2. Click **Filters**.
3. Under **Search Fields** box, enter **Start Date** and click the **Search** button.

   The **Actual Job Start Date** and **Start Date** fields are listed.
4. Click **Start Date**.

   This field is added to the **Choose Filters** pane.
5. In the **Choose Filters** pane:
   1. Set **Filter** to between.
   2. Set **Date Range** to custom.
   3. In the **From** and **To** fields, enter dates in {mm/dd/yyyy} format.
   4. Click **Done**.
6. Click **Save**.

Handling of Lengthy Requests

Throughout NetSuite, the maximum duration for a single HTTP connection is 15 minutes. However, in some cases a Web services request may require more than 15 minutes to process. Because the system cannot send a response after the connection has timed out, the system handles lengthy requests differently from those that complete more quickly. The exact handling varies depending on the operation. For details, see the following:

- Single-Record Requests
- List and Search Requests

Single-Record Requests

If a synchronous single-record operation does not complete within 15 minutes, the operation is allowed to finish. In these cases, no response is sent, but the system updates the Web Services

Usage Log with a partial response. However, note that single-record operations almost always complete before the HTTP connection times out.

A single-record operation is one that creates or affects only one record. For example, add, update, and delete are single-record operations.

List and Search Requests

In general, a list or search request that requires longer than 15 minutes to process is terminated by the system. The exact behavior varies depending on the endpoint.

2014.2 Endpoint

When the 2014.2 endpoint is being used, NetSuite sends a response 10 seconds prior to the connection timeout. The operation is terminated after the record currently being processed has finished. The system uses the messaging described in the following table.

Operation Type	Relevant Operations	Messaging
Search	• search • searchMore • searchMoreWithId • searchNext	The response indicates that the operation did not finish because it timed out.
List	• addList • deleteList • getList • initializeList (for details, see initialize / initializeList) • updateList • upsertList	The response indicates that the operation did not finish because it timed out. For records that were successfully processed, the response indicates success for that record, along with other standard details, such as the record's internal ID. For other records, the status indicates one of the following: • The record's status is unknown – This status is used if the processing of the record was still in progress when the response was being created. • The record was not processed. Using this information and the original SOAP request, you can determine which data to resend.

The following screenshot shows a response for an addList operation that was terminated.

```
<soapenv:Body>
  <addListResponse xmlns=
  "urn:messages_2014_2.platform.webservices.netsuite.com">
    <writeResponseList>
      <platformCore:status isSuccess="false" xmlns:platformCore=
      "urn:core_2014_2.platform.webservices.netsuite.com">
        <platformCore:statusDetail type="WARN">
          <platformCore:code>WARNING</platformCore:code>
          <platformCore:message>The operation exceeded the time
          limit and was aborted.</platformCore:message>
        </platformCore:statusDetail>
      </platformCore:status>
      <writeResponse>
        <platformCore:status isSuccess="true" xmlns:platformCore=
        "urn:core_2014_2.platform.webservices.netsuite.com"/>
        <baseRef internalId="570" externalId="SalesOrderTimeoutV2_2"
        type="salesOrder" xsi:type="platformCore:RecordRef"
        xmlns:platformCore=
        "urn:core_2014_2.platform.webservices.netsuite.com"/>
      </writeResponse>
      <writeResponse>
      <writeResponse>
      <writeResponse>
      <writeResponse>
      <writeResponse>
      <writeResponse>
      <writeResponse>
      <writeResponse>
        <platformCore:status isSuccess="false" xmlns:platformCore=
        "urn:core_2014_2.platform.webservices.netsuite.com">
          <platformCore:statusDetail type="WARN">
            <platformCore:code>WARNING</platformCore:code>
            <platformCore:message>The operation exceeded the time
            limit and the status of the record is unknown.
            </platformCore:message>
          </platformCore:statusDetail>
        </platformCore:status>
      </writeResponse>
      <writeResponse>
        <platformCore:status isSuccess="false" xmlns:platformCore=
        "urn:core_2014_2.platform.webservices.netsuite.com">
          <platformCore:statusDetail type="ERROR">
            <platformCore:code>USER_ERROR</platformCore:code>
            <platformCore:message>The operation exceeded the time
            limit and the record was not processed.
            </platformCore:message>
          </platformCore:statusDetail>
        </platformCore:status>
      </writeResponse>
```

2014.1 Endpoint and Earlier

If you are using 2014.1 or an earlier endpoint, a lengthy operation is stopped after 15 minutes. The system does not send a response, but it updates the Web Services Usage Log with a partial response, which shows only the records that were processed. For help with the Web Services Usage Log, see Using the Web Services Usage Log.

Chapter 7 Web Services Security

NetSuite leverages the latest industry security standards in order to ensure high levels of security around your business data. All Web service requests are controlled by the same security measures used in the NetSuite UI. This includes:

- Authentication
- Authorization
- Session Management
- Encryption

Additional security-related topics include:

- PCI Compliance Password Requirements - for information on password requirements associated with the Payment Card Industry (PCI) Data Security Standard
- Working with Custom Field Security - for information on security applied at the field level and its affect on Web services

Authentication

Authentication is the process of determining the identity of requesters by verifying they are who they claim to be based on the credentials they present.

Important: All Web services operations **require** authentication.

The SuiteTalk Platform supports two separate approaches to authentication:

1. Users can define their credentials in the Passport object and then call the login operation during their request. This approach initiates a session that is governed by timeout and user limits. See Authentication Using the Login Operation for details.
2. Users also have the choice of sending their user credentials in the SOAP header of every request. This approach initiates a "stateless" form of Web services communication in which login is not invoked and timeouts do not apply. See Authentication Using Request Level Credentials for details.

Authentication Using the Login Operation

The SuiteTalk requires a valid user name, password, account number, and role for authentication. These are provided through the login operation using the Passport object.

Example

```
<complexType name="Passport">
```

```
  <sequence>
   <element name="email" type="xsd:string"/>
   <element name="password" type="xsd:string"/>
   <element name="account" type="xsd:string"/>
   <element name="role" type="platformCore:RecordRef" minOccurs="0"/>
  </sequence>
 </complexType>
```

After the requester has been successfully authenticated, a new session is created for that user. When using the *login* operation to authenticate to NetSuite, user credentials are stored in HTTP headers and a JSESSIONID is assigned to every session.

Important: For session information to be successfully transported in SOAP, you must enable support for multiple cookie management in your application. For example, in Microsoft .NET, include the following line:

```
service.CookieContainer = new CookieContainer();
```

Authentication Using Request Level Credentials

Rather than authenticating to NetSuite by invoking login, users have the option of sending their credentials in the SOAP header of each request. Sending credentials with each request eliminates the need for session management and separate logins. This approach, in particular, benefits developers using PHP or other scripting languages that do not have built-in mechanisms for session management, manipulating HTTP headers, or tracking session IDs. Best practices to avoid concurrency errors include the use of request level credentials, along with retries upon concurrency errors.

Note the following about using request level credentials:

- In order to avoid errors, you should not accept cookies when you use request level credentials. If a JSESSIONID is sent, request level credentials cannot be expected to work.
  - To prevent the acceptance of cookies in Java: use setMaintainSession(false).
  - To prevent the acceptance of cookies in .NET, do not create a CookieContainer.
- Users must submit a role with every request.
- When sending user credentials in the SOAP header for an integrated application that has an application ID assigned, users **must** submit the application ID with every request. Users will also be responsible for handling SOAP faults related to exceeded request limits, since they are presumably not synchronizing access to the session.
- As with any request (sent after login is invoked or sent with user credentials in the SOAP header), only one Web services request may be in flight at a time. Any attempt to violate this will result in a SOAP fault.

Important: Users who choose to authenticate to NetSuite using the login operation do not need to provide their credentials in the SOAP header, and can continue to manage their sessions as they have in the past. For information on authentication through login, see Authentication Using the Login Operation.

The following samples show users credentials as they appear in the SOAP header of a request. Also provided is the Java that generated the SOAP.

SOAP Request

```
<soapenv:Header>
    <platformMsgs:passport>
        <platformCore:email>jdoe@netsuite.com</platformCore:email>
        <platformCore:password>mypassword</platformCore:password>
        <platformCore:account>000034</platformCore:account>
        <platformCore:role internalId="3"/>
    </platformMsgs:passport>
</soapenv:Header>
```

Java

```
private void setRequestLevelCredentials(NetSuitePortType aPort, String sEmail, String sPassword
,
 String sAccount, int iRole) throws SOAPException
    {
      NetSuiteBindingStub stub = (NetSuiteBindingStub)aPort;
      stub.clearHeaders();
      org.apache.axis.message.SOAPHeaderElement passportHeader = new
      org.apache.axis.message.SOAPHeaderElement("urn:messages_2008_2.platform.webservices.netsu
ite.com",
 "passport");

        // use the Passport object to create credentials and add to soap header
        RecordRef role = new RecordRef();
        role.setInternalId(String.valueOf(iRole));
        Passport passport = new Passport();
        passport.setEmail(sEmail);
        passport.setPassword(sPassword);
        passport.setAccount(sAccount);
        passport.setRole( role);
        passportHeader.setObjectValue(passport);

        // set header on stub, and disable session management
        stub.setHeader(passportHeader);
        stub.setMaintainSession(false);
    }
```

Important: Users who send request level credentials in a search request must use the searchMoreWithId operation to paginate through search results. See searchMoreWithId for more information.

Authorization

Authorization is the process of ensuring that the requester has the appropriate entitlement to perform the requested operation. When users request to be authenticated, they also provide their NetSuite role. For every Web services request, the system uses the role definition to ensure that the user has the required permission for the requested operation as well as the requested record type. The role must be provided in the Passport type via the *login* operation:

Example

```
<login xmlns="urn:messages_2008_2.platform.webservices.netsuite.com">
<passport>
<ns1:email xmlns:ns1="urn:core_2008_2.platform.webservices.netsuite.com">jblow@webservices.com
</ns1:email>
<ns2:password xmlns:ns2="urn:core_2008_2.platform.webservices.netsuite.com">mypassword<
 ns2:password>
<ns3:account xmlns:ns3="urn:core_2008_2.platform.webservices.netsuite.com">555555</ns3:account>

<ns4:role internalId="3" xmlns:ns4="urn:core_2008_2.platform.webservices.netsuite.com"/>
</passport>
</login>
```

For detailed information on NetSuite roles and permissions and how the SuiteTalk implements roles and permissions rules, refer to Roles and Permissions in Web Services.

Session Management

After a user has been successfully authenticated using the login operation, a sessionID is created that must be passed to each subsequent request. Additional logins are not required as long as the session is active.

In NetSuite, sessions management includes:

- Session Timeouts
- Session Limits
- Manually Managing Cookies
- Reusing Session IDs During Login

Important: The following topics pertain to sessions that have been initiated using the login operation. Users who authenticate to NetSuite by submitting their credentials in the SOAP header of their request should see Authentication Using Request Level Credentials.

Session Timeouts

Web services sessions automatically time out after 20 minutes of inactivity, requiring a login to resume activity. If the server resubmits the cookie during the first 20 minutes, the inactivity timer is reset.

Note: If you need shorter sessions for security reasons, you should create a login route for the client that calls logout when operations are complete.

If you explicitly log out of a session, and then attempt to use the same session, a SESSION_TIMED_OUT error message is returned. Your code should be prepared to handle session timeouts by retrying if an InvalidSessionFault (SESSION_TIMED_OUT) is seen.

Note that timeouts for NetSuite UI and NetSuite Web store sessions are handled differently than Web services session timeouts.

- After 140 minutes of inactivity in the NetSuite application, the screen is locked and a popup displays requiring the user to log in again.
- After 20 minutes of inactivity in a NetSuite Web store, the user is logged out and becomes an anonymous shopper. (There is no automatic relogin to the Web store).

Session Limits

A given login (username/password) is limited to two sessions, one through the browser and one through Web services. These two sessions can occur concurrently. The UI session and the Web services session do not cause termination of each other.

However, if two Web services clients attempt to establish two concurrent sessions, the first session is terminated. Note that more than two independent concurrent sessions are possible with the purchase of products such as Offline Client or Outlook Integration.

Note: Using the same login from multiple client applications, or multiple instances of the same application is NOT supported. However, NetSuite does offer a Web Services Concurrent License for purchase. This license allows designated employees to use the same credentials ten times concurrently before their first session is invalidated. For details see Enabling Web Services Concurrent Users with SuiteCloud Plus.

Manually Managing Cookies

When submitting Web services requests, the NetSuite server can accept any SOAP document as long as it is valid against the NetSuite WSDL — it does not matter how the SOAP was generated. When generating a SOAP document from tools or sources that do NOT automatically generate the NetSuite port objects, you must ensure that all cookies received from the server are managed and passed back to the server in subsequent calls. There are a total of three cookies that are initially returned in the HTTP header of the login response, including the JSESSIONID.

Ensure that your client persists any cookies that our server sends you (re-submits the cookie on the next request). In Axis, this is accomplished by enabling **MaintainSession** on the generated Binding object. In .NET this involves allocating a CookieContainer. If persistence is not maintained, you will receive an InvalidSessionFault — PLEASE_LOGIN_BEFORE_PERFORMING_ACTION.

Example

The following code shows the cookie information returned in the login response.

```
HTTP/1.1 200 OK
Date: Wed, 18 May 2008 18:43:27 GMT
Server: Oracle-Application-Server-10g/9.0.4.0.0 Oracle-HTTP-Server
Set-Cookie: NS_VER=2008.1.0; domain=joe.corp.netsuite.com; path=/
Vary: User-Agent
Set-Cookie: JSESSIONID=ac101fae1f4312dfxxx062fc829447eaa00c3dcd70af41d; Path=/
Set-Cookie: lastUser=TSTDRV198400_935_3; Expires=Wed, 25-May-2008 17:42:24 GMT; Path=/
Cache-Control: private
Keep-Alive: timeout=150, max=1000
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/xml; charset=utf-8
```

For each subsequent request, all of the cookies must be returned in the HTTP header to maintain the session as follows:

C#

```
NetSuiteService nss = new NetSuiteService();
nss.Url = https://webservices.netsuite.com/services/NetSuitePort_2008_1_0 ;
nss.CookieContainer = new CookieContainer();
```

Java with Axis 1.3

```
NetSuiteServiceLocator nss = new NetSuiteServiceLocator();
nss.setMaintainSession(true);
```

Reusing Session IDs During Login

Endpoints starting with v2008_1 support the reuse of session IDs. If you have a valid session at the time you invoke login, NetSuite transparently connects you to your existing session instead of creating a new session. Reusing an existing session (returning the same JSESSIONID) eliminates the overhead of generating a new session and thereby improves performance.

For multi-threaded applications, the primary benefit to session ID reuse is that a thread executing a login does not invalidate sessions of existing threads. Instead, multiple threads share the same reference to a session. The trade-off is that the session returned by login may be in use, and as expected, any attempt to execute an operation with a busy session will result in a WS_CONCUR_SESSION_DISALLWD error.

Note that the strategy for sharing a single set of credentials across threads remains the same as it was for endpoints older than v2008_1. Because only one session can exist for a given set of credentials, and a session cannot be used for more than one operation at a time, multiple threads sharing those credentials should synchronize usage of the session. If collisions are not expected often, a simpler alternative is to implement a *retry* strategy for operations that fail due to a WS_CONCUR_SESSION_DISALLWD error.

Important: Be aware that session ID reuse **does not apply** to the following:

- Sessions managed through a SuiteCloud Plus license. These users require multiple JSESSIONIDs to continue their multi-threaded access through the SuiteTalk APIs.
- Users who have logged in with a different account or role than the initial (existing) session. In this case, as expected, the first session becomes invalid, and a new session is created. Note, however, if a role is not provided during a second login, and a session already exists, users will assume the role of their initial session, regardless of their settings (for example, UIdefault, WSdefault, or last role used).

Encryption

Web services communications are not viewable by a third party as they travel on the Internet. Encryption is implemented using 128-bit encryption with HTTPS/SSL at the transport level. No non-secure Web service requests are granted.

PCI Compliance Password Requirements

When using NetSuite's Credit Card Payments feature, be aware of the Payment Card Industry (PCI) Data Security Standard password requirements. Anyone using the following roles or any custom role with the View Unencrypted Credit Cards permission must change his or her NetSuite password at least every ninety (90) days:

- Administrator
- Accountant
- Bookkeeper

- Controller
- A/R Clerk

If the number of days set in the Password Expiration in Days field on the General Preferences page is less than ninety days, that requirement remains in effect. For example, if your company is set to expire passwords every sixty days, your password expiration date does not change. However, if your company is set to expire passwords every 120 days, this setting automatically changes to 90 days for employees using these roles.

In addition, passwords for those with access to unencrypted credit card numbers must have a minimum of seven (7) characters. If the number of characters set in the Minimum Password Length field on the General Preferences field is greater, that requirement also remains in effect.

All employees using roles with access to unencrypted credit card numbers will be asked to change passwords to meet the PCI compliance requirements.

Working with Custom Field Security

Custom field security can be applied on a per-field basis. If field security has been applied to a field in the UI, the custom field schema will include the field level security metadata. As a result, users should be aware that the permissions they specify on custom fields will apply to any existing integration that is in place using endpoints 2008_1 and older. Certain field permissions such as NONE and VIEW might break the integration in an unintended way.

Note: See the help topic *Restricting Access to Custom Fields* for details on this feature.

If the custom field security feature is turned on, Web services will respect the access level set on each field. For example, if you have set a field permission to NONE, and you have code that references the field, you will not be able to read or write to that field in your code. NetSuite Web services will essentially ignore that field when the SOAP request is sent. Similarly, if a field permission has been set to VIEW, you will be able to read the field's value, but you will be unable to set the field's value.

Web services developers should keep custom field security in mind when designing their integrations. Before beginning a project, they should work with their company's NetSuite administrator to review which fields may have custom security applied.

Chapter 8 Platform Features

The SuiteTalk platform refers to the software infrastructure used to expose NetSuite application functionality via Web services. This section describes some of the core features of the SuiteTalk and includes the following sections:

- Enabling Web Services Concurrent Users with SuiteCloud Plus: provides an overview of the Web Services Plus license type.
- Operations and Their Internal ID Requirements: describes IDs and how they are used.
- Invoking a UI Session for an External Application: describes how to submit an https POST for single login between external web services applications and the NetSuite UI.

Important: Since September 2012, when NetSuite began hosting new customer accounts in multiple data centers, the URL to be used for Web services access depends upon the data center hosting each NetSuite account. For example, it could be webservices.netsuite.com or webservices.na1.netsuite.com. In the 2012.2 and later endpoints, you can use the getDataCenterUrls operation to dynamically discover the correct URL for Web services access to a NetSuite account. A REST service is also available for this purpose. See the help topic *Using the REST roles Service to Get User Accounts, Roles, and Domains*.

Enabling Web Services Concurrent Users with SuiteCloud Plus

Both new and existing users can be designated as "concurrent Web services users" if one or more SuiteCloud Plus license has been purchased. Each SuiteCloud Plus license allows one designated user to create up to ten concurrent Web services sessions. Any user can be assigned a SuiteCloud Plus license, provided that there are enough licenses available in the account. In some cases, you may be able to purchase a second SuiteCloud Plus license to allow two concurrent Web services users (supporting up to 20 sessions), and a third SuiteCloud Plus license to allow three concurrent Web services users (supporting up to 30 sessions).

Note: SuiteCloud Plus provides other upgrades. It increases the number of scheduled script queues. For details, see the help topic *Deploying Scheduled Scripts to Multiple Queues Through SuiteCloud Plus* . It allows the use of multi-threading for CSV import jobs and increases the number of import job queues from one to five. For details, see the help topic *Using Multiple Threads and Multiple Queues to Run CSV Import Jobs*. For a summary of SuiteCloud Plus capabilities, see the help topic *Using SuiteCloud Plus*.

Use of SuiteCloud Plus does not eliminate the need for session management. Depending on the expected throughput, one or more SuiteCloud Plus licenses may be needed to meet the required bandwidth. A user who has been assigned the SuiteCloud Plus license will be granted a new JSESSIONID for up to ten logins, meaning up to ten simultaneous requests originating from the same userid. The eleventh login from a specific user will be rejected. This means that users need to track their sessions and implement pooling and queuing if they believe they will exceed ten active sessions at any given time. Also, as best practices to avoid concurrency errors, do the following: authenticate using request level credentials, and retry if a concurrency error occurs.

Note that the same error message, "Only one request may be made against a session at a time", is used both for Web Services with single licenses and for Web services with SuiteCloud Plus enabled. If you are using Suite Cloud Plus, the system may generate multiple sessions behind the scenes for a user. In this case, if a request comes when an idle session cannot be found, this concurrency error is thrown and the request is not completed.

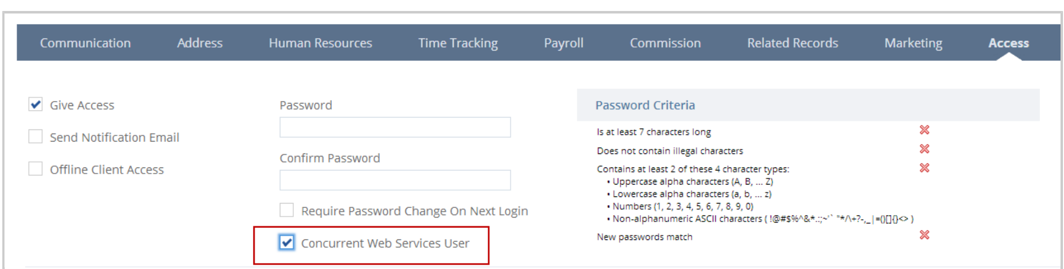
SuiteCloud Plus is not available in NetSuite Small Business. Therefore, if you develop an application that relies on this license type, you cannot implement the application in accounts running NetSuite Small Business.

Your account must have the Web services feature enabled before you can assign a user to be a Web services concurrent user. See Enabling the Web Services Feature for steps on enabling Web services.

Note: Generally, you can purchase one SuiteCloud license per NetSuite account. Contact your sales representative to discuss any need for additional licenses.

To assign the SuiteCloud Plus License to a user:

1. Open the employee record.
2. Click the **Access** subtab.
3. Check the **Concurrent Web Services User** box.
4. Click **Save**.



Benefits of SuiteCloud Plus for Web Services Users

NetSuite recommends SuiteCloud Plus to customers building external applications that expect a maximum of ten concurrent interactions (threads) with NetSuite at any given time.

Customers with ten distinct user seats must manage ten sets of user credentials on the client. With a SuiteCloud Plus license, customers only need to manage one set of user credentials, and NetSuite will grant ten distinct sessions for up to ten processes to authenticate as the same user identity concurrently. Without the SuiteCloud Plus license, customers must build a pooling mechanism into their code and manage the sessions themselves.

Note that customers who have already have a pooling mechanism may not necessarily benefit from a SuiteCloud Plus license. However, new customers who build integrations in which the integrated client is stateless (often true for PHP clients) will benefit from not having to worry about managing sessions.

Additionally, a SuiteCloud Plus lLicense offers reduced administrative overhead since administrators only have to manage one user with ten sessions compared to ten separate users. (For example, when passwords expire, administrators only have to maintain one password on the client side rather than ten.)

Operations and Their Internal ID Requirements

The following table lists the ID requirements for each operation. Note that IDs are only required in calls where specific records corresponding to the call exist in the NetSuite database. For example, on an add operation, no ID is required in the request since the record does not yet exist.

An internal ID is created by the system and returned in the response. On search operations, because a specific record is not being called, an internal ID is not required in the request. However, an internal ID for each record found is returned in the response.

Note: Since externalIDs are provided by the client application, if an externalID is desired for a record, it can be submitted on an Add operation.

Operation	Record ID Required in Request	Record ID Returned in Response
login	n/a	n/a
logout	n/a	n/a
add/addList	No	Yes
update/updateList	Yes	Yes
delete/deleteList	Yes	Yes
getDeleted	No	Yes
get/getList	Yes	Yes
getAll	No	Yes
search/searchNext/searchMore/ searchMoreWithId	No	Yes
getSavedSearch	No	Yes

The following samples show the use of internal IDs as they pertain to operations. Click to see samples for any of the following:

- add operation
- update operation
- deleteList operation
- getList operation
- search operation

add operation

For an add operation, an internal ID is *not* required in the request since the record does not yet exist. The internal ID (in this case 100101) is returned in the response.

Note: There are internal IDs listed in the request, but these internal IDs are *embedded* in the request and do not correspond to the actual record being added (an event) but to other existing records associated with the event record being added.

Request

```
<soap:Body>
<platformMsgs:add>
 <platformMsgs:record xsi:type="actSched:CalendarEvent">
  <actSched:title>Web Services Meeting</actSched:title>
  <actSched:organizer internalId="-5" type="calendarEvent" xsi:type="platformCore:RecordRef">
   <platformCore:name>Mr. Wolfe</platformCore:name>
  </actSched:organizer>
  <actSched:location>Main Conference Room</actSched:location>
  <actSched:attendeeList replaceAll="true" xsi:type="actSched:CalendarEventAttendeeList">
   <actSched:attendee xsi:type="actSched:CalendarEventAttendee">
    <actSched:sendEmail>false</actSched:sendEmail>
    <actSched:attendee internalId="21" type="calendarEvent"
    xsi:type="platformCore:RecordRef"/>
    <actSched:response>_accepted</actSched:response>
    <actSched:attendance>_optional</actSched:attendance>
   </actSched:attendee>
   <actSched:attendee xsi:type="actSched:CalendarEventAttendee">
    <actSched:sendEmail>false</actSched:sendEmail>
    <actSched:attendee internalId="27" type="calendarEvent"
    xsi:type="platformCore:RecordRef"/>
    <actSched:response>_accepted</actSched:response>
    <actSched:attendance>_optional</actSched:attendance>
   </actSched:attendee>
  </actSched:attendeeList>
 </platformMsgs:record>
</platformMsgs:add>
</soap:Body>
```

Response

```
<soapenv:Body>
```

```
 <addResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
  <writeResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
   <ns1:status isSuccess="true" xmlns:ns1="urn:core_2_5.platform.webservices.netsuite.com"/>
   <baseRef internalId="100101" type="calendarEvent" xsi:type="ns2:RecordRef"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ns2="urn:core_2_5.platform.webservices.netsuite.com"/>
  </writeResponse>
 </addResponse>
</soapenv:Body>
```

update operation

In this example the record added above is being updated — since the internal ID matches the one created in the add operation (100101). Here, the internal ID and the record type are required in the request and both are also returned in the response.

Request

```
<soap:Body>
 <platformMsgs:update>
  <platformMsgs:record internalId="100101" xsi:type="actSched:CalendarEvent">
   <actSched:title>Web Services Meeting (Platform)</actSched:title>
  </platformMsgs:record>
 </platformMsgs:update>
</soap:Body>
```

Response

```
<soapenv:Body>
 <updateResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
  <writeResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
   <ns1:status isSuccess="true" xmlns:ns1="urn:core_2_5.platform.webservices.netsuite.com"/>
   <baseRef internalId="100101" type="calendarEvent" xsi:type="ns2:RecordRef"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ns2="urn:core_2_5.platform.webservices.netsuite.com"/>
  </writeResponse>
 </updateResponse>
</soapenv:Body>
```

deleteList operation

In the following delete request, the internal IDs are required for the request and returned in the response. In this case, three Event records are deleted (100101, 100102, and 100103).

Request

```
<soap:Body>
<platformMsgs:deleteList>
 <platformMsgs:baseRef internalId="100101" type="calendarEvent" xsi:type="platformCore:RecordRe
f"/>
 <platformMsgs:baseRef internalId="100102" type="calendarEvent" xsi:type="platformCore:RecordRe
f"/>
 <platformMsgs:baseRef internalId="100103" type="calendarEvent" xsi:type="platformCore:RecordRe
```

```
f"/>
</platformMsgs:deleteList>
</soap:Body>
```

Response

```
<soapenv:Body>
<deleteListResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
<writeResponseList xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
 <writeResponse>
  <ns1:status isSuccess="true" xmlns:ns1="urn:core_2_5.platform.webservices.netsuite.com"/>
  <baseRef internalId="100101" type="calendarEvent" xsi:type="ns2:RecordRef"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ns2="urn:core_2_5.platform.webservices.netsuite.com"/>
 </writeResponse>
 <writeResponse>
  <ns3:status isSuccess="true" xmlns:ns3="urn:core_2_5.platform.webservices.netsuite.com"/>
  <baseRef internalId="100102" type="calendarEvent" xsi:type="ns4:RecordRef"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ns4="urn:core_2_5.platform.webservices.netsuite.com"/>
 </writeResponse>
 <writeResponse>
  <ns5:status isSuccess="true" xmlns:ns5="urn:core_2_5.platform.webservices.netsuite.com"/>
  <baseRef internalId="100103" type="calendarEvent" xsi:type="ns6:RecordRef"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ns6="urn:core_2_5.platform.webservices.netsuite.com"/>
 </writeResponse>
</writeResponseList>
</deleteListResponse>
</soapenv:Body>
```

getList operation

In this example, an internal ID is called for each record to be retrieved — in this case, three different Event records. Again, the internal ID is required for the request and returned in the response.

Request

```
<soap:Body>
<platformMsgs:getList>
 <platformMsgs:baseRef internalId="100104" type="calendarEvent" xsi:type="platformCore:RecordRe
f"/>
 <platformMsgs:baseRef internalId="100105" type="calendarEvent" xsi:type="platformCore:RecordRe
f"/>
 <platformMsgs:baseRef internalId="100106" type="calendarEvent" xsi:type="platformCore:RecordRe
f"/>
</platformMsgs:getList>
</soap:Body>
```

Response

```
<soapenv:Body>
<getListResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
<readResponseList xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
```

```
<readResponse>
 <ns1:status isSuccess="true" xmlns:ns1="urn:core_2_5.platform.webservices.netsuite.com"/>
 <record internalId="100104" xsi:type="ns2:CalendarEvent"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:ns2="urn:scheduling_2_5.activities.webservices.netsuite.com">
  <ns2:title>Customization Meeting</ns2:title>
  <ns2:organizer internalId="-5">
  .........[more fields]
 </record>
</readResponse>
<readResponse>
 <ns8:status isSuccess="true" xmlns:ns8="urn:core_2_5.platform.webservices.netsuite.com"/>
 <record internalId="100105" xsi:type="ns9:CalendarEvent"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:ns9="urn:scheduling_2_5.activities.webservices.netsuite.com">
  <ns9:title>Web Services Meeting (Records)</ns9:title>
  <ns9:organizer internalId="-5">
  .........[more fields]
 </record>
</readResponse>
<readResponse>
 <ns15:status isSuccess="true" xmlns:ns15="urn:core_2_5.platform.webservices.netsuite.com"/>
 <record internalId="100106" xsi:type="ns16:CalendarEvent"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:ns16="urn:scheduling_2_5.activities.webservices.netsuite.com">
  <ns16:title>Web Services Meeting</ns16:title>
  <ns16:organizer internalId="-5">
  .........[more fields]
 </record>
</readResponse>
</readResponseList>
</getListResponse>
</soapenv:Body>
```

search operation

For the search operation, an internal ID is not required for the request but is returned for each record found that matches the specified criteria. In this case, two Event records are returned for a search request calling for each Event record that contains Web Services in the title field.

(Request)

```
<soap:Body>
<platformMsgs:search>
 <platformMsgs:searchRecord xsi:type="actSched:CalendarEventSearch">
  <actSched:title operator="contains" xsi:type="platformCore:SearchStringField">
   <platformCore:searchValue>Web Services</platformCore:searchValue>
  </actSched:title>
 </platformMsgs:searchRecord>
</platformMsgs:search>
</soap:Body>
```

(Response)

```
<soapenv:Body>
```

```
<searchResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
<searchResult xmlns="urn:core_2_5.platform.webservices.netsuite.com">
 <status isSuccess="true"/>
 <totalRecords>2</totalRecords>
 <pageSize>10</pageSize>
 <totalPages>1</totalPages>
 <pageIndex>1</pageIndex>
 <recordList>
  <record internalId="100105" xsi:type="ns1:CalendarEvent"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns1="urn:scheduling_2_5.activities.webservices.netsuite.com">
   <ns1:title>Web Services Meeting (Records)</ns1:title>
   <ns1:organizer internalId="-5">
   .....[more fields]
  </record>
  <record internalId="100106" xsi:type="ns2:CalendarEvent"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns2="urn:scheduling_2_5.activities.webservices.netsuite.com">
   <ns2:title>Web Services Meeting</ns2:title>
   <ns2:organizer internalId="-5">
   .....[more fields]
  </record>
 </recordList>
</searchResult>
</searchResponse>
</soapenv:Body>
```

Invoking a UI Session for an External Application

As an alternative to using SuiteTalk web services, you can allow an external application to access NetSuite by submitting a POST to a specific wslogin page URL. To use this approach, you submit the appropriate credentials and a taskId identifying the desired page. The wslogin page then authenticates the request, provides a new session, and redirects to the requested page.

This feature is **NOT** intended to support any type of user access to NetSuite. It is intended for use solely as part of an integration strategy. For example, you could use this feature in conjunction with an external web services application that moves data into NetSuite. The external application would need to have user credentials but, again, this method is not recommended as a means of providing NetSuite access to actual users.

The following table lists the POST parameters that can be submitted.

Parameter	Type	Required for Authentication	Description
email	text	Yes	Must reflect a valid email address of an entity in your account.
password	password	Yes	The password of the entity associated with the email provided.
taskid	text	Yes	For a list of currently exposed NetSuite TaskIDs, refer to Task IDs. To find the TaskID for a given page while in NetSuite, view the source of the page and search for

Parameter	Type	Required for Authentication	Description
			main_help_anchor. A snippet similar to the following is found: 'main_help_anchor' href="#" onclick="nlPopupHelp(' **LIST_SAVEDSEARCH'**,'Full'); (where LIST_SAVEDSEARCH is the taskID of the page)
id	text	Yes	The ID of a record type available in your NetSuite account.
role	text	Yes, unless the entity (user) has a default role specified on the View All Roles page	Sets the role for the login. This must be a valid role for the entity (user) logging in.
e	text	Optional	If set to T, the record is displayed in edit mode. Any other value is ignored and causes the record to be displayed in view mode.

The post URL is —

```
https://system.netsuite.com/app/webservices/wslogin.nl?c=######
```

— where ###### is your account number.

Because the URL associated with the POST mechanism begins with https, the credentials supplied will be encrypted and, therefore, will NOT be vulnerable during transmission.

Note: Initiating a new browser session using this mechanism will invalidate an existing session that might have been invoked elsewhere using the same credentials.

Chapter 9 Types

This section describes the following:

- Built-in Types — primitive or derived datatypes defined in the XML Schema specification
- Complex Types — NetSuite derived datatypes
- Custom Field Types — datatypes used for custom fields
- Search Types — datatypes used for search records
- Platform Enumerations — datatypes used to populate system defined lists

A WSDL document in Web services uses the XML Schema to define the structure, semantics and constraints of the messages sent between the sender and recipient. The XML Schema is a W3C standard.

There is a strong analogy between object-oriented programming and XML Schema. A type defined in XML Schema can be used to represent an instance of that type in an XML document. Therefore, the body of a SOAP message consists of a combination of such instances of an XML Schema type.

Built-in Types

Built-in (or primitive) types are those that are not defined in terms of other datatypes. They are used as a standardized way to define, send, receive and interpret basic data types in SOAP messages. Primitive data types used in the SuiteTalk can be modified for display purposes. For example, although a price field may be passed in the SOAP messages using an integer primitive data type, the NetSuite UI may format the value with a currency symbol for display purposes.

Of the extensive set of built-in (or primitive) types provided by the XML Schema language specification, the SuiteTalk implementation uses the following built-in types. For detailed information on XML Schema built-in types, refer tohttp://www.w3.org/TR/xmlschema-2/.

- **string** —represents character strings in XML
- **int** — derived from the decimal type and represents the standard concept of the integer numbers
- **double** — is patterned after the IEEE double-precision 64-bit floating point type
- **boolean** — represents the concept of binary-valued logic: {true, false}. In NetSuite, a boolean field can be set to true, false, 1 to indicate true, or 0 to indicate false.
  - In the 2010.2 endpoint and beyond, SuiteTalk validates boolean values. Each value must be a true xsd boolean (0,1,true,false) as per the W3C specification. For details, see the specification.

  - In endpoints prior to 2010.2, invalid values are treated as false.

- **dateTime** — represents integer-valued year, month, day, hour and minute properties, a decimal-valued second property and a boolean timezoned property (timestamp). For example, **2005-09-21T15:24:00.000-07:00**, where 2005-09-21 is the date, 15:24:00.000 is the time and -07:00 is the timezone offset.

Important: When posting dateTime values to NetSuite, it is recommended that you indicate timezone. If no timezone is indicated, the "America/Los_Angeles" timezone will be used (note that for backward compatibility reasons NetSuite does not use GMT).

Note: SOAP encoding is NOT sensitive to your NetSuite Timezone preference as defined in the user preferences. When using Axis, an Axis Web Services client encodes in GMT, regardless of how the machine/JVM is configured. Netsuite will generally encode in PST.

- **date** — represents the top-open intervals of exactly one day in length on the timelines of dateTime, beginning on the beginning moment of each day
- **time** — represents an instant of time that recurs every day

Note: Date, time and dateTime types listed above conform to the ISO 8601 standard for international dates.

Please see http://www.w3.org/TR/2001/RED-xmlschema-2-20010502/datatypes#dateTime for more details.

Complex Types

Complex types are structured types built by aggregating elements of simple or previously defined complex types. NetSuite complex types are defined in the platform core and platform core types XSDs. All NetSuite record types ultimately reference one of these complex types.

NetSuite complex types include the following:

- Passport
- Record
- RecordList
- BaseRef
- RecordRef

- CustomRecordRef
- ListOrRecordRef
- Status
- StatusDetail
- NullField
- ReadResponse
- ListReadResponse
- ListWriteResponse
- WriteResponse

Passport

The Passport type is used by the login operation to encapsulate the security credentials required for authentication. The passport type is defined in the core.xsd.

Field Name	XML Schema Type	Req	Notes
email	xsd:string	Y	
password	xsd:string	Y	
account	xsd:string	Y	Must correspond to a valid NetSuite account number
role	RecordRef	Y	A role is a set of permissions that lets a user access specific areas of data. If not set, NetSuite uses the default Web Services role set for this user.

The default Web services role for a given user is dependent on the values set when setting the permissions for given role. These are set through the UI in Setup > Users/Roles > Manage Roles.

Record

The Record type is an abstract type used as the parameter for the add, addList, delete, deleteList, update and updateList operations. It is also returned in the get, getList, search, searchMore and searchNext operations. All business object types extend the Record type. The record type is defined in core.xsd.

Field Name	XML Schema Type	Req	Notes
id	xsd:string (attribute)	N	

Field Name	XML Schema Type	Req	Notes
xsi:type	xsi:type (attribute)	N	This is a field (attribute) that is automatically implemented by the XML Schema. The value should represent the concrete Record type such as Customer or Event.
nullFieldList	NullFields[]	N	A list of fields that are to be set to null explicitly in the update operation. **Note:** You cannot set the Custom Form field to NULL. Any request to set this field to NULL is ignored.

RecordList

The RecordList type is an array of the Record type. The recordList type is defined in core.xsd.

Field Name	XML Schema Type	Req	Notes
record	Record []	N	

BaseRef

The BaseRef type is an abstract type used to reference any existing record in NetSuite including other business records and custom records. The BaseRef type is defined in core.xsd.

Field Name	XML Schema Type	Req	Notes
name	BaseRef	N	

RecordRef

The RecordRef type is used to reference an existing record in NetSuite in get operations. Typically, a RecordRef references one of the following:

- Another business object such as a customer or sales order
- An entry in a user defined list such as the category list for contacts or sales tax items

The recordRef type descends from BaseRef and is defined in core.xsd.

Field Name	XML Schema Type	Req	Notes
internalId	xsd:string (attribute)	Y	See Using Internal IDs, External IDs, and References.
type	xsd:string (attribute)	N	Reference to a value in a Web services only system list. If the type is known by context, the type is NOT required.
name	xsd:string	N	This is a read-only field that is populated by NetSuite when it's a part of a get or search response. If this field is populated during a write operation, it will be ignored.

CustomRecordRef

The CustomRecordRef type is used to reference any existing custom record in NetSuite. The CustomRecordRef type descends from BaseRef and is defined in core.xsd.

Important: Setting the RecordRef.type to **customRecord** on an add does NOT return a CustomRecord. You must use CustomRecordRef. The CustomRecordRef has a typeId to indicate which kind of CustomRecord it is.

Field Name	XML Schema Type	Req	Notes
internalId	xsd:string (attribute)	Y	References the primary record internal Id. This Id corresponds to the type of custom record. For a list of possible values, see ListOrRecordRef.
typeId	xsd:string	Y	References the custom record type Id.
type	xsd:string (attribute)	N	Reference to a value in a Web services only system list.
name	xsd:string	N	

ListOrRecordRef

The listOrRecordRef type is defined in core.xsd.

Field Name	XML Schema Type	Req	Notes
internalId	xsd:string (attribute)	Y	See Using Internal IDs, External IDs, and References.
externalId	xsd:string (attribute)	N	Use to reference records by their external ID in select and multi-select custom fields.
typeId	xsd:string (attribute)	N	Reference to a value in a Web services only system list.
name	xsd:string	N	This is a read-only field that is populated by NetSuite when it's a part of a get or search response. If this field is populated during a write operation, it is ignored.

Each record type in NetSuite has a corresponding internal ID (or typeId). This internal ID is required when using ListOrRecordRef since the type of record being referenced needs to be specified.

For example, in the following code a new ListOrRecordRef object is created. The list references a specific Entity record as designated by the internalId of **1011** and specifies that the record is of the type **customer** (**-2**). Note that customer records have an internal ID of -2 as shown in the table below.

```
ListOrRecordRef[] fieldNameEntity = new ListOrRecordRef[1];
fieldNameEntity[0] = new ListOrRecordRef();
```

```
fieldNameEntity[0].setInternalId("1011");
fieldNameEntity[0].setTypeId("-2");
```

Status

The Status type contains an array of objects of type StatusDetail. The status type is defined in core.xsd.

Field Name	XML Schema Type	Req	Notes
statusDetail	StatusDetail []	N	Used to capture the specific details for the status. See StatusDetail.
isSuccesss	xsd:Boolean (attribute)	Y	Indicates whether the status is successful or not. If false, one or more statusDetail objects are populated.

StatusDetail

The StatusDetail type is used to capture the specific details for the status. The statusDetail type is defined in core.xsd.

Field Name	XML Schema Type	Req	Notes
code	xsd:string	Y	The status code. See Web Services Error Handling and Error Codes for a listing of codes.
message	xsd:string	Y	The detailed message for this status. See Web Services Error Handling and Error Codes for details.
type	xsd:string		Reference to a value in a Web services only system list. Values: error, warning See Web Services Error Handling and Error Codes for details.

NullField

The NullField type is defined in core.xsd. It contains the following fields.

Field Name	XML Schema Type	Req	Notes
name	xsd:string	Y	Name of the field to be null. The specified name must exactly match an existing field name.

Note: You cannot set the Custom Form field to NULL. Any request to set this field to NULL is ignored.

ReadResponse

The ReadResponse type is used by the following read operations.

- The getResponse output message for the get operation.
- The searchResponse output message for the search operations.

These types have a field named readResponse of the type ReadResponse. The ReadResponse type is defined in message.xsd.

Field Name	XML Schema Type	Req	Notes
status	Status	Y	
recordRef	RecordRef	N	

ListReadResponse

The ListReadResponse type is used by the following operations.

- The GetListResponse output message for the getList operation.

These types have a field named response of type ListReadResponse. The ListReadResponse type is defined in message.xsd.

Field Name	XML Schema Type	Req	Notes
response	ReadResponse[]	Y	An array of ReadResponse types.

ListWriteResponse

The ListWriteResponse type is used by the following operations.

- The AddListResponse output message for the addList operation.
- The UpdateListResponse output message for the updateList operation.
- The DeleteListResponse output message for the deleteList operation.

These types have a field named response of type ListWriteResponse. The ListWriteResponse type is defined in message.xsd.

Field Name	XML Schema Type	Req	Notes
response	WriteResponse[]	Y	An array of WriteResponse types.

WriteResponse

The WriteResponse type is used by the following operations:

- The AddResponse output message for the add operation

- The UpdateResponse output message for the update operation
- The DeleteResponse output message for the delete operation

These types have a field named writeResponse of type WriteResponse. The WriteResponse type is defined in message.xsd.

Field Name	XML Schema Type	Req	Notes
status	Status	Y	
recordRef	RecordRef	N	

Custom Field Types

Custom fields are represented by the type CustomFieldRef, which is an abstract type. The table below contains a list of concrete custom field types that extend the CustomFieldRef type. Each type is followed by its corresponding type in the UI.

XML Schema Type	Custom Field Type in UI
LongCustomFieldRef	Integer
DoubleCustomFieldRef	Decimal Number
BooleanCustomFieldRef	Check Box
StringCustomFieldRef	Free-Form Text Text Area Phone Number E-mail Address Hyperlink Rich Text
DateCustomFieldRef	Date Time of Day or Date/Time (both in one field)
SelectCustomFieldRef	List/Record Document
MultiSelectCustomFieldRef	Multiple Select

CustomFieldRef

The CustomFieldRef type is an abstract type.

Field Name	XML Schema Type	Req	Notes
internalId	xsd:string (attribute)	Y	References a unique instance of a custom field type.

To locate the internal ID for a given custom field in the UI, go to Customization > Lists, Records, & Fields > *[Custom Field]* (where *[Custom Field]* is the type of custom field such as

CRM). The internal IDs for each custom field that has been created is listed in the ID column in the UI.

LongCustomFieldRef

The LongCustomFieldRef type extends the CustomFieldRef abstract type.

Field Name	XML Schema Type	Req	Notes
value	xsd:int	Y	
internalId	xsd:string (attribute)	Y	References a unique instance of a custom field type.

DoubleCustomFieldRef

The DoubleCustomFieldRef type extends the CustomFieldRef abstract type.

Field Name	XML Schema Type	Req	Notes
value	xsd:double	Y	
internalId	xsd:string (attribute)	Y	References a unique instance of a custom field type.

BooleanCustomFieldRef

The BooleanCustomFieldRef type extends the CustomFieldRef abstract type.

Field Name	XML Schema Type	Req	Notes
value	xsd:boolean	Y	
internalId	xsd:string (attribute)	Y	References a unique instance of a custom field type.

StringCustomFieldRef

The StringCustomFieldRef type extends the CustomFieldRef abstract type.

Field Name	XML Schema Type	Req	Notes
value	xsd:string	Y	
internalId	xsd:string (attribute)	Y	References a unique instance of a custom field type.

DateCustomFieldRef

The DateCustomFieldRef type extends the CustomFieldRef abstract type.

Field Name	XML Schema Type	Req	Notes
value	xsd:datetime	Y	
internalId	xsd:string (attribute)	Y	References a unique instance of a custom field type.

SelectCustomFieldRef

The SelectCustomFieldRef type extends the CustomFieldRef abstract type. This references a single ListOrRecordRef and also requires an InternalId attribute to indicate the field name.

Field Name	XML Schema Type	Req	Notes
value	ListorRecordRef	Y	A single ListOrRecordRef.
internalId	xsd:string (attribute)	Y	References a unique instance of a custom field type.

MultiSelectCustomFieldRef

The MultiSelectCustomFieldRef type extends the CustomFieldRef abstract type. This references an array of ListOrRecordRef's and also requires an internalId attribute to indicate the field name.

Field Name	XML Schema Type	Req	Notes
value	ListorRecordRef[]	Y	An array of type RecordRef
internalId	xsd:string (attribute)	Y	References a unique instance of a custom field type.

CustomFieldList

Field Name	XML Schema Type	Req	Notes
value	customFieldRef []	Y	An array of type customFieldRef. The actual entries in the array will be of a concrete type that extends customFieldRef.

The following is an XML excerpt from a SOAP body that illustrates a custom field list that contains all the available custom field types.

SOAP Sample

```
<customFieldList>
 <customField xsi:type="BooleanCustomFieldRef" internalId="CUSTEVENT7">
  <value>true</value>
 </customField>
 <customField xsi:type="DateCustomFieldRef" internalId="CUSTEVENT4">
  <value>2003-01-20T18:47:00</value>
 </customField>
 <customField xsi:type="DoubleCustomFieldRef" internalId="CUSTEVENT2">
  <value>23.465</value>
 </customField>
 <customField xsi:type="LongCustomFieldRef" internalId="CUSTEVENT5">
  <value>23</value>
 </customField>
 <customField xsi:type="ListCustomFieldRef" internalId="CUSTEVENT3">
  <value internalId="offsite"/>
 </customField>
```

```
 <customField xsi:type="StringCustomFieldRef" internalId="CUSTEVENT12">
  <value>John Williams</value>
 </customField>
 <customField xsi:type="MultiSelectCustomFieldRef" internalId="CUSTEVENT11">
  <value internalId="important"/>
  <value internalId="strategic"/>
 </customField>
<customFieldList>
```

Search Types

Every NetSuite record that supports search has corresponding search and advanced search objects. For example, the SuiteTalk listRel XSD contains a Customer object, as well as its corresponding CustomerSearch, CustomerSearchAdvanced, and CustomerSearchRow search objects.

- When using a < *Record* > **Search** object, all search fields within this search object belong to one of the types decribed in Search XML Schema Types.
- When searching on custom records or custom fields, all search fields belong to one of the types described in Search Custom Field XML Schema Types.
- When using either the < *Record* > **SearchAdvanced** or < *Record* > **SearchRow** search objects, all search fields within these "advanced search" objects belong to one of the types described in Search Column Custom XML Schema Types or Search Column Custom XML Schema Types.

Search XML Schema Types

The following sections define available search types. Every search field within a search object type belongs to one of these search types.

SearchPreferences

Field Name	XML Schema Type	Req	Notes
bodyFieldsOnly	boolean	Y/N	See Setting Search Preferences.
pageSize	int	Y/N	See Setting Search Preferences.
returnSearchColumns	boolean	Y/N	See Setting Search Preferences.

SearchRequest

Field Name	XML Schema Type	Req	Notes
preferences	boolean	Y	
searchRecord	SearchRecord	Y	

SearchResult

Field Name	XML Schema Type	Req	Notes
totalRecords	int	N	
pageSize	int	N	
totalPages	int	N	
pageIndex	int	N	
searchId	string	N	
status			
recordList	platformCore:RecordList		
searchRowList	platformCore:SearchRowList		

SearchStringField

Field Name	XML Schema Type	Req	Notes
operator	platformCoreTyp: SearchStringFieldOperator (attribute)	Y	Reference to a value in a system list. For more information on available values, see Platform Enumerations.
searchValue	xsd:string	Y	

SearchBooleanField

Field Name	XML Schema Type	Req	Notes
operator	xsd:boolean	Y	The available values are true or false.

SearchDoubleField

Field Name	XML Schema Type	Req	Notes
operator	platformCoreTyp: SearchDoubleFieldOperator(attribute)	Y	Reference to a value in a system list. For more information on available values, see Platform Enumerations.
searchValue	xsd:double	Y	
searchValue2	xsd:double	N	If the operator is between or notBetween searchValue2 must be populated.

SearchLongField

Field Name	XML Schema Type	Req	Notes
operator	platformCoreTyp: SearchLongFieldOperator(attribute)	Y	Reference to a value in a system list. For more information on available values, see Platform Enumerations.

Field Name	XML Schema Type	Req	Notes
searchValue	xsd:long	Y	
searchValue2	xsd:long	N	If the operator is between or notBetween searchValue2 must be populated.

SearchTextNumberField

Field Name	XML Schema Type	Req	Notes
operator	platformCoreTyp: SearchTextNumberFieldOperator (attribute)	Y	Reference to a value in a system list. For more information on available values, see Platform Enumerations.
searchValue	xsd:string	Y	
searchValue2	xsd:string	N	If the operator is between or notBetween searchValue2 must be populated.

SearchDateField

Field Name	XML Schema Type	Req	Notes
operator	platformCoreTyp: SearchDate (attribute)	Y	Reference to a value in a system list. For more information on available values, see Platform Enumerations.
predefinedSearchValue	plateformCoreTyp: SearchDate	N	Reference to a value in a system list. For more information on available values, see Platform Enumerations.
searchValue	xsd:dateTime	N	Either predefinedSearchValue or searchValue should be populated.
searchValue2	xsd: dateTime	N	If the operator is between or notBetween searchValue2 must be populated.

SearchMultiSelectField

This search type is used to specify a list of one or more internal IDs that reference other user defined records in the system.

Note: Note that the maximum number of values that can be specified in a MultiSelectField is 1000.

Field Name	XML Schema Type	Req	Notes
operator	platformCoreTyp: SearchMultiSelectFieldOperator (attribute)	Y	Reference to a value in a system list. For more information on available values, see Platform Enumerations.
searchValue	platformCore: RecordRef	Y	An array of type SearchMultiSelectRefValue.

SearchEnumMultiSelectField

This search type is used to specify a list of one or more system defined constants.

Field Name	XML Schema Type	Req	Notes
operator	platformCoreTyp: SearchEnumMultiSelect FieldOperator (attribute)	Y	Reference to a value in a system list. For more information on available values, see Platform Enumerations.
searchValue	xsd:string	Y	

Search Custom Field XML Schema Types

The following sections define the available search types for custom fields. Every search field within a search object type belongs to one of these search types.

SearchCustomField

This is an abstract type.

Field Name	XML Schema Type	Req	Notes
id	xsd:string (attribute)	Y	References a unique instance of a custom field

SearchStringCustomField

The SearchStringCustomField type extends the SearchCustomField abstract type.

Field Name	XML Schema Type	Req	Notes
internalId	xsd:string (attribute)	Y	Inherited from the SearchCustomField. Reference a unique instance of a custom field.
operator	platformCoreTyp: SearchStringFieldOperator (attribute)	Y	The type is an enumeration type that restricts the value to a predefined list. For more information on available values, see Platform Enumerations.
searchValue	xsd:string	Y	

SearchBooleanCustomField

The SearchBooleanCustomField type extends the SearchCustomField abstract type.

Field Name	XML Schema Type	Req	Notes
internalId	xsd:string (attribute)	Y	Inherited from the SearchCustomField. Reference a unique instance of a custom field.

Field Name	XML Schema Type	Req	Notes
searchValue	xsd: boolean	Y	The type is an enumeration type that restricts the value to true or false.

SearchLongCustomField

The SearchLongCustomField type extends the SearchCustomField abstract type.

Field Name	XML Schema Type	Req	Notes
internalId	xsd:string (attribute)	Y	Inherited from the SearchCustomField. Reference a unique instance of a custom field.
operator	platformCoreTyp: SearchLongFieldOperator (attribute)	Y	The type is an enumeration type that restricts the value to a predefined list. For more information on available values, see Platform Enumerations.
searchValue	xsd:long	Y	
searchValue2	xsd:long	N	

SearchDoubleCustomField

The SearchDoubleCustomField type extends the SearchCustomField abstract type.

Field Name	XML Schema Type	Req	Notes
internalId	xsd:string (attribute)	Y	Inherited from the SearchCustomField. Reference a unique instance of a custom field.
operator	platformCoreTyp: SearchDoubleFieldOperator (attribute)	Y	The type is an enumeration type that restricts the value to a predefined list. For more information on available values, see Platform Enumerations.
searchValue	xsd:double	Y	
searchValue2	xsd:double	N	

SearchDateCustomField

The SearchDateCustomField type extends the SearchCustomField abstract type.

Field Name	XML Schema Type	Req	Notes
internalId	xsd:string (attribute)	Y	Inherited from the SearchCustomField. Reference a unique instance of a custom field.
operator	platformCoreTyp: SearchDateFieldOperator (attribute)	Y	The type is an enumeration type that restricts the value to a predefined list. For more information on available values, see Platform Enumerations.

Field Name	XML Schema Type	Req	Notes
predefinedSearchValue	platformCoreTyp: SearchDate		
searchValue	xsd:dateTime	Y	
searchValue2	xsd:dateTime	Y	

SearchMultiSelectCustomField

The SearchMultiSelectCustomField type extends the SearchCustomField abstract type.

Field Name	XML Schema Type	Req	Notes
internalId	xsd:string (attribute)	Y	Inherited from the SearchCustomField. Reference a unique instance of a custom field.
operator	platformCoreTyp: SearchMultiSelectFieldOperator (attribute)	Y	The type is an enumeration type that restricts the value to a predefined list. For more information on available values, see Platform Enumerations.
searchValue	ListOrRecordRef	Y	

SearchEnumMultiSelectCustomField

The SearchEnumMultiSelectCustomField type extends the SearchCustomField abstract type.

Field Name	XML Schema Type	Req	Notes
internalId	xsd:string (attribute)	Y	Inherited from the SearchCustomField. Reference a unique instance of a custom field.
operator	platformCoreTyp: SearchEnumMultiSelectFieldOperator (attribute)	Y	The type is an enumeration type that restricts the value to a predefined list. For more information on available values, see Platform Enumerations.
searchValue	xsd:string	Y	

SearchCustomFieldList

Field Name	XML Schema Type	Req	Notes
customField	platformCore: SearchCustomField []	Y	

Search Column XML Schema Types

The following search types support advanced search functionality in Web services.

SearchColumnField

This is an abstract type.

Field Name	XML Schema Type	Req	Notes
customLabel	xsd:string		

SearchColumnBooleanField

The SearchColumnBooleanField type extends the SearchColumnField abstract type.

Field Name	XML Schema Type	Req	Notes
searchValue	xsd:boolean	Y	

SearchColumnStringField

The SearchColumnStringField type extends the SearchColumnField abstract type.

Field Name	XML Schema Type	Req	Notes
searchValue	xsd:string	Y	

SearchColumnLongField

The SearchColumnLongField type extends the SearchColumnField abstract type.

Field Name	XML Schema Type	Req	Notes
searchValue	xsd:long	Y	

SearchColumnTextNumberField

The SearchColumnStringField type extends the SearchColumnField abstract type.

Field Name	XML Schema Type	Req	Notes
searchValue	xsd:string	Y	

SearchColumnDoubleField

The SearchColumnDoubleField type extends the SearchColumnField abstract type.

Field Name	XML Schema Type	Req	Notes
searchValue	xsd:double	Y	

SearchColumnDateField

The SearchColumnDateField type extends the SearchColumnField abstract type.

Field Name	XML Schema Type	Req	Notes
searchValue	xsd:dateTime	Y	

SearchColumnEnumSelectField

The SearchColumnEnumSelectField type extends the SearchColumnField abstract type.

Field Name	XML Schema Type	Req	Notes
searchValue	xsd:string	Y	

SearchColumnSelectField

The SearchColumnSelectField type extends the SearchColumnField abstract type. This references a single ListOrRecordRef and also requires an internalId attribute to indicate the field name.

Field Name	XML Schema Type	Req	Notes
searchValue	RecordRef	Y	

Search Column Custom XML Schema Types

The following search types support advanced search functionality in Web services.

SearchColumnCustomField

This is an abstract type.

Field Name	XML Schema Type	Req	Notes
customLabel	xsd:string		

SearchColumnBooleanCustomField

The SearchColumnBooleanField type extends the SearchColumnCustomField abstract type.

Field Name	XML Schema Type	Req	Notes
searchValue	xsd:boolean	Y	
internalId	xsd:string (attribute)	Y	

SearchColumnStringCustomField

The SearchColumnStringCustomField type extends the SearchColumnCustomField abstract type.

Field Name	XML Schema Type	Req	Notes
searchValue	xsd:string	Y	

Field Name	XML Schema Type	Req	Notes
internalId	xsd:string (attribute)	Y	

SearchColumnLongCustomField

The SearchColumnLongCustomField type extends the SearchColumnCustomField abstract type.

Field Name	XML Schema Type	Req	Notes
searchValue	xsd:long	Y	
internalId	xsd:string (attribute)	Y	

SearchColumnDoubleCustomField

The SearchColumnDoubleCustomField type extends the SearchColumnCustomField abstract type.

Field Name	XML Schema Type	Req	Notes
searchValue	xsd:double	Y	
internalId	xsd:string (attribute)	Y	

SearchColumnDateCustomField

The SearchColumnDateCustomField type extends the SearchColumnCustomField abstract type.

Field Name	XML Schema Type	Req	Notes
searchValue	xsd:dateTime	Y	
internalId	xsd:string (attribute)	Y	

SearchColumnEnumMultiSelectCustomField

The SearchColumnEnumMultiSelectCustomField type extends the SearchColumnCustomField abstract type.

Field Name	XML Schema Type	Req	Notes
searchValue	xsd:string	Y	
internalId	xsd:string (attribute)	Y	

SearchColumnSelectCustomField

The SearchColumnSelectCustomField type extends the SearchColumnCustomField abstract type. This references a single ListOrRecordRef and also requires an internalId attribute to indicate the field name.

Field Name	XML Schema Type	Req	Notes
searchValue	ListOrRecordRef	Y	A single ListOrRecordRef
internalId	xsd:string (attribute)	Y	

SearchColumnMultiSelectCustomField

The SearchColumnMultiSelectCustomField type extends the SearchColumnCustomField abstract type. This references an array of ListOrRecordRefs.

Field Name	XML Schema Type	Req	Notes
searchValue	ListOrRecordRef[]	Y	
internalId	xsd:string (attribute)	Y	

SearchColumnCustomFieldList

The SearchColumnCustomFieldList type extends the SearchColumnCustomField abstract type.

Field Name	XML Schema Type	Req	Notes
customField	SearchColumnCustomField		

Sample Code

SOAP Request — Opportunity Search

Following is an example that contains an excerpt of the SOAP body that illustrates an opportunity search containing several search field types.

```
<opportunitySearch>
  <projectedTotal operator="lessThan">
   <searchValue>100000</searchValue>
  </projectedTotal>
  <title operator="contains">
   <searchValue>Enterprise</searchValue>
  </title>
  <createdDateRange operator="between">
   <fromValue>2003-10-02</fromValue>
   <toValue>2003-10-12</toValue>
  </createdDateRange>
  <opportunityStatusList operator="anyOf">
   <searchValue>inProgress</searchValue>
   <searchValue>closedWon</searchValue>
  </opportunityStatusList>
  <customFieldList>
   <customField xsi:type="SearchSelectCustomField" internalId="SalesEngineer"
   operator="equals">
    <searchValue>Buddy Williams</searchValue>
```

```
    </customField>
    <customField xsi:type="SearchBooleanCustomField" internalId="hasSalesEngineer"
    operator="true"/>
    <customField xsi:type="SearchStringCustomField" internalId="DemoNotes"
    operator="startsWith">
     <searchValue>CRM</searchValue>
    </customField>
    <customField xsi:type="SearchMultiSelectCustomField" internalId="ProductAreas"
    operator="noneOf">
     <searchValue>Inventory</searchValue>
     <searchValue>Warehousing</searchValue>
    </customField><
   </customFieldList>
</opportunitySearch>
```

Platform Enumerations

The following search types are used throughout Web services to populate system defined lists. The tables below outline the available values that should be used to populate these fields in your Web services requests. These enumerations are defined in the platformCoreTyp XSD.

Search Types (Table 1)					
Enumerations	**SearchString FieldOperator**	**SearchLongFieldOperator**	**SearchDoubleFieldOperator**	**SearchPeriod Field**	**SearchTime FieldOperator**
is	X				
isNot	X				
startsWith	X				
doesNotStartWith	X				
contains	X				
doesNotContain	X				
equalTo		X	X	X	
lessThan		X	X		
greaterThan		X	X		
lessThanOrEqualTo		X	X		
greaterThanOrEqualTo		X	X		
notEqualTo		X	X		
notLessThan		X	X		
notGreaterThan		X	X		
notLessThanOrEqualTo		X	X		
notGreaterThanOrEqualTo		X	X		
between		X	X		X
notBetween		X	X		
empty		X	X		

Search Types (Table 1)					
Enumerations	**SearchString FieldOperator**	**SearchLongFieldOperator**	**SearchDoubleFieldOperator**	**SearchPeriod Field**	**SearchTime FieldOperator**
notEmpty		X	X		

Search Types (Table 2)				
Enumerations	**Se Fi t o r**	**Se M t Fi t o r**	**Se i Se d O**	**SearchDate**
anyOf		X	X	
noneOf		X	X	
on	X			
before	X			
after	X			
onOrBefore	X			
onOrAfter	X			
within	X			
empty	X			
notOn	X			
notBefore	X			
notAfter	X			
notOnOrBefore	X			
notOnOrAfter	X			
notWithin	X			
notEmpty	X			
lastBusinessWeek				X
lastFiscalQuarter				X
lastFiscalQuarterToDate				X
lastFiscalYear				X
lastFiscalYearToDate				X
lastMonth				X
lastMonthToDate				X
lastRollingQuarter				X
lastRollingYear				X
lastWeek				X
lastWeekToDate				X

Search Types (Table 2)				
Enumerations	**Se Fi t o r**	**Se M t Fi t o r**	**Se i Se d O**	**SearchDate**
nextBusinessWeek				X
nextFiscalQuarter				X
nextFiscalYear				X
nextFourWeeks				X
nextMonth				X
nextOneMonth				X
nextOneQuarter				X
nextOneWeek				X
nextOneYear				X
nextWeek				X
previousOneDay				X
previousOneMonth				X
previousOneQuarter				X
previousOneWeek				X
previousOneYear				X
previousRollingQuarter				X
previousRollingYear				X
sameMonthLastFiscalQuarter				X
sameMonthLastFiscalQuarterToDate				X
sameMonthLastFiscalYear				X
sameMonthLastFiscalYearToDate				X
sameQuarterLastFiscalYear				X
sameQuarterLastFiscalYearToDate				X
thisBusinessWeek				X
thisFiscalQuarter				X
thisFiscalQuarterToDate				X
thisFiscalYear				X
thisFiscalYearToDate				X
thisMonth				X
thisMonthToDate				X

Search Types (Table 2)				
Enumerations	**Se Fi t o r**	**Se M t Fi t o r**	**Se i Se d O**	**SearchDate**
thisRollingQuarter				X
thisRollingYear				X
thisWeek				X
thisWeekToDate				X
thisYear				X
today				X
tomorrow				X
yesterday				X

Chapter 10 Web Services Operations

SuiteTalk exposes NetSuite as a data source for programmatic access, hence most of the data operations developers expect, such as insert, update, delete, and select/search are supported. There are also a number of operations that are available as supporting operations for data read/write operations (for example: initialize and getSelectValue), or for providing metadata (for example: getCustomization), or for exposing an application function to programmatic access (for example: attach, detach).

See these sections for complete details on all SuiteTalk operations:

- Web Services Standard Operations
- Web Services List Operations
- Web Services Asynchronous Operations

Important: Since September 2012, when NetSuite began hosting new customer accounts in multiple data centers, the URL to be used for Web services access depends upon the data center hosting each NetSuite account. For example, it could be webservices.netsuite.com or webservices.na1.netsuite.com. In the 2012.2 and later endpoints, you can use the getDataCenterUrls operation to dynamically discover the correct URL for Web services access to a NetSuite account. A REST service is also available for this purpose. For general information about this service, see the help topic *Using the REST roles Service to Get User Accounts, Roles, and Domains*. For code samples, see Java REST Example for Web Services Domain Discovery and .NET REST Example for Web Services Domain Discovery.

Web Services Standard Operations

The following operations are supported in SuiteTalk. They are organized in alphabetical order:

Operation / API	Summary
add / addList	Use to add one or more records into the system. The system returns a NetSuite identifier (internalId) that is unique for each record created within a record type.
attach / detach	Use to attach or detach another record or file to/from another record.
changeEmail	Use to change the email address for the account.
changePassword	Use to change the password for the account.
changePasswordOrEmail	Use to change a users email or password.
checkAsyncStatus	Use to check the status of an asynchronous Web services submission.
delete / deleteList	Use to delete one or more records in the system. The records to be deleted are identified by either the internal or external ID and the record type.

Operation / API	Summary
get / getList	Use to query the system for one or more records. You must provide either the internal or external ID and the record type for each query item.
getAll	Use to return a list of records that do not have a search interface.
getAsyncResult	Use to retrieve the results of an asynchronous Web services submission.
getBudgetExchangeRate	Use to get and filter all data related to the Budget Exchange Rates table.
getConsolidatedExchangeRate	Use to get and filter all data related to the Consolidated Exchange Rates table.
getCurrencyRate	Use to get currency exchange rates from the Currency Exchange Rate table.
getCustomization	This operation has been deprecated.
getCustomizationId	Use to retrieve the internalIds, externalIds, and/or scriptIds of all custom objects of a specified type.
getDataCenterUrls	Use for dynamic discovery of data-center-specific URLs for partner applications' access to NetSuite.
getDeleted	Use to retrieve a list of deleted records of a given type during a specified period.
getItemAvailability	Use to retrieve the inventory availability for a given list of items.
getPostingTransactionSummary	Use to retrieve a summary of the actual data in an account.
getSavedSearch	Use to retrieve a list of existing saved searches for each record type. Specify the search record type to get a list of record references of the saved search.
getSelectValue	Use to retrieve valid values for a given recordRef field where the referenced record type is not yet exposed in the Web services API or when the logged in role does not have permission to the instances of the record type.
getServerTime	Use to get server time, resulting in more accurate and reliable sync'ing of data than using using local client time. The client will use the time from server to determine if the record has changed since the last synchronization.
initialize / initializeList	Use to emulate the UI workflow by pre-populating fields on transaction line items with values from a related record. The initializeList operation can be used to run batch processes to retrieve initialized records.
login	Use to login into NetSuite. This operation is similar to the NetSuite UI and requires you to provide a valid username, password, role, and account number.
logout	Use to logout from the system. The logout operation invalidates the current session.
mapSso	Use to automate the mapping between external applications credentials and NetSuite's credentials for a user.
search	Use to search for a set of records based on specific search criteria. This operation supports pagination, so that large result sets can be retrieved in smaller sets. For more information on how to control pagination, refer to Setting Web Services Preferences.
searchMore	Used to retrieve more records after an initial search operation is invoked.

Operation / API	Summary
searchMoreWithId	Use to retrieve search results for users who send requests to NetSuite by providing request-level credentials, rather than by invoking login. In this case, users cannot call searchMore or searchNext.
searchNext	Use to retrieve the next set of records after an initial search operation is invoked.
ssoLogin	Use to establish a single sign-on connection. This operation allows for a partner application to login on behalf of the user in to NetSuite, without the user's credentials ever going through partner servers.
update / updateList	Use to update one or more existing records in the system by providing new values for the fields to be updated for each record. The records to be updated are identified by either the internal or external ID and the record type.
updateInviteeStatus / updateInviteeStatusList	Allows event invitees to accept or decline NetSuite events. After invitees have responded to the event, the Event record is updated with their response.
upsert / upsertList	Use to add new records and update existing records in a single operation. Records are identified by external ID and record type. If a record of the specified type with a matching external ID exists in the system, it is updated. If it does not exist, a new record is created.

Web Services List Operations

Within a single soap request, only one operation can be performed. For example, one add, one addList or one delete. However, a single **list** operation (addList, updateList, upsertList, deleteList, getList, and initializeList) allows you to work with multiple record types. For example, with a single addList operation you can add 3 customers, 4 opportunities, and 1 contact.

Note that list operations process records in the order in which the records are submitted. For example, an addList operation will add all records in the order they appear in the list.

Web Services Asynchronous Operations

The following asynchronous equivalents are available for these list operations:

- addList: asyncAddList
- updateList: asyncUpdateList
- upsertList: asyncUpsertList
- deleteList: asyncDeleteList
- getList: asyncGetList
- search: asyncSearch

- initializeList: asyncInitializeList

You can use the *checkAsyncStatus* and *getAsyncResult* operations to track asynchronous requests. For information on asynchronous request processing, see Synchronous Versus Asynchronous Request Processing.

add

The add operation is used to add a new instance of a record in NetSuite. It is similar to the addList operation except that it allows only one record to be added at a time.

Note: Although records of a particular type may be used in multiple integration scenarios, each record instance can only have a single external ID value. In order to maintain data integrity, only a single integrated application can set and update external ID values for each record type. External ID values for all records of each type must all come from the same external application.

Request

The AddRequest type is used for the request. It contains the following fields.

Element Name	XSD Type	Notes
record	Record	The record type is an abstract type so an instance of a type that extends record must be used — such as Customer or Event.

Response

The AddResponse type is used for the response. It contains the following fields.

Element Name	XSD Type	Notes
response	WriteResponse	Contains details on the status of the operation and a reference to the created record.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault

- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

SOAP Request

In this example, a single customer record is added.

```
<soap:Body>
<platformMsgs:add>
        <platformMsgs:record xsi:type="listRel:Customer">
                <listRel:entityId>Shutter Fly</listRel:entityId>
                <listRel:companyName>Shutter Fly, Inc</listRel:companyName>
                <listRel:unsubscribe>false</listRel:unsubscribe>
        </platformMsgs:record>
</platformMsgs:add>
</soap:Body>
```

SOAP Response

In the response, notice that the internalID for the record added is returned.

```
<soapenv:Body>
<addResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
<writeResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
 <ns1:status isSuccess="true" xmlns:ns1="urn:core_2_5.platform.webservices.netsuite.com"/>
  <baseRef internalId="979" type="customer" xsi:type="ns2:RecordRef"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns2="urn:core_2_5.platform.webservices.netsuite.com"/>
</writeResponse>
</addResponse>
</soapenv:Body>
```

C#

```
private void addCustomer()
{
 // This operation requires a valid session
 this.login( true );

 Customer customer = new Customer();

 // Set entityId, company name, and email
 if ( "true".Equals( _dataCollection["promptForFieldValues"] ) )
 {
  _out.writeLn(
  "\nPlease enter the following customer information. " +
  "Note that some fields have already been populated. " );
  _out.write( "Entity name: " );
  customer.entityId = _out.readLn();
  _out.write( "Company name: " );
  customer.companyName = _out.readLn();
```

```
 _out.write( "E-mail: " );
 customer.email = _out.readLn();
 _out.write( "Sales Rep key: " );
 RecordRef salesRep = new RecordRef();
 salesRep.internalId = _out.readLn();
 //salesRep.type = RecordType.contact;
 //salesRep.typeSpecified = true;
 customer.salesRep = salesRep;
}
else
{
 customer.entityId = "XYZ Inc";
 customer.companyName = "XYZ, Inc.";
 customer.email = "bsanders@yahoo.com";
}

// Set email preference
customer.emailPreference = CustomerEmailPreference._hTML;
customer.emailPreferenceSpecified = true;

// Set entity status. The nsKey can be obtained from Setup > SFA > Customer Statuses.
// The default status is "Closed Won" which has an nsKey of 13
RecordRef status = new RecordRef();
if ( "true".Equals( _dataCollection["promptForFieldValues"] ) )
{
 _out.write( "Entity status nsKey (press enter for default value of Closed Won): " );
 String statusKey = _out.readLn();
 if ( statusKey.Equals( "" ) )
 {
  status.internalId = "13";
 }
 else
 {
  status.internalId = statusKey;
 }
}
else
{
 status.internalId = "13";
}

customer.entityStatus = status;

// Populate the address list for this customer. You can put in as many
// adresses as you like.
CustomerAddressbook address = new CustomerAddressbook();
address.defaultShipping = true;
address.defaultShippingSpecified = true;
address.defaultBilling = false;
address.defaultBillingSpecified = true;
address.label = "Shipping Address";
address.addressee = "William Sanders";
address.attention = "William Sanders";
address.addr1 = "4765 Sunset Blvd";
address.city = "San Francisco";
address.state = "CA";
address.zip = "94131";
address.country = Country._unitedStates;

// Attach the CustomerAddressbookList to the customer
```

```
CustomerAddressbookList addressList = new CustomerAddressbookList();
CustomerAddressbook[] addresses = new CustomerAddressbook[1];
addresses[0] = address;
addressList.addressbook = addresses;
customer.addressbookList = addressList;

// Invoke add() operation
WriteResponse response = _service.add( customer );

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

// Process the response
if ( response.status.isSuccess )
{
 _out.info(
 "\nThe following customer was added successfully:" +
 "\nkey=" + ((RecordRef) response.baseRef).internalId +
 "\nentityId=" + customer.entityId +
 "\ncompanyName=" + customer.companyName +
 "\nemail=" + customer.email +
 "\nstatusKey=" + customer.entityStatus.internalId +
 "\naddressbookList[0].label=" + customer.addressbookList.addressbook[0].label );

 // Create a second customer that's a sub-customer of the first customer
 /*
 Customer subCustomer = new Customer();

 subCustomer.entityId = "XYZ Japan";

 // Set the parent customer
 recordRef = new RecordRef();
 recordRef.internalId = ((RecordRef) response.baseRef).internalId;
 //recordRef.internalId = "125";
 subCustomer.parent = recordRef;

 // Invoke add() operation
 response = _service.add( subCustomer );

 // Process the response
 //"\nThe add() operation with job id " + _service.sessionInfo.jobID + " was processed " );
 if ( !response.status.isSuccess )
 {
  _out.info( "\nThe customer was not added:" );
  _out.error( getStatusDetails( response.status ) );
 }
 else
 {
  _out.info(
  "\nThe following customer was added successfully:" +
  "\nkey=" + ((RecordRef) response.baseRef).internalId +
  "\nentityId=" + customer.entityId +
  "\ncompanyName=" + customer.companyName);
 }
 */
}
else
{
```

```
  _out.error( "The customer was not added:", true );
  _out.error( getStatusDetails( response.status ) );
 }
}
```

Java

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

 Customer customer = new Customer();

 // Set entityId, company name, and email
 if ("true".equals(_properties.getProperty("promptForFieldValues"))) {
  _console
  .writeLn("\nPlease enter the following customer information. "
  + "Note that some fields have already been populated. ");
  _console.write("Entity name: ");
  customer.setEntityId(_console.readLn());
  _console.write("Company name: ");
  customer.setCompanyName(_console.readLn());
  _console.write("E-mail: ");
  customer.setEmail(_console.readLn());
 } else {
  customer.setEntityId("XYZ Inc");
  customer.setCompanyName("XYZ, Inc.");
  customer.setEmail("bsanders@yahoo.com");
 }

 // Set email preference.
 customer.setEmailPreference(CustomerEmailPreference._hTML);

 // Set entity status. The nsKey can be obtained from Setup > SFA >
 // Customer Statuses.
 // The default status is "Closed Won" which has an nsKey of 13
 RecordRef status = new RecordRef();
 if ("true".equals(_properties.getProperty("promptForFieldValues"))) {
  _console.write("Entity status nsKey (press enter for default value of Closed Won): ");
  String statusKey = _console.readLn();
  if (statusKey.equals("")) {
   status.setInternalId("13");
  } else {
   status.setInternalId(statusKey);
  }
 } else {
  status.setInternalId("13");
 }

 customer.setEntityStatus(status);

 // Populate the address list for this customer. You can put in as many
 // adresses as you like.
 CustomerAddressbook address = new CustomerAddressbook();
 address.setDefaultShipping(Boolean.TRUE);
 address.setDefaultBilling(Boolean.FALSE);
 address.setLabel("Shipping Address");
 address.setAddressee("William Sanders");
```

```
 address.setAttention("William Sanders");
 address.setAddr1("4765 Sunset Blvd");
 address.setCity("San Francisco");
 address.setState("CA");
 address.setZip("94131");
 address.setCountry(Country._unitedStates);

 // Attach the CustomerAddressbookList to the customer
 CustomerAddressbookList addressList = new CustomerAddressbookList();
 CustomerAddressbook[] addresses = new CustomerAddressbook[1];
 addresses[0] = address;
 addressList.setAddressbook(addresses);
 customer.setAddressbookList(addressList);

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

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

 // Process the response
 if (response.getStatus().isIsSuccess()) {
  _console.info("\nThe following customer was added successfully:"
  + "\nkey="
  + ((RecordRef) response.getBaseRef()).getInternalId()
  + "\nentityId="
  + customer.getEntityId()
  + "\ncompanyName="
  + customer.getCompanyName()
  + "\nemail="
  + customer.getEmail()
  + "\nstatusKey="
  + customer.getEntityStatus().getInternalId()
  + "\naddressbookList[0].label="
  + customer.getAddressbookList().getAddressbook(0)
  .getLabel());
 } else {
  _console.error("The customer was not added:", true);
  _console.error(getStatusDetails(response.getStatus()));
 }
}
```

addList

The addList operation is used to add one or more new instances of a record to NetSuite. If there are multiple records, they can either be of the same record type or different record types. For example, it's possible to add a customer and a contact within a single request using this operation. However, each record entered must have a unique signature. Adding two records with the same signature results in a SOAP fault. The signature consists of parameters required to identify a record as unique.

For example, in the case of entities, a record is uniquely identified by its name, its type and its parent hierarchy. So you could have two records with the same entityId (or name) belonging to two different record types as follows:

Customer (the type):
John Smith
MyCompany: John Smith

Contact
John Smith

But a second record such as the following would be invalid:

Contact
John Smith

Note: An asynchronous equivalent is available for this operation, **asyncAddList**. For information about asynchronous request processing, see Synchronous Versus Asynchronous Request Processing.

Request

The AddListRequest type is used for the request.

Element Name	XSD Type	Notes
record[]	Record	Contains an array of record objects. The record type is an abstract type so an instance of a type that extends record must be used—such as Customer or Event.

Response

The AddListResponse type is used for the response.

Element Name	XSD Type	Notes
response[]	WriteResponse	Contains an array of WriteResponse objects, each of which contains details on the status of that add operation and a reference to that created record.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault

- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

SOAP Request

In the following example, two customer records are added. When using the addList operation, you can submit two different record types in the same request.

```
<soap:Body>
<platformMsgs:addList>
        <platformMsgs:record xsi:type="listRel:Customer">
                <listRel:entityId>Shutter Fly</listRel:entityId>
                <listRel:companyName>Shutter Fly, Inc</listRel:companyName>
                <listRel:unsubscribe>false</listRel:unsubscribe>
        </platformMsgs:record>
        <platformMsgs:record xsi:type="listRel:Customer">
                <listRel:entityId>GNC</listRel:entityId>
                <listRel:companyName>GNC Corp</listRel:companyName>
                <listRel:unsubscribe>false</listRel:unsubscribe>
        </platformMsgs:record>
</platformMsgs:addList>
</soap:Body>
```

SOAP Response

In the response, notice that the internalID for each record added is returned.

```
<soapenv:Body>
<addListResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
<writeResponseList xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
 <writeResponse>
  <ns1:status isSuccess="true" xmlns:ns1="urn:core_2_5.platform.webservices.netsuite.com"/>
   <baseRef internalId="980" type="customer" xsi:type="ns2:RecordRef"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ns2="urn:core_2_5.platform.webservices.netsuite.com"/>
 </writeResponse>
 <writeResponse>
  <ns3:status isSuccess="true" xmlns:ns3="urn:core_2_5.platform.webservices.netsuite.com"/>
   <baseRef internalId="981" type="customer" xsi:type="ns4:RecordRef"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ns4="urn:core_2_5.platform.webservices.netsuite.com"/>
 </writeResponse>
</writeResponseList>
</addListResponse>
</soapenv:Body>
```

Java Sample

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

                Customer cust1 = new Customer();
                cust1.setEntityId("Shutter Fly");
                cust1.setCompanyName("Shutter Fly, Inc");
                cust1.setUnsubscribe(false);

                Customer cust2 = new Customer();
                cust2.setEntityId("GNC");
                cust2.setCompanyName("GNC Corp");
                cust2.setUnsubscribe(false);

                Customer[] customers = new Customer[2];
                customers[0] = cust1;
                customers[1] = cust2;

                WriteResponseList responseList = _port.addList(customers);
        }
```

attach / detach

The attach and detach operations can be used to define or remove a relationship between two records. For example, a Contact record can be associated with a Partner record, or an Opportunity record can be associated with a Customer record.

Important: A user error is thrown if you attempt to attach files or record that do not exist.

You can also use the attach / detach operations to attach or detach a file to or from a record. Any file that is in the NetSuite file cabinet, for example an MS Word or Excel file or a PDF can be attached to any record other than a custom record.

Note that when attaching Contacts to other entity records, the Contact's role can also be specified during the request. Contact Roles are roles available in a user defined list at List > Relationships > Contacts. This list has been exposed as ContactRole in accounting.xsd.

Note: Contact records can be attached to all entity records expect for other Contact or Group records.

The following table lists all records that support the attach/detach operations. It also lists which records can accept file attachments as well as which records can accept Contact records as attachments.

Record Type	Accepts File Attachments	Accepts Contact Record Attachments
Transactions		
Check	X	X
Expense Report	X	
Inventory Adjustment	X	X

Record Type	Accepts File Attachments	Accepts Contact Record Attachments
Item Fulfillment	X	X
Journal Entry	X	X
Intercompany Journal Entry	X	X
Opportunity	X	X
Sales Order	X	X
Customer Payment	X	X
Return Authorization	X	X
Credit Memo	X	X
Cash Refund	X	X
Estimate	X	X
Invoice	X	X
Cash Sale	X	X
Purchase Order	X	X
Purchase Order Receipt	X	X
Customer Payment	X	X
Customer Refund	X	X
Customer Deposit	X	X
Customer Deposit Application	X	X
Vendor Bill	X	X
Work Order	X	
Work Order Issue	X	
Work Order Close	X	
Work Order Completion	X	
Entities		
Customer	X	X
Contact	X	
Employee	X	
Partner	X	X
Vendor	X	X
Project	X	X
Group	X	**Note:** Although you cannot currently attach Contact records to Group, you can attach members to create a static group.
Activities		
Task	X	

Record Type	Accepts File Attachments	Accepts Contact Record Attachments
Event	X	
Phone Call	X	
Project Task	X	
Support		
SupportCase	X	
Issue	X	
Marketing		
Campaign	X	

Request (attach)

The AttachRequest type is used for this request. It contains the following fields:

Element Name	XSD Type	Notes
attachReference	AttachReference	

Request (detach)

The DetachRequest type is used for this request. It contains the following fields:

Element Name	XSD Type	Notes
detachReference	DetachReference	

Response (attach)

The AttachResponse type is used for this response. It contains the following fields:

Element Name	XSD Type	Notes
response	WriteResponse	

Response (detach)

The DetachResponse type is used for this response. It contains the following fields:

Element Name	XSD Type	Notes
response	WriteResponse	

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

SOAP Request (attach)

```
<attach xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
  <attachReferece xsi:type="ns1:AttachContactReference"
  xmlns:ns1="urn:core_2_6.platform.webservices.netsuite.com">
      <ns1:attachTo internalId="176" type="customer" xsi:type="ns1:RecordRef">
        <ns1:name xsi:type="xsd:string">Adelina Shonkwiler</ns1:name>
          </ns1:attachTo>
       <ns1:contact internalId="1467" xsi:type="ns1:RecordRef"/>
         <ns1:contactRole internalId="-10" xsi:type="ns1:RecordRef">
            <ns1:name xsi:type="xsd:string">Primary Contact</ns1:name>
          </ns1:contactRole>
   </attachReferece>
 </attach>
```

SOAP Request (detach)

```
<detach xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
   <attachReferece xsi:type="ns1:AttachBasicReference"
   xmlns:ns1="urn:core_2_6.platform.webservices.netsuite.com">
          <ns1:attachTo internalId="176" type="customer" xsi:type="ns1:RecordRef">
           <ns1:name xsi:type="xsd:string">Adelina Shonkwiler</ns1:name>
               </ns1:attachTo>
   <ns1:attachedRecord internalId="1467" type="contact" xsi:type="ns1:RecordRef"/>
     </attachReferece>
 </detach>
```

SOAP Response (attach)

```
<attachResponse xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
    <writeResponse>
      <ns1:status isSuccess="true"
    xmlns:ns1="urn:core_2_6.platform.webservices.netsuite.com"/>
           <baseRef internalId="176" type="customer" xsi:type="ns2:RecordRef"
      xmlns:ns2="urn:core_2_6.platform.webservices.netsuite.com">
               <ns2:name>Adelina Shonkwiler</ns2:name>
```

```
            </baseRef>
     </writeResponse>
 </attachResponse>
```

SOAP Response (detach)

```
<detachResponse xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
   <writeResponse>
     <ns1:status isSuccess="true"
   xmlns:ns1="urn:core_2_6.platform.webservices.netsuite.com"/>
          <baseRef internalId="176" type="customer" xsi:type="ns2:RecordRef"
     xmlns:ns2="urn:core_2_6.platform.webservices.netsuite.com">
               <ns2:name>Adelina Shonkwiler</ns2:name>
            </baseRef>
   </writeResponse>
</detachResponse>
```

Java (attach operation)

```
public void attach() throws Exception
    {
        RecordRef contactRef = new RecordRef();
        contactRef.setInternalId("1467");
        contactRef.setType(RecordType.contact);

        RecordRef contactRoleRef = new RecordRef();
        contactRoleRef.setInternalId("-10");
        contactRoleRef.setName("Primary Contact");

        RecordRef customerRef = new RecordRef();
        customerRef.setInternalId("176");
        customerRef.setType(RecordType.customer);

        AttachContactReference attachRef = new AttachContactReference();
        attachRef.setContact(contactRef);
        attachRef.setAttachTo(customerRef);
        attachRef.setContactRole(contactRoleRef);

        WriteResponse attachResponse = sessMgr.getPort().attach(attachRef);
    }
```

Java (detach operation)

```
public void detach() throws Exception
    {
        RecordRef contactRef = new RecordRef();
        contactRef.setInternalId("1467");
        contactRef.setType(RecordType.contact);

        RecordRef customerRef = new RecordRef();
        customerRef.setInternalId("176");
        customerRef.setType(RecordType.customer);

        AttachContactReference detachRef = new AttachContactReference();
        detachRef.setContact(contactRef);
```

```
        detachRef.setAttachTo(customerRef);

        WriteResponse detachResponse = sessMgr.getPort().detach(detachRef);
    }
```

changePasswordOrEmail

The changePasswordOrEmail operation is supported in the 2012.1 and earlier endpoints. Use this operation to change **either** a user's email address or password. You cannot use this operation to change both the email address and password in a single call.

Warning: If you upgrade to the 2012.2 endpoint or a later version, any existing calls to changeEmailOrPassword will no longer work. Instead, you must use the changeEmail operation to change a user's email address, and you must use changePassword to change a user's password

Important: When using the changePasswordOrEmail operation to change email addresses, be aware that the addresses are not changed immediately. The old email address remains in effect until the new email address is validated by the user, through a validation link sent to the new address. A notification is also sent to the old address. This behavior is the same as when users change their email addresses through the user interface.

For details on how to change a user's email address or password using the NetSuite user interface, see the help topic *Resetting Your Login Email or Password*.

Request

The ChangePasswordOrEmailRequest type is used for the request.

Element Name	XSD Type	Notes
changePasswordOrEmailCredentials	ChangePasswordOrEmailCredentials	

The ChangePasswordOrEmailCredentials type takes the following fields:

- currentPassword
- newEmail
- newEmail2
- newPassword
- newPassword2
- justThisAccount

Response

The ChangePasswordOrEmailResponse type is used for the response. It does not contain any fields.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InsufficientPermissionFault
- InvalidAccountFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- UnexpectedErrorFault
- InvalidVersionFault

Example — Changing an Email Address

The following example shows how to use the changeEmailOrPassword operation to change a user's email address.

Java

```
public void test_ChangePwdOrEmail() throws Exception {
 sessMgr.loginWithEmail("oldEmail@ws.com");
 ChangePasswordOrEmailCredentials c = new ChangePasswordOrEmailCredentials();
 c.setCurrentPassword("xxxxxx");
 c.setNewEmail("newEmail@ws.com");
 c.setNewEmail2("newEmail@ws.com");
 sessMgr.getPort().changePasswordOrEmail(c);
 sessMgr.loginWithEmail("newEmail@ws.com");
}
```

SOAP Request

```
<changePasswordOrEmail xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
 <changePasswordOrEmailCredentials>
  <ns1:currentPassword xmlns:ns1="urn:core_2_6.platform.webservices.netsuite.com">xxxxxx< ns1:c
urrentPassword>
  <ns2:newEmail xmlns:ns2="urn:core_2_6.platform.webservices.netsuite.com">newEmail@ws.com<ns2:
newEmail>
  <ns3:newEmail2 xmlns:ns3="urn:core_2_6.platform.webservices.netsuite.com">newEmail@ws.com<ns3
:newEmail2>
 </changePasswordOrEmailCredentials>
</changePasswordOrEmail>
```

SOAP Response

```
<changePasswordOrEmailResponse xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
 <sessionResponse>
  <ns1:status isSuccess="true"xmlns:ns1="urn:core_2_6.platform.webservices.netsuite.com"/>
 </sessionResponse>
</changePasswordOrEmailResponse>
```

Example — Changing a Password

The following example shows how to change a password. Note that with this approach, you have to declare values for NewEmail and NewEmail2 — but you should specify empty strings. These fields are ignored, but you must declare them, or the operation fails.

Java

```
public void changePassword() throws Exception
{
 ChangePasswordOrEmailCredentials newCredentials = new ChangePasswordOrEmailCredentials();
 newCredentials.setCurrentPassword("currentPwd ");
 newCredentials.setNewPassword("newPwd ");
 newCredentials.setNewPassword2("newPwd ");
 newCredentials.setNewEmail("");   // NewEmail is mandatory, and you must specify an empty stri
ng for the operation to complete successfully
 newCredentials.setNewEmail2("");  // NewEmail2 is mandatory, and you must specify an empty str
ing for the operation to complete successfully

 SessionResponse response = sessMgr.getPort().changePasswordOrEmail(newCredentials);
}
```

SOAP Request

```
<changePasswordOrEmail xmlns="urn:messages_2011_1.platform.webservices.netsuite.com">
 <changePasswordOrEmailCredentials>
  <ns7:currentPassword xmlns:ns7="urn:core_2011_1.platform.webservices.netsuite.com">currentPwd
</ns7:currentPassword>
  <ns8:newEmail xmlns:ns8="urn:core_2011_1.platform.webservices.netsuite.com"></ns8:newEmail>
  <ns9:newEmail2 xmlns:ns9="urn:core_2011_1.platform.webservices.netsuite.com"></ns9:newEmail2>

  <ns10:newPassword xmlns:ns10="urn:core_2011_1.platform.webservices.netsuite.com">newPwd</ns10
:newPassword>
  <ns11:newPassword2 xmlns:ns11="urn:core_2011_1.platform.webservices.netsuite.com">newPwd</ns1
1:newPassword2>
 </changePasswordOrEmailCredentials>
</changePasswordOrEmail>
```

changeEmail

Use the changeEmail operation to change a user's email address.

Important: This operation is supported as of the 2012.2 endpoint. The 2012.2 and later endpoints support changeEmail and changePassword operations that replace changePasswordOrEmail. Endpoints 2012.2 and later **do not support** the changePasswordOrEmail operation.

Important: Be aware that a user's email address is not changed immediately. The old email address remains in effect until the new email address is validated by the user, through a validation link sent to the new address. A notification is also sent to the old address. This behavior is the same as when users change their email addresses through the user interface.

This topic provides details about how to change a user's email address with Web services code. For details about how to make this change in the NetSuite user interface, see the help topic *Resetting Your Login Email or Password*.

Request

The ChangeEmailRequest type is used for the request.

Element Name	XSD Type	Notes
changeEmail	ChangeEmail	

The ChangeEmail type takes the following fields:

- currentPassword
- newEmail
- newEmail2
- justThisAccount

Response

The ChangeEmailResponse type is used for the response. It does not contain any fields.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InsufficientPermissionFault
- InvalidAccountFault
- InvalidCredentialsFault

- InvalidVersionFault
- ExceededRequestLimitFault
- UnexpectedErrorFault
- UserError

Sample Code

SOAP Request

```
<changeEmail xmlns="urn:messages_2012_2.platform.webservices.netsuite.com">
 <changeEmail>
  <ns6:currentPassword xmlns:ns6="urn:core_2012_2.platform.webservices.netsuite.com">xxxxxxx</
ns6:currentPassword>
  <ns7:newEmail xmlns:ns7="urn:core_2012_2.platform.webservices.netsuite.com">newEmail@tester.c
om</ns7:newEmail>
  <ns8:newEmail2 xmlns:ns8="urn:core_2012_2.platform.webservices.netsuite.com"> newEmail @teste
r.com</ns8:newEmail2>
  <ns9:justThisAccount xmlns:ns9="urn:core_2012_2.platform.webservices.netsuite.com">true</ns9:
justThisAccount>
 </changeEmail>
</changeEmail>
```

SOAP Response

```
<changeEmailResponse xmlns="urn:messages_2012_2.platform.webservices.netsuite.com">
              <sessionResponse>
                     <platformCore:status isSuccess="true" xmlns:platformCore="urn:core_2012_2.
platform.webservices.netsuite.com"/>
  </sessionResponse>
</changeEmailResponse>
```

Java

```
public void test_ChangeEmail() throws Exception {
 sessMgr.loginWithEmail("oldEmail@ws.com");
 ChangeEmail c = new ChangeEmail();
 c.setCurrentPassword("xxxxxx");
 c.setNewEmail("newEmail@ws.com");
 c.setNewEmail2("newEmail@ws.com");
 c.setJustThisAccount(true);
 sessMgr.getPort().changeEmail(c);
 sessMgr.loginWithEmail("newEmail@ws.com");
}
```

changePassword

Use the changePassword operation to change a user's NetSuite password.

Important: This operation is supported as of the 2012.2 endpoint. The 2012.2 and later endpoints support changeEmail and changePassword operations that replace changePasswordOrEmail. Endpoints 2012.2 and later **do not support** the changePasswordOrEmail operation.

This topic provides details about how to change a user's password with Web services code. For details about how to make this change in the NetSuite user interface, see the help topic *Resetting Your Login Email or Password*.

Request

The ChangePasswordRequest type is used for the request.

Element Name	XSD Type	Notes
changePassword	ChangePassword	

The ChangePassword type takes the following fields:

- currentPassword
- newPassword
- newPassword2
- justThisAccount

Response

The ChangePasswordResponse type is used for the response. It does not contain any fields.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InsufficientPermissionFault
- InvalidAccountFault
- InvalidCredentialsFault
- InvalidVersionFault
- ExceededRequestLimitFault
- UnexpectedErrorFault
- UserError

Sample Code

SOAP Request

```
<changePassword xmlns="urn:messages_2012_2.platform.webservices.netsuite.com">
 <changePassword>
  <ns6:currentPassword xmlns:ns6="urn:core_2012_2.platform.webservices.netsuite.com">foobar1</
ns6:currentPassword>
  <ns7:newPassword xmlns:ns7="urn:core_2012_2.platform.webservices.netsuite.com">foobar2</ns7:n
ewPassword>
  <ns8:newPassword2 xmlns:ns8="urn:core_2012_2.platform.webservices.netsuite.com">foobar2</ns8:
newPassword2>
  <ns9:justThisAccount xmlns:ns9="urn:core_2012_2.platform.webservices.netsuite.com">true</ns9:
justThisAccount>
 </changePassword>
</changePassword>
```

SOAP Response

```
<changePasswordResponse xmlns="urn:messages_2012_2.platform.webservices.netsuite.com">
<sessionResponse>
<platformCore:status isSuccess="true" xmlns:platformCore="urn:core_2012_2.platform.webservices.
netsuite.com"/>
</sessionResponse>
</changePasswordResponse>
```

Java

```
public void test_ChangePwd() throws Exception {
 sessMgr.loginWithEmail("oldEmail@ws.com");
 ChangePassword c = new ChangePassword();
 c.setCurrentPassword("xxxxxx");
 c.setNewPassword("yyyyyyy");
 c.setNewPassword2("yyyyyyy");
 c.setJustThisAccount(true);
 sessMgr.getPort().changePassword(c);
 sessMgr.loginWithEmail("newEmail@ws.com");
}
```

checkAsyncStatus

See the help topic *checkAsyncStatus* in Synchronous Versus Asynchronous Request Processing for details.

delete

The delete operation is used to delete an existing instance of a record in NetSuite. It is identical in terms of functionality to the deleteList operation but only allows one record to be deleted per request.

Request

The DeleteRequest type is used for the request. It contains the following fields.

Element Name	XSD Type	Notes
record	RecordRef	Must contain a reference to an existing instance of a record.

Response

The DeleteResponse type is used for the response. It contains the following fields.

Element Name	XSD Type	Notes
response	WriteResponse	Contains details on the status of the delete operation and a reference to the deleted record.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Note: If you attempt to delete a record that does not exist (meaning no match is found for its key value), an INVALID_KEY_OR_REF error is returned.

Sample Code

SOAP Request

In this example, a single customer record is deleted.

```
<soap:Body>
<platformMsgs:delete>
 <platformMsgs:baseRef internalId="980" type="customer" xsi:type="platformCore:RecordRef"/>
</platformMsgs:delete>
</soap:Body>
```

SOAP Response

```
<soapenv:Body>
<deleteResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
<writeResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
 <ns1:status isSuccess="true" xmlns:ns1="urn:core_2_5.platform.webservices.netsuite.com"/>
 <baseRef internalId="980" type="customer" xsi:type="ns2:RecordRef"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:ns2="urn:core_2_5.platform.webservices.netsuite.com"/>
</writeResponse>
</deleteResponse>
</soapenv:Body>
```

C#

```
private void deleteCustomRecord()
{
 // This operation requires a valid session
 this.login( true );
 CustomRecordRef customRecordRef = new CustomRecordRef();
 //Prompt user for the nsKey for Custom Record type to be deleted
 _out.write( "Enter nsKey for Custom Record type to be deleted: " );
 customRecordRef.typeId = _out.readLn();;
 //Prompt user for nsKey for Custom Record to be deleted
 _out.write( "Enter nsKey for Custom Record to be deleted: " );
 customRecordRef.internalId = _out.readLn();
 System.Console.Out.Write( "Delete the record above? [Y/N]:" );
 String userResponse = _out.readLn().ToUpper();
 // Delete records
 if ( userResponse.Equals( "Y" ) )
 {
  WriteResponse delResponse = _service.delete( customRecordRef );

  // process response
  _out.info( "\nThe following Custom Record deleted:\n" );
  if ( delResponse.status.isSuccess )
  {
   _out.info( "key=" + ((CustomRecordRef) delResponse.baseRef).internalId );
  }
  else
  {
   _out.errorForRecord( getStatusDetails( delResponse.status ) );
  }
 }
 else
 {
  _out.info( "\nSince your answer was not Y, the records were not deleted." );
 }
}
```

Java

```
public void deleteCustomRecord() throws RemoteException,
ExceededUsageLimitFault, UnexpectedErrorFault, InvalidSessionFault,
ExceededRecordCountFault {
 // This operation requires a valid session
```

```
 this.login(true);
 CustomRecordRef customRecordRef = new CustomRecordRef();
 // Prompt user for the nsKey for Custom Record type to be deleted
 _console.write("Enter nsKey for Custom Record type to be deleted: ");
 customRecordRef.setTypeId(_console.readLn());
 // Prompt user for nsKey for Custom Record to be deleted
 _console.write("Enter nsKey for Custom Record to be deleted: ");
 customRecordRef.setInternalId(_console.readLn());
 // customRecordRef.typeSpecified = true;
 _console.write("Delete the record above? [Y/N]:");
 String userResponse = _console.readLn().toUpperCase();
 // Delete records
 if (userResponse.equals("Y")) {
  WriteResponse delResponse = _port.delete(customRecordRef);

  // process response
  _console.info("\nThe following Custom Record deleted:\n");
  if (delResponse.getStatus().isIsSuccess()) {
   _console
   .info("key="
   + ((CustomRecordRef) delResponse.getBaseRef())
   .getInternalId());
  } else {
   _console.errorForRecord(getStatusDetails(delResponse
   .getStatus()));
  }
 } else {
  _console
  .info("\nSince your answer was not Y, the records were not deleted.");
 }
}
```

deleteList

The deleteList operations is used to delete one or more existing instances of a certain record type in NetSuite.

If there are multiple records, they can either be of the same record type or different record types. For example, it's possible to delete a customer and a contact within a single request using this operation.

Note: An asynchronous equivalent is available for this operation, **asyncDeleteList**. For information about asynchronous request processing, see Synchronous Versus Asynchronous Request Processing.

Request

The DeleteListRequest type is used for the request. It contains the following fields.

Element Name	XSD Type	Notes
record[]	RecordRef	Must contain a reference to an existing instance of a record.

Response

The DeleteListResponse type is used for the response. It contains the following fields.

Element Name	XSD Type	Notes
response[]	WriteResponse	Contains an array of WriteResponse objects, each of which contains details on the status of the delete operation and a reference to the deleted record.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

SOAP Request

In the following example, two customer records are deleted. You can delete records of different types in the same request.

```
<soap:Body>
<platformMsgs:deleteList>
 <platformMsgs:baseRef internalId="981" type="customer" xsi:type="platformCore:RecordRef"/>
 <platformMsgs:baseRef internalId="982" type="customer" xsi:type="platformCore:RecordRef"/>
</platformMsgs:deleteList>
</soap:Body>
```

SOAP Response

Note that the status for each item in the request is returned. If a single item in the list errors, the other records submitted will still be processed.

```
<soapenv:Body>
<deleteListResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
```

```
<writeResponseList xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
 <writeResponse>
  <ns1:status isSuccess="true" xmlns:ns1="urn:core_2_5.platform.webservices.netsuite.com"/>
  <baseRef internalId="981" type="customer" xsi:type="ns2:RecordRef"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns2="urn:core_2_5.platform.webservices.netsuite.com"/>
 </writeResponse>
 <writeResponse>
  <ns3:status isSuccess="true" xmlns:ns3="urn:core_2_5.platform.webservices.netsuite.com"/>
  <baseRef internalId="982" type="customer" xsi:type="ns4:RecordRef"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns4="urn:core_2_5.platform.webservices.netsuite.com"/>
 </writeResponse>
</writeResponseList>
</deleteListResponse>
</soapenv:Body>
```

C#

```
private void deleteCustomerList()
{
 // This operation requires a valid session
 this.login( true );

 // Prompt for list of nsKeys and put in an array
 _out.write( "\nEnter nsKeys for customer records to be deleted (separated by commas): " );
 String reqKeys = _out.readLn();
 string [] nsKeys = reqKeys.Split( new Char[] {','} );

 // First get the records from NS
 _out.write( "\nChecking validity of nsKeys by using getList() to retrieve records ...\n" );
 int numRecords = getCustomerList( nsKeys, true );

 // Delete records, but only if there are records to delete
 if ( numRecords > 0 )
 {
  // Build an array of RecordRef objects
  RecordRef[] recordRefs = new RecordRef[ nsKeys.Length ];
  for (int i=0; i<nsKeys.Length; i++ )
  {
   RecordRef recordRef = new RecordRef();
   recordRef.internalId = nsKeys[i];
   recordRefs[i] = recordRef;
   recordRefs[i].type = RecordType.customer;
   recordRefs[i].typeSpecified = true;
  }

  System.Console.Out.Write( "\nDelete all the records above? [Y/N]:" );
  String userResponse = _out.readLn().ToUpper();
  System.Console.Out.Write( "\n" );

  // Delete records
  if ( userResponse.Equals( "Y" ) )
  {
   WriteResponse[] delResponses = _service.deleteList( recordRefs );

   // process response
   _out.info( "\nThe following customers were deleted:\n" );
   for (int i=0; i<delResponses.Length; i++ )
```

```
    {
     if ( delResponses[i].status.isSuccess )
     {
      _out.info( "Customer[" + i + "]:" );
      _out.info( "key=" + ((RecordRef) delResponses[i].baseRef).internalId );
     }
     else
     {
      _out.info( "Customer[" + i + "]:" );
      _out.errorForRecord( getStatusDetails( delResponses[i].status ) );
     }
    }
   }
   else
   {
    _out.info( "\nSince your answer was not Y, the records were not deleted." );
   }
  }
  else
  {
   _out.info( "\nThere were no valid records to be deleted." );
  }
}
```

Java

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

 // Prompt for list of nsKeys and put in an array
 _console
 .write("\nEnter nsKeys for customer records to be deleted (separated by commas): ");
 String reqKeys = _console.readLn();
 String[] nsKeys = reqKeys.split(",");

 // First get the records from NS
 _console
 .write("\nChecking validity of nsKeys by using getList() to retrieve records ...\n");
 int numRecords = getCustomerList(nsKeys, true);

 // Delete records, but only if there are records to delete
 if (numRecords > 0) {
  // Build an array of RecordRef objects
  RecordRef[] recordRefs = new RecordRef[nsKeys.length];
  for (int i = 0; i < nsKeys.length; i++) {
   RecordRef recordRef = new RecordRef();
   recordRef.setInternalId(nsKeys[i]);
   recordRefs[i] = recordRef;
   recordRefs[i].setType(RecordType.customer);
  }

  _console.write("\nDelete all the records above? [Y/N]:");
  String userResponse = _console.readLn().toUpperCase();
  _console.write("\n");

  // Delete records
```

```
  if (userResponse.equals("Y")) {
   WriteResponseList delResponseList = _port
   .deleteList(recordRefs);

   // process response
   WriteResponse[] delResponses = delResponseList
   .getWriteResponse();
   _console.info("\nThe following customers were deleted:\n");
   for (int i = 0; i < delResponses.length; i++) {
    if (delResponses[i].getStatus().isIsSuccess()) {
     _console.info("Customer[" + i + "]:");
     _console.info("key="
     + ((RecordRef) delResponses[i].getBaseRef())
     .getInternalId());
    } else {
     _console.info("Customer[" + i + "]:");
     _console
     .errorForRecord(getStatusDetails(delResponses[i]
     .getStatus()));
    }
   }
  } else {
   _console
   .info("\nSince your answer was not Y, the records were not deleted.");
  }
 } else {
  _console.info("\nThere were no valid records to be deleted.");
 }
}
```

get

The get operation is used to retrieve a record by providing the unique id that identifies that record.

Request

The getRequest type is used for the request. It contains the following fields.

Element Name	XSD Type	Notes
recordRef	RecordRef	A recordRef object that specifies the id of the record to be retrieved.

Response

The getResponse type is used for the response. It contains the following fields.

Element Name	XSD Type	Notes
status	Status	The status for this operation. All applicable errors or warnings are listed within this type.
record	Record	A record that represents the specified id. The actual record returned needs to be a type that extends the abstract type Record.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

SOAP Request

In this example, a single customer record is retrieved. Note that the internal ID for the specific instance of the record and the record type (customer) must be specified.

```
<soap:Body>
<platformMsgs:get>
 <platformMsgs:baseRef internalId="983" type="customer" xsi:type="platformCore:RecordRef">
  <platformCore:name/>
 </platformMsgs:baseRef>
</platformMsgs:get>
</soap:Body>
```

SOAP Response

```
<soapenv:Body>
<getResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
<readResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
 <ns1:status isSuccess="true" xmlns:ns1="urn:core_2_5.platform.webservices.netsuite.com"/>
 <record internalId="983" xsi:type="ns2:Customer"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:ns2="urn:relationships_2_5.lists.webservices.netsuite.com">
  <ns2:entityId>Shutter Fly</ns2:entityId><ns2:isInactive>false</ns2:isInactive>
  <ns2:companyName>Shutter Fly, Inc</ns2:companyName>
  <ns2:entityStatus internalId="6">
  .
  ...[more fields]
  .
  <ns2:customFieldList>
   <ns6:customField internalId="custentity_map" xsi:type="ns6:StringCustomFieldRef"
   xmlns:ns6="urn:core_2_5.platform.webservices.netsuite.com">
    <ns6:value>http://maps.google.com</ns6:value>
```

```
    </ns6:customField>
   </ns2:customFieldList>
  </record>
</readResponse>
</getResponse>
</soapenv:Body>
```

C#

```
private void getCustomer()
{
 // This operation requires a valid session
 this.login( true );

 // Prompt for the nsKey
 _out.write( "\nnsKey for record to be retrieved: " );
 String nsKey = _out.readLn();

 // Invoke the get() operation to retrieve the record
 RecordRef recordRef = new RecordRef();
 recordRef.internalId = nsKey;
 recordRef.type = RecordType.customer;
 recordRef.typeSpecified = true;

 ReadResponse response = _service.get( recordRef );

 // Process response from get() operation
 _out.info( "\nRecord returned from get() operation: " );
 if ( !response.status.isSuccess )
 {
  _out.info(
  "ERROR: " +
  getStatusDetails( response.status ) );
 }
 else
 {
  Customer customer = (Customer) response.record;
  _out.info(
  "\nnsKey=" + customer.internalId + ", " +
  "\nentityId=" + customer.entityId +
  (customer.companyName==null ? "" : ("\ncompanyName=" + customer.companyName)) +
  (customer.stage==null ? "" : ("\nstage=" + customer.stage)) +
  (customer.email==null ? "" : ("\nemail=" + customer.email)) +
  (customer.phone==null ? "" : ("\nphone=" + customer.phone)) +
  "\nisInactive=" + customer.isInactive +
  (!customer.dateCreatedSpecified ? "" : ("\ndateCreated=" +
  customer.dateCreated.ToShortDateString())) );
 }
}
```

Java

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

```
 // Prompt for the nsKey
 _console.write("\nnsKey for record to be retrieved: ");
 String nsKey = _console.readLn();

 // Invoke the get() operation to retrieve the record
 RecordRef recordRef = new RecordRef();
 recordRef.setInternalId(nsKey);
 recordRef.setType(RecordType.customer);

 ReadResponse response = _port.get(recordRef);

 // Process response from get() operation
 _console.info("\nRecord returned from get() operation: ");
 if (!response.getStatus().isIsSuccess()) {
  _console.info("ERROR: " + getStatusDetails(response.getStatus()));
 } else {
  Customer customer = (Customer) response.getRecord();
  _console
  .info("\nnsKey="
  + customer.getInternalId()
  + ", "
  + "\nentityId="
  + customer.getEntityId()
  + (customer.getCompanyName() == null ? ""
  : ("\ncompanyName=" + customer
  .getCompanyName()))
  + (customer.getStage() == null ? ""
  : ("\nstage=" + customer.getStage()))
  + (customer.getEmail() == null ? ""
  : ("\nemail=" + customer.getEmail()))
  + (customer.getPhone() == null ? ""
  : ("\nphone=" + customer.getPhone()))
  + "\nisInactive="
  + customer.getIsInactive()
  + (customer.getDateCreated() != null ? ""
  : ("\ndateCreated=" + customer
  .getDateCreated().toString())));
 }
}
```

getAll

The getAll operation is used to retrieve a list of all records of the specified type. Records that support the getAll operation are listed in the GetAllRecordType, as defined in the platformCoreType system constants XSD file.

Note: You cannot use the search operation to retrieve state values. You must use the getAll operation. The getAll 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.

Request

The getAllRequest type is used for the request. It contains the following fields.

Element Name	XSD Type	Notes
recordType	GetAllRecordType	Specify the record type.

Response

The getList response type is used for the response. It contains the following fields.

Element Name	XSD Type	Notes
status	Status	The status for this operation. All applicable errors or warnings are listed within this type.
recordList	Record[]	A list of records that correspond to the specified ids. The actual records returned need to be of a type that extends the abstract type Record.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

getAsyncResult

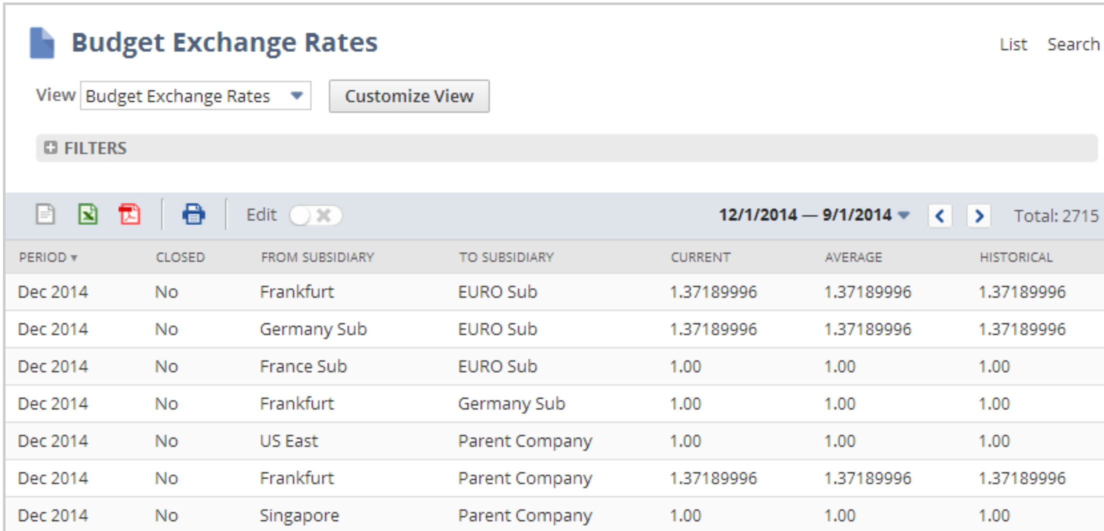
See Checking for Detailed Results of an Asynchronous Job in Asynchronous Request Processing for details.

getBudgetExchangeRate

On the Budget Exchange Rates table, you can maintain exchange rates between the root-parent and child subsidiaries for use in the budgeting process. Use the getBudgetExchangeRate operation to get and filter all data related to this table.

Important: This operation can be used only in NetSuite OneWorld accounts.

In the UI, you can see the Budget Exchange Rates table by going to List > Accounting > Budget Exchange Rates. Note that in Web services, this table is read-only.



PERIOD	CLOSED	FROM SUBSIDIARY	TO SUBSIDIARY	CURRENT	AVERAGE	HISTORICAL
Dec 2014	No	Frankfurt	EURO Sub	1.37189996	1.37189996	1.37189996
Dec 2014	No	Germany Sub	EURO Sub	1.37189996	1.37189996	1.37189996
Dec 2014	No	France Sub	EURO Sub	1.00	1.00	1.00
Dec 2014	No	Frankfurt	Germany Sub	1.00	1.00	1.00
Dec 2014	No	US East	Parent Company	1.00	1.00	1.00
Dec 2014	No	Frankfurt	Parent Company	1.37189996	1.37189996	1.37189996
Dec 2014	No	Singapore	Parent Company	1.00	1.00	1.00

For general information on the Budget Exchange Rate table, see the help topic *Using Budget Exchange Rates*.

Request

The GetBudgetExchangeRateRequest type is used for the request.

Element Name	XSD Type	Notes
budgetExchangeRateFilter	BudgetExchageRateFilter	You can filter the returned exchange rates for a budget using this filter.

BudgetExchangeRateFilter

Element Name	XSD Type	Notes
period	RecordRef	References an existing period. This argument is required.
fromSubsidiary	RecordRef	References the receiving subsidiary. This argument is optional.
toSubsidiary	RecordRef	References the originating subsidiary. This argument is optional.

Response

The GetBudgetExchangeRateResult type is used for the response.

Element Name	XSD Type	Notes
status	Status	The status for this operation. All applicable errors or warnings are listed within this type.
budgetExchangeRateList	BudgetExchangeRateList	Returns a list of available exchange rates in a budget.

BudgetExchangeRateList

Element Name	XSD Type	Notes
budgetExchangeRate	BudgetExchangeRate	References the exchange rate for a budget.

BudgetExchangeRate

Element Name	XSD Type	Notes
period	RecordRef	References an existing period.
fromSubsidiary	RecordRef	References the receiving subsidiary.
toSubsidiary	RecordRef	References the originating subsidiary.
currentRate	double	References the current rate.
averageRate	double	References the average rate.
historicalRate	double	References the historical rate.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

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>
         <getBudgetExchangeRate xmlns="urn:messages_2009_1.platform.webservices.netsuite.com">

            <budgetExchangeRateFilter>
```

```
                <ns1:period internalId="3" xmlns:ns1="urn:core_2009_1.platform.webservices.netsu
ite.com"/>
                <ns2:fromSubsidiary internalId="4" xmlns:ns2="urn:core_2009_1.platform.webservic
es.
  netsuite.com"/>
            </budgetExchangeRateFilter>
         </getBudgetExchangeRate>
      </soapenv:Body>
   </soapenv:Envelope>
```

SOAP Response

```
<?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>
         <getBudgetExchangeRateResponse xmlns="urn:messages_2009_1.platform.webservices.netsuit
e.com">
            <platformCore:getBudgetExchangeRateResult
  xmlns:platformCore="urn:core_2009_1.platform.webservices.netsuite.com">
               <platformCore:status isSuccess="true"/>
               <platformCore:budgetExchangeRateList>
                  <platformCore:budgetExchangeRate>
                     <platformCore:period internalId="3"/>
                     <platformCore:fromSubsidiary internalId="4"/>
                     <platformCore:toSubsidiary internalId="1"/>
                     <platformCore:currentRate>1.9586</platformCore:currentRate>
                     <platformCore:averageRate>1.9586</platformCore:averageRate>
                     <platformCore:historicalRate>1.9586</platformCore:historicalRate>
                  </platformCore:budgetExchangeRate>
               </platformCore:budgetExchangeRateList>
            </platformCore:getBudgetExchangeRateResult>
         </getBudgetExchangeRateResponse>
      </soapenv:Body>
   </soapenv:Envelope>
```

Java

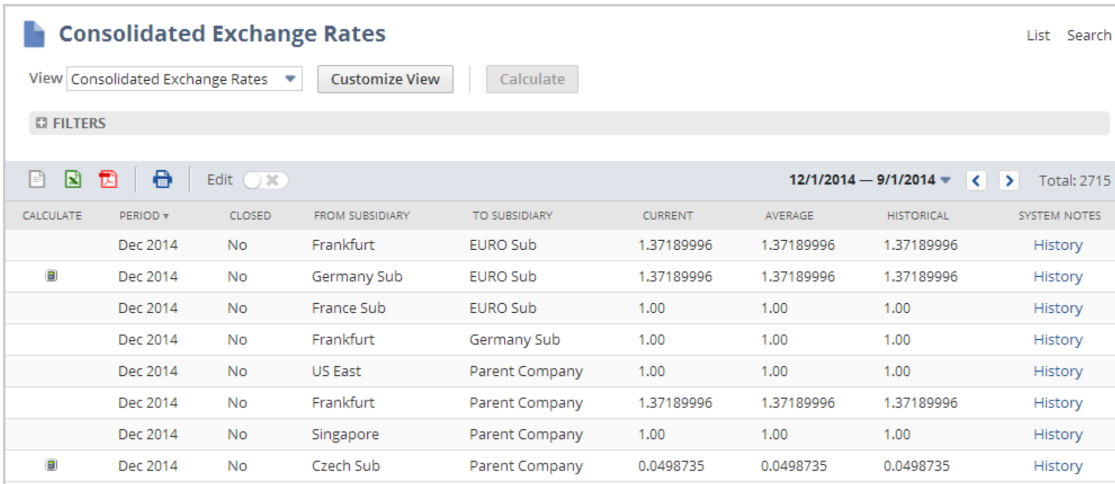
```
/* Make Record Ref out of an internalId */
 public static RecordRef mrr(String internalId)
 {
  RecordRef toRet = new RecordRef();
  toRet.setInternalId(internalId);
  return toRet;
 }

c.getPort().getBudgetExchangeRate(new BudgetExchangeRateFilter(mrr("3"),mrr("4"),null));
```

getConsolidatedExchangeRate

On the Consolidated Exchange Rates table, you can maintain exchange rates between the root-parent and child subsidiaries. Use the getConsolidatedExchangeRate operation to get and filter all data related to this table.

Important: This operation can be used only in NetSuite OneWorld acccounts.

In the UI, you can see the Consolidated Exchange Rates table by going to List > Accounting > Consolidated Exchange Rates. Note that in Web services, this table is read-only.


Consolidated Exchange Rates

List Search

View Consolidated Exchange Rates | Customize View | Calculate

FILTERS

Edit | 12/1/2014 — 9/1/2014 | Total: 2715

CALCULATE	PERIOD	CLOSED	FROM SUBSIDIARY	TO SUBSIDIARY	CURRENT	AVERAGE	HISTORICAL	SYSTEM NOTES
	Dec 2014	No	Frankfurt	EURO Sub	1.37189996	1.37189996	1.37189996	History
	Dec 2014	No	Germany Sub	EURO Sub	1.37189996	1.37189996	1.37189996	History
	Dec 2014	No	France Sub	EURO Sub	1.00	1.00	1.00	History
	Dec 2014	No	Frankfurt	Germany Sub	1.00	1.00	1.00	History
	Dec 2014	No	US East	Parent Company	1.00	1.00	1.00	History
	Dec 2014	No	Frankfurt	Parent Company	1.37189996	1.37189996	1.37189996	History
	Dec 2014	No	Singapore	Parent Company	1.00	1.00	1.00	History
	Dec 2014	No	Czech Sub	Parent Company	0.0498735	0.0498735	0.0498735	History

For general information on currencies and working with the Consolidated Exchange Rate table, see the following topics in the NetSuite Help Center:

- *Creating Currency Records*
- *Setting Currency Exchange Rates*

Note: If you choose, you can set exchange rates in the Consolidated Exchange Rates table directly. See the steps under “To set exchange rates in the Consolidated Exchange Rates table” for details.

- *Using the Currency Exchange Rate Integration Feature*
- *Working with Currencies*

Request

The GetConsolidatedExchangeRateRequest type is used for the request.

Element Name	XSD Type	Notes
consolidatedExchangeRateFilter	ConsolidatedExchangeRateFilter	You can filter the returned consolidated exchange rates using this filter.

ConsolidatedExchangeRateFilter

Element Name	XSD Type	Notes
period	RecordRef	References an existing period. This argument is required.
fromSubsidiary	RecordRef	References the receiving subsidiary. This argument is optional.
toSubsidiary	RecordRef	References the originating subsidiary. This argument is optional.

Response

The GetConsolidatedExchangeRateResult type is used for the response.

Element Name	XSD Type	Notes
status	Status	The status for this operation. All applicable errors or warnings are listed within this type.
consolidatedExchangeRateList	ConsolidatedExchangeRateList	Returns a list of available consolidated exchange rates.

ConsolidatedExchangeRateList

Element Name	XSD Type	Notes
consolidatedExchangeRate	ConsolidatedExchangeRate	References an existing period.

ConsolidatedExchangeRate

Element Name	XSD Type	Notes
period	RecordRef	References an existing period.
fromSubsidiary	RecordRef	References the receiving subsidiary.
toSubsidiary	RecordRef	References the originating subsidiary.
currentRate	double	References the current rate.
averageRate	double	References the average rate.
historicalRate	double	References the historical rate.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault

- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

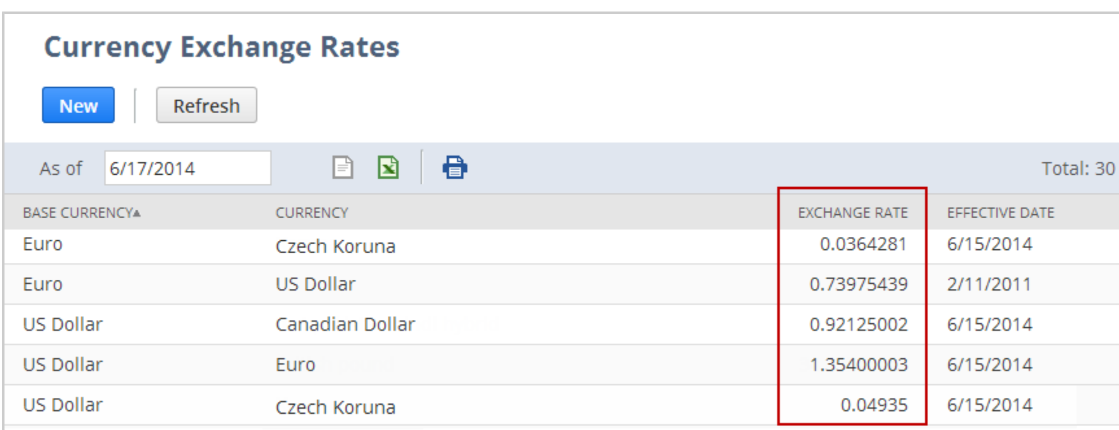
The getConsolidatedExchangeRate and getBudgetExchangeRate operations are used in the same way. Therefore, please refer to the Sample Code for getBudgetExchangeRate.

getCurrencyRate

Use this operation to get the exchange rate between two currencies based on a certain date. The exchange rate values you are getting are those that appear in the Exchange Rate column of the Currency Exchange Rates table, which you can view at Lists > Accounting > Currency Exchange Rates.

The role used to execute this operation must have at least View level of the Currency permission (a Lists type permission).

Note: The Multiple Currencies feature must be enabled in your account before using the getCurrencyRate operation. For information on enabling this feature, see the help topic *Enabling the Multiple Currencies Feature*.


Currency Exchange Rates

New | Refresh

As of 6/17/2014 Total: 30

BASE CURRENCY	CURRENCY	EXCHANGE RATE	EFFECTIVE DATE
Euro	Czech Koruna	0.0364281	6/15/2014
Euro	US Dollar	0.73975439	2/11/2011
US Dollar	Canadian Dollar	0.92125002	6/15/2014
US Dollar	Euro	1.35400003	6/15/2014
US Dollar	Czech Koruna	0.04935	6/15/2014

Request

The GetCurrencyRateRequest type is used for the request. This method accepts the argument CurrencyRateFilter. With this filter you specify what you want returned in the results.

If you do not specify anything in the filter, the resulting CurrencyRateList will contain all currency exchange rates (all combinations) for the actual date. For example, if you specify baseCurrency only (for example, Euro), you will get all rates for Euro {USD,YEN,SING,....}.

Element Name	XSD Type	Notes
currencyRateFilter	CurrencyRateFilter	Filter the returned currency exchange rates using this filter.

CurrencyRateFilter

Element Name	XSD Type	Notes
baseCurrency	RecordRef	References the base currency. This argument is optional.
fromCurrency	RecordRef	References the from or "exchange" currency. This argument is optional.
effectiveDate	dateTime	This argument is optional. If a date is not provided, the date defaults to the current date.

Response

The GetCurrencyRateResponse type is used for the response.

Element Name	XSD Type	Notes
status	Status	The status for this operation. All applicable errors or warnings are listed within this type.
currencyRate	double	Returns a list of available currency exchange rates.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- InsufficientPermissionFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault

- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

SOAP Request

```
<soapenv:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/200
1/XMLSchema-instance" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header>
        <ns1:passport soapenv:mustUnderstand="0" soapenv:actor="http://schemas.xmlsoap.org/soap
/actor/next" xmlns:ns1="urn:messages_2012_2.platform.webservices.netsuite.com">
            <ns2:email xmlns:ns2="urn:core_2012_2.platform.webservices.netsuite.com">email@nets
uite.com</ns2:email>
            <ns3:password xmlns:ns3="urn:core_2012_2.platform.webservices.netsuite.com">*******
</ns3:password>
            <ns4:account xmlns:ns4="urn:core_2012_2.platform.webservices.netsuite.com">1234567<
/ns4:account>
            <ns5:role internalId="3" xmlns:ns5="urn:core_2012_2.platform.webservices.netsuite.c
om"/>
        </ns1:passport>
    </soapenv:Header>
    <soapenv:Body>
        <getCurrencyRate xmlns="urn:messages_2012_2.platform.webservices.netsuite.com">
            <currencyRateFilter>
                <ns6:baseCurrency internalId="2" xmlns:ns6="urn:core_2012_2.platform.webservice
s.netsuite.com"/>
                <ns7:fromCurrency internalId="1" xmlns:ns7="urn:core_2012_2.platform.webservice
s.netsuite.com">
                <ns8:effectiveDate xmlns:ns8="urn:core_2012_2.platform.webservices.netsuite.com
">2012-08
   11T06:58:59.628Z</ns8:effectiveDate>
            </currencyRateFilter>
        </getCurrencyRate>
    </soapenv:Body>
</soapenv:Envelope>
```

SOAP Response

```
<soapenv:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/200
1/XMLSchema-instance" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header>
        <platformMsgs:documentInfo xmlns:platformMsgs="urn:messages_2012_2.platform.webservices
.netsuite.com">
            <platformMsgs:nsId>WEBSERVICES_1326288_08092012257172984776216537_6c7c330232bb5</pl
atformMsgs:nsId>
        </platformMsgs:documentInfo>
    </soapenv:Header>
    <soapenv:Body>
        <getCurrencyRateResponse xmlns="urn:messages_2012_2.platform.webservices.netsuite.com">
```

```
            <platformCore:getCurrencyRateResult xmlns:platformCore="urn:core_2012_2.platform.we
bservices.netsuite.com">
                <platformCore:status isSuccess="true"/>
                <platformCore:currencyRateList>
                    <platformCore:currencyRate>
                        <platformCore:baseCurrency internalId="2"/>
                        <platformCore:fromCurrency internalId="1"/>
                        <platformCore:exchangeRate>0.42283298</platformCore:exchangeRate>
                        <platformCore:effectiveDate>1970-01-01T00:00:00.000-08:00</platformCore
:effectiveDate>
                    </platformCore:currencyRate>
                </platformCore:currencyRateList>
            </platformCore:getCurrencyRateResult>
        </getCurrencyRateResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

Java

This sample shows how to get the actual rate for tomorrow and for USD to GBP.

```
    public void getCurrencyRate() throws Exception
 {
  this.login();

  CurrencyRateFilter crf = new CurrencyRateFilter();
  Calendar cal = Calendar.getInstance();
  cal.add(Calendar.DAY_OF_YEAR, +1); //tomorrow
  crf.setEffectiveDate(cal);

  RecordRef currencyRef1 = new RecordRef();
  currencyRef1.setInternalId("1"); //USD
  crf.setBaseCurrency(currencyRef1);

  RecordRef currencyRef2 = new RecordRef();
  currencyRef2.setInternalId("2"); //GBP
  crf.setFromCurrency(currencyRef2);

  GetCurrencyRateResult gcrr = _port.getCurrencyRate(crf);
  CurrencyRateList crl = gcrr.getCurrencyRateList();
  CurrencyRate cr = crl.getCurrencyRate(0);
  Double rate = cr.getExchangeRate();

  System.out.println("Exchange rate is = " + rate);
 }
```

Note: The mrr() method is a convenience method for specifying a record without the need to indicate a record type.

```
/* Make Record Ref out of an internalId */
        public static RecordRef mrr(String internalId)
        {
                RecordRef toRet = new RecordRef();
                toRet.setInternalId(internalId);
                return toRet;
```

```
    }
```

getCustomization

The getCustomization operation is obsolete. It was replaced in version 2010.1 with getCustomizationId.

getCustomizationId

When integrating with NetSuite through Web services, in many cases, you will want to know which custom objects exist in an account. You may also want to obtain metadata for these custom objects so that your application can handle any business logic that is specific to the account.

To learn which custom objects exist in an account, use the getCustomizationId operation to retrieve the internalIds, externalIds, and/or scriptIds of all custom objects of a specified type. These types, enumerated in CustomizationType, include the following:

- crmCustomField
- customList
- customRecordType
- entityCustomField
- itemCustomField
- itemOptionCustomField
- otherCustomField
- transactionBodyCustomField
- transactionColumnCustomField

Note: The CustomizationType object is defined in the coreTypes XSD.

Once the IDs are returned, you can then pass the ID value(s) to either the get or getList operation to obtain metadata for specific custom objects (see Using get and getList with getCustomizationId for more details). If you are returning large sets of metadata, the getCustomizationId operation allows you to add pagination to break your response into multiple pieces.

Note the following:

- To see the UI equivalent of a NetSuite custom object, go to Customization > Lists, Records, & Fields > [custom object type]. For information on working with each object type in Web services, see the help topic *SuiteBuilder Overview*. For a more general understanding of the NetSuite customization model, see the help topic *SuiteBuilder (Customization)*.

- Normally, you cannot add or update the internalId of a NetSuite object. Custom objects, however, are an exception to this rule. You can specify the ID on add, but you cannot update the ID thereafter. Also note that internalIds for custom objects can be set to any unique alphanumeric string up to 30 characters long. This string cannot include any spaces, but it can include underscores (_).

Using get and getList with getCustomizationId

If you want only the internalIds, externalIds, and/or scriptIds for custom objects of a specific type, you can call getCustomizationId on that custom type (for example, crmCustomField). However, if you also want the metadata for any or all of the custom objects referenced in the getCustomizationId response, you must invoke a get or getList request and specify each object's ID.

When using either the get or getList operations in conjunction with getCustomizationId, be aware of the following:

- If you specify more than one ID in a get | getList call, the IDs you pass are read in the following order: internalId, externalId, scriptId.
- If an invalid scriptId is specified but the internalId or externalId is correct, the scriptId is ignored.
- NetSuite will reject get | getList requests if only a scriptId is passed for record types that do not support scriptIds.

Request

The getCustomizationIdRequest type is used for this request. It contains the following fields:

Element Name	XSD Type	Notes
customizationType	CustomizationType	Any of the custom object types enumerated in CustomizationType.
includeInactives	boolean	A value of true or false is required. A value of false means no inactive custom objects are returned in the response. A value of true means that both active and inactive custom objects are returned in the response. **Important:** • To determine which of the returned custom objects are inactive, you must perform either a get or getList operation to obtain the metadata for the custom object type(s). For custom records and custom lists, the value is specified in the **isInactive** element.

Element Name	XSD Type	Notes
		• Although the includeInactives argument is required, the value you set applies to custom record and custom list objects only. In NetSuite, custom fields cannot be set to active or inactive. Therefore, when using getCustomizationId on a custom field type, the value you specify for includeInactives will be ignored. Whether you specify true or false, all custom field objects (of the type specified in your request) will be returned.

Response

The getCustomizationIdResult type is used for the response. It contains the following fields:

Element Name	XSD Type	Notes
status	Status	The status for this operation. All applicable errors or warnings are listed within this type.
totalRecords	xsd:int	The total number of records for this request. Depending on the pageSize value, some or all the records may be returned in the response.
customizationRefList	CustomizationRef	A list of custom objects that correspond to the specified customization type. Also returns the internalId, externalId, and/or scriptId of each object.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

The following sample is a two-part sample. The first part of the sample shows how to construct a getCustomizationId request so that the IDs for all custom record type objects in a

NetSuite account are returned. Notice that in the getCustomizationId request, the value of the includeInactives parameter is set to false, meaning that only custom records marked as “active” in the account will be returned.

The second part of the sample shows how to take internalIds that are returned, and make a getList request to return the metadata for each custom record type.

C# Sample

```
NetSuiteService nss = new NetSuiteService();

// credential code ignored here in this sample ...

// Perform getCustomizationId on custom record type
CustomizationType ct = new CustomizationType();
ct.getCustomizationTypeSpecified=true;
ct.getCustomizationType = GetCustomizationType.customRecordType;

// Retrieve active custom record type IDs. The includeInactives param is set to false.
GetCustomizationIdResult getCustIdResult = nss.getCustomizationId(ct, false);

// Retrieve the metadata of the returned custom record types
ReadResponse [] readResp = nss.getList(getCustIdResult.customizationRefList);
```

SOAP Request (getCustomizationId)

```
        <getCustomizationId xmlns="urn:messages_2010_1.platform.webservices.netsuite.com">
           <customizationType getCustomizationType="customRecordType" />
           <includeInactives>false</includeInactives>
        </getCustomizationId>
```

SOAP Request (getList)

```
           <getList xmlns="urn:messages_2010_1.platform.webservices.netsuite.com">
              <baseRef xmlns:q1="urn:core_2010_1.platform.webservices.netsuite.com"
  xsi:type="q1:CustomizationRef" internalId="15" scriptId="customrecord15"
     type="customRecordType">
                 <q1:name>Customer Satisfaction Survey</q1:name>
              </baseRef>
           </getList>
```

SOAP Response (getCustomizationId)

```
<getCustomizationIdResponse xmlns="urn:messages_2010_1.platform.webservices.netsuite.com">
            <platformCore:getCustomizationIdResult
  xmlns:platformCore="urn:core_2010_1.platform.webservices.netsuite.com">
               <platformCore:status isSuccess="true"/>
               <platformCore:totalRecords>1</platformCore:totalRecords>
               <platformCore:customizationRefList>
                  <platformCore:customizationRef internalId="15" scriptId="customrecord15"
```

```
   type="customRecordType">
                      <platformCore:name>Customer Satisfaction Survey</platformCore:name>
                   </platformCore:customizationRef>
                </platformCore:customizationRefList>
             </platformCore:getCustomizationIdResult>
          </getCustomizationIdResponse>
```

SOAP Response (getList)

```
 <getListResponse xmlns="urn:messages_2010_1.platform.webservices.netsuite.com">
               <readResponseList>
                  <readResponse>
                     <platformCore:status isSuccess="true"
   xmlns:platformCore="urn:core_2010_1.platform.webservices.netsuite.com"/>
                     <record internalId="15" xsi:type="setupCustom:CustomRecordType"
   xmlns:setupCustom="urn:customization_2010_1.setup.webservices.netsuite.com">
                        <setupCustom:recordName>Customer Satisfaction Survey</setupCustom:recor
dName>
                        <setupCustom:includeName>true</setupCustom:includeName>
                        <setupCustom:showId>true</setupCustom:showId>
                        <setupCustom:showCreationDate>true</setupCustom:showCreationDate>
                        <setupCustom:showCreationDateOnList>true</setupCustom:showCreationDateO
nList>
                        <setupCustom:showLastModified>true</setupCustom:showLastModified>
                        <setupCustom:showLastModifiedOnList>true</setupCustom:showLastModifiedO
nList>
                        <setupCustom:showOwner>false</setupCustom:showOwner>
                        <setupCustom:showOwnerOnList>false</setupCustom:showOwnerOnList>
                        <setupCustom:showOwnerAllowChange>false</setupCustom:showOwnerAllowChan
ge>
                        <setupCustom:usePermissions>false</setupCustom:usePermissions>
                        <setupCustom:allowAttachments>false</setupCustom:allowAttachments>
                        <setupCustom:showNotes>true</setupCustom:showNotes>
                        <setupCustom:enableMailMerge>false</setupCustom:enableMailMerge>
                        <setupCustom:isOrdered>false</setupCustom:isOrdered>
                        <setupCustom:allowInlineEditing>false</setupCustom:allowInlineEditing>
                        <setupCustom:isAvailableOffline>false</setupCustom:isAvailableOffline>
                        <setupCustom:allowQuickSearch>false</setupCustom:allowQuickSearch>
                        <setupCustom:isInactive>false</setupCustom:isInactive>
                        <setupCustom:disclaimer><b>Record numbers cannot be reverted back to na
mes once
    updated.<br/></setupCustom:disclaimer>
                        <setupCustom:enableNumbering>false</setupCustom:enableNumbering>
                        <setupCustom:numberingCurrentNumber>413</setupCustom:numberingCurrentNu
mber>
                        <setupCustom:allowNumberingOverride>false</setupCustom:allowNumberingOv
erride>
                        <setupCustom:isNumberingUpdateable>false</setupCustom:isNumberingUpdate
able>
                        <setupCustom:scriptId>customrecord15</setupCustom:scriptId>
                        <setupCustom:fieldList>
                           <setupCustom:customField internalId="CUSTRECORD_CONTACT_TYPE">
                              <setupCustom:label>Contact type</setupCustom:label>
                              <setupCustom:fieldType>_listRecord</setupCustom:fieldType>
                              <setupCustom:selectRecordType internalId="16"
    xmlns:platformCore="urn:core_2010_1.platform.webservices.netsuite.com">
                                 <platformCore:name>Customer contact type</platformCore:name>
                              </setupCustom:selectRecordType>
                              <setupCustom:storeValue>true</setupCustom:storeValue>
```

```
                              <setupCustom:showInList>false</setupCustom:showInList>
                              <setupCustom:globalSearch>false</setupCustom:globalSearch>
                              <setupCustom:isParent>false</setupCustom:isParent>
                              <setupCustom:displayType>_normal</setupCustom:displayType>
                              <setupCustom:isMandatory>true</setupCustom:isMandatory>
                              <setupCustom:checkSpelling>false</setupCustom:checkSpelling>
                              <setupCustom:defaultChecked>false</setupCustom:defaultChecked>
                              <setupCustom:isFormula>false</setupCustom:isFormula>
                              <setupCustom:recType>15</setupCustom:recType>
                              <setupCustom:roleRestrict>false</setupCustom:roleRestrict>
                              <setupCustom:accessLevel>_edit</setupCustom:accessLevel>
                              <setupCustom:searchLevel>_edit</setupCustom:searchLevel>
                           </setupCustom:customField>
                           <setupCustom:customField internalId="CUSTRECORD_IF_OTHER">
                              <setupCustom:label>Other contact type</setupCustom:label>
                              <setupCustom:fieldType>_freeFormText</setupCustom:fieldType>
                              <setupCustom:storeValue>true</setupCustom:storeValue>
                              <setupCustom:showInList>false</setupCustom:showInList>
                              <setupCustom:globalSearch>false</setupCustom:globalSearch>
                              <setupCustom:isParent>false</setupCustom:isParent>
                              <setupCustom:displayType>_normal</setupCustom:displayType>
                              <setupCustom:displayWidth>42</setupCustom:displayWidth>
                              <setupCustom:isMandatory>false</setupCustom:isMandatory>
                              <setupCustom:checkSpelling>false</setupCustom:checkSpelling>
                              <setupCustom:defaultChecked>false</setupCustom:defaultChecked>
                              <setupCustom:isFormula>false</setupCustom:isFormula>
                              <setupCustom:recType>15</setupCustom:recType>
                              <setupCustom:roleRestrict>false</setupCustom:roleRestrict>
                              <setupCustom:accessLevel>_edit</setupCustom:accessLevel>
                              <setupCustom:searchLevel>_edit</setupCustom:searchLevel>
                           </setupCustom:customField>

// custom field list continues ...

                     </setupCustom:fieldList>
                     <setupCustom:formsList>
                           <setupCustom:forms>
                              <setupCustom:formName>Standard Customer Satisfaction Survey
Form<setupCustom:formName>
                              <setupCustom:formPref>false</setupCustom:formPref>
                           </setupCustom:forms>
                        </setupCustom:formsList>
                        <setupCustom:parentsList>
                           <setupCustom:parents>
                              <setupCustom:childDescr>Customer</setupCustom:childDescr>
                           </setupCustom:parents>
                        </setupCustom:parentsList>
                     </record>
                  </readResponse>
               </readResponseList>
            </getListResponse>
```

getDataCenterUrls

The getDataCenterUrls operation supports dynamic discovery of the correct URLs for external client access to NetSuite. This discovery is required because NetSuite began hosting new customer accounts in multiple data centers as of September 2012. Each data center has a different domain, so the domain to be used for external client access depends upon the data center hosting each NetSuite account.

This operation supports discovery of the following domains:

- restDomain - https://rest.netsuite.com or https://rest.na1.netsuite.com (used for RESTlet access)
- systemDomain - https://system.netsuite.com or https://system.na1.netsuite.com
- webservicesDomain - https://webservices.netsuite.com or https://webservices.na1.netsuite.com

If you are building an integration for a single customer, you can use this operation to determine the correct URL for that particular customer. If you are building a partner application, you need to incorporate this operation into your application logic so that can dynamically determine the correct URL for NetSuite access across multiple NetSuite customers.

REST Service as Alternate Method for Dynamic Discovery of URL

The getDataCenterUrls operation is only available in the 2012.2 and later endpoints. However you do not need to upgrade your endpoint to support dynamic discovery of the correct Web services domain. NetSuite also provides a REST service that you can use.

- For general information about this service, see the help topic *Using the REST roles Service to Get User Accounts, Roles, and Domains*.
- For example code that uses this REST service, see:
  - Java REST Example for Web Services Domain Discovery
  - .NET REST Example for Web Services Domain Discovery

  Note that NetSuite Developer Resources sample applications also incorporate RESTful code for dynamic discovery of Web services domain. These samples are available at: http://www.netsuite.com/portal/developers/resources/suitetalk-sample-applications.shtml.

Request

The GetDataCenterUrlsRequest type is used for this request. It requires the following field:

Element Name	XSD Type	Notes
account	string	NetSuite account ID. **Note:** If an invalid account ID is specified, the default URLs are returned.

Authenticated requests contain the following fields in addition to account:

Element Name	XSD Type	Notes
email	string	Email used to log in to NetSuite.
password	string	Password used to log in to NetSuite.
role	RecordRef	References an existing role for the user.

Response

The GetDataCenterUrlsResult type is used for the response. It contains the following fields:

Element Name	XSD Type	Notes
status	Status	The status for this operation. All applicable errors or warnings are listed within this type.
restDomain	string	URL used by external client applications to access NetSuite through a RESTlet.
systemDomain	string	
webservicesDomain	string	URL used by external client applications to access NetSuite through Web services.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- ExceededRequestSizeFault
- UnexpectedErrorFault

Also, a DATA_REQD error is thrown if no account ID is provided in the request. For an example, see SOAP Response (empty account).

Sample Code

Java Sample (public)

```
/**
 * Since 12.2 endpoint accounts are located in multiple datacenters with different domain names
.
 * In order to use the correct one, wrap the Locator and get the correct domain first.
 *
 * See getDataCenterUrls WSDL method documentation in the Help Center.
 */
private static class DataCenterAwareNetSuiteServiceLocator extends NetSuiteServiceLocator
{
```

```
 private String account;

 public DataCenterAwareNetSuiteServiceLocator(String account)
 {
  this.account = account;
 }

 @Override
 public NetSuitePortType getNetSuitePort(URL defaultWsDomainURL)
 {
  try
  {
   NetSuitePortType _port = super.getNetSuitePort(defaultWsDomainURL);
   // Get the webservices domain for your account
   DataCenterUrls urls = _port.getDataCenterUrls(account).getDataCenterUrls();
   String wsDomain = urls.getWebservicesDomain();

   // Return URL appropriate for the specific account
   return super.getNetSuitePort(new URL(wsDomain.concat(defaultWsDomainURL.getPath())));
  }
  catch (Exception e)
  {
   throw new RuntimeException(e);
  }
 }
}

---
// Locate the NetSuite web service.
NetSuiteServiceLocator service = new DataCenterAwareNetSuiteServiceLocator("myAccount");
// Get the service port (to the correct datacenter)
_port = service.getNetSuitePort(new URL("https://webservices.netsuite.com/services/NetSuitePort
_2012_2"));
```

Java Sample (authenticated)

```
public void getDataCenterUrlsAuthenticated() throws Exception
 {
  this.login();

  String account = "1326288";
  System.out.println("Account: " + account);
  DataCenterUrls urls = _port.getDataCenterUrls(account).getDataCenterUrls();
  System.out.println("\tREST domain: " + urls.getRestDomain());
  System.out.println("\tWeb Services domain: " + urls.getWebservicesDomain());
  System.out.println("\tSystem domain: " + urls.getSystemDomain());
  System.out.println();
 }
```

Java Sample (public)

```
public void testDataCenterUrlsPublic() throws Exception
{
    NLWsClient c = new NLWsClient();
    c.initPort();
    String account = "1326288";
```

```
    System.out.println("Account: " + account);
    DataCenterUrls urls = c.getPort().getDataCenterUrls(account).getDataCenterUrls();


    System.out.println("\tREST domain: " + urls.getRestDomain());
    System.out.println("\tWeb Services domain: " + urls.getWebservicesDomain());
    System.out.println("\tSystem domain: " + urls.getSystemDomain());
    System.out.println();
}
```

Java Sample (authenticated)

```
public void testDataCenterUrlsAuthenticated() throws Exception
{
    NLWsClient c = new NLWsClient(new String[]{"happygilmore@abc.com","foobar1","1326288","3"},
443,"https://webservices.netsuite.com"});
    c.useRequestLevelCredentials();
    String account = "1326288";


    System.out.println("Account: " + account);
    DataCenterUrls urls = c.getPort().getDataCenterUrls(account).getDataCenterUrls();


    System.out.println("\tREST domain: " + urls.getRestDomain());
    System.out.println("\tWeb Services domain: " + urls.getWebservicesDomain());
    System.out.println("\tSystem domain: " + urls.getSystemDomain());
    System.out.println();
}
```

SOAP Request (public)

```
<?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>
         <getDataCenterUrls xmlns="urn:messages_2012_2.platform.webservices.netsuite.com">
            <account>1326288</account>
         </getDataCenterUrls>
      </soapenv:Body>
   </soapenv:Envelope>
```

SOAP Response (public)

```
<?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:Header>
         <platformMsgs:documentInfo xmlns:platformMsgs="urn:messages_2012_2.platform.webservice
s.netsuite.com">
            <platformMsgs:nsId>WEBSERVICES_1326288_08162012790521873260264554_10fa27f165429</pl
atformMsgs:nsId>
```

```
        </platformMsgs:documentInfo>
      </soapenv:Header>
      <soapenv:Body>
         <getDataCenterUrlsResponse xmlns="urn:messages_2012_2.platform.webservices.netsuite.co
m">
            <getDataCenterUrlsReturn xsi:type="GetDataCenterUrlsResult" xmlns:platformCore="urn
:core_2012_2.platform.webservices.netsuite.com">
               <platformCore:status isSuccess="true"/>
               <platformCore:dataCenterUrls>
                  <platformCore:restDomain>https://rest.netsuite.com</platformCore:restDomain>
                  <platformCore:webservicesDomain>https://webservices.netsuite.com</platformCor
e:webservicesDomain>
                  <platformCore:systemDomain>https://system.netsuite.com</platformCore:systemDo
main>
               </platformCore:dataCenterUrls>
            </getDataCenterUrlsReturn>
         </getDataCenterUrlsResponse>
      </soapenv:Body>
   </soapenv:Envelope>
```

SOAP Request (authenticated)

```
<?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:Header>
         <ns1:passport soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next" soapenv:mustU
nderstand="0" xmlns:ns1="urn:messages_2012_2.platform.webservices.netsuite.com">
            <ns2:email xmlns:ns2="urn:core_2012_2.platform.webservices.netsuite.com">happygilmo
re@abc.com</ns2:email>
            <ns3:password xmlns:ns3="urn:core_2012_2.platform.webservices.netsuite.com">foobar1
</ns3:password>
            <ns4:account xmlns:ns4="urn:core_2012_2.platform.webservices.netsuite.com">1326288<
/ns4:account>
            <ns5:role internalId="3" xmlns:ns5="urn:core_2012_2.platform.webservices.netsuite.c
om"/>
         </ns1:passport>
      </soapenv:Header>
      <soapenv:Body>
         <getDataCenterUrls xmlns="urn:messages_2012_2.platform.webservices.netsuite.com">
            <account>1326288</account>
         </getDataCenterUrls>
      </soapenv:Body>
   </soapenv:Envelope>
```

SOAP Response (authenticated)

```
<?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:Header>
         <platformMsgs:documentInfo xmlns:platformMsgs="urn:messages_2012_2.platform.webservice
s.netsuite.com">
            <platformMsgs:nsId>WEBSERVICES_1326288_08162012790521873260264554_10fa27f165429</pl
atformMsgs:nsId>
         </platformMsgs:documentInfo>
```

```
      </soapenv:Header>
      <soapenv:Body>
         <getDataCenterUrlsResponse xmlns="urn:messages_2012_2.platform.webservices.netsuite.co
m">
            <getDataCenterUrlsReturn xsi:type="GetDataCenterUrlsResult" xmlns:platformCore="urn
:core_2012_2.platform.webservices.netsuite.com">
               <platformCore:status isSuccess="true"/>
               <platformCore:dataCenterUrls>
                  <platformCore:restDomain>https://rest.netsuite.com</platformCore:restDomain>
                  <platformCore:webservicesDomain>https://webservices.netsuite.com</platformCor
e:webservicesDomain>
                  <platformCore:systemDomain>https://system.netsuite.com</platformCore:systemDo
main>
               </platformCore:dataCenterUrls>
            </getDataCenterUrlsReturn>
         </getDataCenterUrlsResponse>
      </soapenv:Body>
   </soapenv:Envelope>
```

SOAP Response (empty account)

```
<?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:Header>
         <platformMsgs:documentInfo xmlns:platformMsgs="urn:messages_2012_2.platform.webservice
s.netsuite.com">
            <platformMsgs:nsId>WEBSERVICES__0911201212589142971204370270_d03cd34b349479541b</pl
atformMsgs:nsId>
         </platformMsgs:documentInfo>
      </soapenv:Header>
      <soapenv:Body>
         <getDataCenterUrlsResponse xmlns="urn:messages_2012_2.platform.webservices.netsuite.co
m">
            <getDataCenterUrlsReturn xsi:type="GetDataCenterUrlsResult" xmlns:platformCore="urn
:core_2012_2.platform.webservices.netsuite.com">
               <platformCore:status isSuccess="false">
                  <platformCore:statusDetail type="ERROR">
                     <platformCore:code>DATA_REQD</platformCore:code>
                     <platformCore:message>You need to provide a proper value for the required
field: account.</platformCore:message>
                  </platformCore:statusDetail>
               </platformCore:status>
            </getDataCenterUrlsReturn>
         </getDataCenterUrlsResponse>
      </soapenv:Body>
   </soapenv:Envelope>
```

getDeleted

The getDeleted operation is used to retrieve a list of deleted records of a given type during a specified period. The response does not differentiate between which records were deleted via Web services versus the UI. This operation is useful in order to easily synchronize information in a client application to the data currently in NetSuite.

Important: In order for a role to use the getDeleted operation, the role has to have both the Deleted Records and the Web Services permissions. The only exceptions are the Administrators and Full Access roles, which automtatically have all permissions. To give the Deleted Records permission to a role, go to Setup > Users/Roles > Manage Roles. Open the role for editing. On the Permissions > Setup subtab, add the Deleted Records permission. Note that users who have this permission can access results about any type of record that was deleted, even if they do not have permission to create or modify that record type. They also have permission to use the Deleted Record search type in the user interface, unless the Web Services Only option has been selected for the role.

For example, an Outlook client application plugin maintains a list of contacts and synchronizes that list with NetSuite. The getDeleted operation can be used to determine contact deletions since the last synchronization with NetSuite.

Deleted records in NetSuite are tracked by internal ID and record type. In cases where internal IDs are not unique within a record type, the deletions are not tracked and therefore can not be retrieved using the getDeleted operation. For a complete list of supported record types for this operation, refer to the DeletedRecordType enumeration in the coreTypes xsd. Entity records are unique in that you may have an entity that belongs to several different subtypes. For example, you may have a partner that is also a contact. If the partner record is deleted, the contact record remains intact.

The getDeleted operation returns a maximum of 1,000 records. If there are more than 1,000 records in the time interval you specify using the deletedDate filter, the ExceededRecordCountFault will be thrown.

Data about deleted records remains available indefinitely.

Request

The GetDeletedRequest type is used for the request.

Element Name	XSD Type	Notes
getDeletedFilter	GetDeletedFilter	See GetDeletedFilter.

GetDeletedFilter

The GetDeletedFilter allows you to retrieve data based on the date or record type of the deleted record.

Element Name	XSD Type	Notes
deletedDate	SearchDateField	A search filter with all the available search operations. For example, search for records that were deleted since a given dateTime or between two dateTime criteria.

Element Name	XSD Type	Notes
type	SearchEnumMultiSelectField	The type of the record that was deleted. For a complete list of all of the values currently supported by the getDeleted operation, refer to the DeletedRecordType enumerations in the coreTypes XSD.

Response

The DeletedRecord type is used for the response.

Element Name	XSD Type	Notes
deletedDate	dateTime	The status for this operation. All applicable errors or warnings are listed within this type.
record	BaseRef	Returns a list of available items.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

C#

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

            GetDeletedFilter fil = new GetDeletedFilter();

            SearchEnumMultiSelectField type = new SearchEnumMultiSelectField();
            type.@operator = SearchEnumMultiSelectFieldOperator.anyOf;
            type.operatorSpecified = true;
            String[] searchStringArray = new String[1];
            searchStringArray[0] = "_contact";
            type.searchValue = searchStringArray;
```

```
            DateTime searchDate = new DateTime();

            searchDate = DateTime.Now;
            searchDate = DateTime.Parse(searchDate.ToString("dd/MM/yyyy"));

            SearchDateField srchdtfldSearchDate = new SearchDateField();
            srchdtfldSearchDate.searchValueSpecified = true;
            srchdtfldSearchDate.searchValue = searchDate;
            srchdtfldSearchDate.operatorSpecified = true;
            srchdtfldSearchDate.@operator = SearchDateFieldOperator.on;

            fil.type = type;
            fil.deletedDate = srchdtfldSearchDate;

            GetDeletedResult response = _service.getDeleted(fil);
        }
```

Java

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

        SearchDateField searchDate = new SearchDateField();
        searchDate.setOperator(SearchDateFieldOperator.within);
        searchDate.setPredefinedSearchValue(SearchDate.thisBusinessWeek);

        String[] typeArray = new String[1];
        typeArray[0] = "_salesOrder";

        SearchEnumMultiSelectField type = new SearchEnumMultiSelectField();
        type.setOperator(SearchEnumMultiSelectFieldOperator.anyOf);
        type.setSearchValue(typeArray);

        GetDeletedFilter getDeletedFilter = new GetDeletedFilter();
        getDeletedFilter.setDeletedDate(searchDate);
        getDeletedFilter.setType(type);

        GetDeletedResult result = _port.getDeleted(getDeletedFilter);
}
```

PHP

```
// Include NetSuite Class Library
require_once '../PHPToolkit/NetSuiteService.php';

// Instantiate new service
$service = new NetSuiteService();

// Create request and filter
$getDeletedRequest = new GetDeletedRequest();
$getDeletedFilter = new GetDeletedFilter();

// Date to search on
$myDate = '2010-08-26T00:00:00.000Z';

// Use setFields to populate the filter
```

```
setFields($getDeletedFilter,array("deletedDate" => array( "operator" => "on","searchValue" => $
myDate)));

// Set the filter on the request
$getDeletedRequest->getDeletedFilter = $getDeletedFilter;

// Perform getCustomization operation
$getDeletedResponse = $service->getDeleted($getDeletedRequest);

// Display response
print_r($getDeletedResponse);
```

SOAP Request

```
<soapenv:Body>
<platformMsgs:getDeleted xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="h
ttp://www.w3.org/2001/XMLSchema-instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:pla
tformCoreTyp="urn:types.core_2_5.platform.webservices.netsuite.com" xmlns:platformCore="urn:cor
e_2_5.platform.webservices.netsuite.com" xmlns:platformMsgs="urn:messages_2_5.platform.webservi
ces.netsuite.com">
 <platformMsgs:getDeletedFilter>
  <platformCore:deletedDate>
   <platformCore:predefinedSearchValue>lastBusinessWeek
   </platformCore:predefinedSearchValue>
  </platformCore:deletedDate>
  <platformCore:type>
   <platformCore:searchValue>contact</platformCore:searchValue>
  </platformCore:type>
 </platformMsgs:getDeletedFilter>
</platformMsgs:getDeleted>
</soapenv:Body>
```

getItemAvailability

The getItemAvailability operation can be used to retrieve the inventory availability for a given list of items.

You can filter the returned list using a lastQtyAvailableChange filter. If set, only items with quantity available changes recorded as of this date are returned.

If the Multi-Location Inventory feature is enabled, this operation returns results for all locations. For locations that do not have any items available, only location IDs and names are listed in results.

Warning: This operation supports up to 10,000 records. If this limit is exceeded, an error is returned.

Request

The GetItemAvailabilityRequest type is used for the request.

Element Name	XSD Type	Notes
itemAvailabilityFilter	ItemAvailabilityFilter	You can filter the returned itemAvailability using this filter.

ItemAvailabilityFilter

Element Name	XSD Type	Notes
item	RecordRefList	References an exiting item record in NetSuite.
lastQtyAvailableChange	dateTime	If set, only items with quantity available changes recorded as of the specified date are returned.

Response

The GetItemAvailabilityResult type is used for the response.

Element Name	XSD Type	Notes
status	Status	The status for this operation. All applicable errors or warnings are listed within this type.
itemAvailabilityList	List	Returns a list of available items.

ItemAvailabilityList

Element Name	XSD Type	Notes
item	RecordRef	References an existing item record.
lastQtyAvailableChange	dateTime	If set, only items with quantity available changes recorded as of this date are returned.
locationId	RecordRef	References a location in a user defined list at Lists > Accounting > Locations.
quantityOnHand	double	The number of units of an item in stock.
onHandValueMli	double	
reorderPoint	double	The stock level at which a new order for the item needs to be placed
preferredStockLevel	double	The preferred quantity of this item maintained in inventory for a specific location.
quantityOnOrder	double	The number of units of an item pending receipt from a vendor.
quantityCommitted	double	The number of units of an item reserved by unfulfilled sales orders.
quantityBackOrdered	double	The number of units of an item reserved by unfulfilled sales orders.
quantityAvailable	double	The number of units in stock that have not been committed to fulfill sales.

ItemAvailabilityFilter

Element Name	XSD Type	Notes
item	RecordRefList	
lastQtyAvailableChange	dateTime	

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

SOAP Request (for C# sample)

```
<soap:Body>
<platformMsgs:getItemAvailability>
 <platformMsgs:itemAvailabilityFilter>
  <platformCore:item>
   <platformCore:recordRef internalId="390" type="inventoryItem">
    <platformCore:name/>
   </platformCore:recordRef>
  </platformCore:item>
  <platformCore:lastQtyAvailableChange/>
 </platformMsgs:itemAvailabilityFilter>
</platformMsgs:getItemAvailability>
</soap:Body>
```

SOAP Response (for C# sample)

```
<soapenv:Body>
<getItemAvailabilityResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
 <getItemAvailabilityResult
  xmlns="urn:core_2_5.platform.webservices.netsuite.com">
  <status isSuccess="true"/>
```

```
  <itemAvailabilityList>
   <itemAvailability>
    <item internalId="390" type="inventoryItem">
     <name>testItem</name>
    </item>
    <locationId internalId="1" type="location">
     <name>East Coast</name>
    </locationId>
    <quantityOnHand>20.0</quantityOnHand>
    <onHandValueMli>0.0</onHandValueMli>
    <quantityCommitted>0.0</quantityCommitted>
    <quantityAvailable>20.0</quantityAvailable>
   </itemAvailability>
   <itemAvailability>
    <item internalId="390" type="inventoryItem">
     <name>testItem</name>
    </item>
    <locationId internalId="2" type="location">
     <name>West Coast</name>
    </locationId>
   </itemAvailability>
  </itemAvailabilityList>
 </getItemAvailabilityResult>
</getItemAvailabilityResponse>
</soapenv:Body>
```


C# Sample

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

            RecordRef item1 = new RecordRef();
            item1.internalId = "59";
            item1.type = RecordType.inventoryItem;
            item1.typeSpecified = true;

            RecordRef[] recordrefs = new RecordRef[1];
            recordrefs[0] = item1;

            ItemAvailabilityFilter filter = new ItemAvailabilityFilter();
            filter.item = recordrefs;
            GetItemAvailabilityResult res = _service.getItemAvailability(filter);

        }
```

Java Sample

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

                RecordRef item1 = new RecordRef();
                item1.setInternalId("25");
                item1.setType(RecordType.inventoryItem);

                RecordRef item2 = new RecordRef();
                item2.setInternalId("76");
```

```
                item2.setType(RecordType.giftCertificateItem);

                RecordRef[] recordRefArray = new RecordRef[2];
                recordRefArray[0] = item1;
                recordRefArray[1] = item2;

                RecordRefList itemRefList = new RecordRefList();
                itemRefList.setRecordRef(recordRefArray);

                ItemAvailabilityFilter itemAvailability = new ItemAvailabilityFilter();
                itemAvailability.setItem(itemRefList);

                GetItemAvailabilityResult response = _port.getItemAvailability(itemAvailability
);
        }
```

getList

The getList operation is used to retrieve a list of one or more records by providing the unique ids that identify those records.

If there are multiple ids provided, they can either belong to the same record type or different record types. For example, it is possible to retrieve a customer and a contact within a single request using this operation.

If some of the provided ids are invalid, the request is still processed for the valid ids and the response will contain a warning that indicates that some of the ids were invalid.

Note: An asynchronous equivalent is available for this operation, **asyncGetList**. For information about asynchronous request processing, see Synchronous Versus Asynchronous Request Processing.

Request

The getListRequest type is used for the request. It contains the following fields.

Element Name	XSD Type	Notes
recordRef	RecordRef	An array of recordRef objects that specify the ids of the records to be retrieved.

Response

The getListResponse type is used for the response. It contains the following fields.

Element Name	XSD Type	Notes
status	Status	The status for this operation. All applicable errors or warnings are listed within this type.

Element Name	XSD Type	Notes
recordList	Record[]	A list of records that correspond to the specified ids. The actual records returned need to be of a type that extends the abstract type Record.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

SOAP Request

In the following example, two records are retrieved — one customer record and one employee record. Note that you must provide the internal ID of the specify instance of the record and the record type for the getList.

```
<soap:Body>
<platformMsgs:getList>
 <platformMsgs:baseRef internalId="983" type="customer" xsi:type="platformCore:RecordRef"/>
 <platformMsgs:baseRef internalId="-5" type="employee" xsi:type="platformCore:RecordRef"/>
</platformMsgs:getList>
</soap:Body>
```

SOAP Response

```
<soapenv:Body>
<getListResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
<readResponseList xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
 <readResponse>
  <ns1:status isSuccess="true" xmlns:ns1="urn:core_2_5.platform.webservices.netsuite.com"/>
  <record internalId="983" xsi:type="ns2:Customer"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns2="urn:relationships_2_5.lists.webservices.netsuite.com">
   <ns2:entityId>Shutter Fly</ns2:entityId>
   <ns2:isInactive>false</ns2:isInactive>
   <ns2:companyName>Shutter Fly, Inc</ns2:companyName>
   .
   ...[more fields]
   .
```

```
   <ns2:customFieldList>
    <ns6:customField internalId="custentity_map"
    xsi:type="ns6:StringCustomFieldRef"
    xmlns:ns6="urn:core_2_5.platform.webservices.netsuite.com">
      <ns6:value>http://maps.google.com</ns6:value>
    </ns6:customField>
   </ns2:customFieldList>
  </record>
 </readResponse>
 <readResponse>
  <ns8:status isSuccess="true" xmlns:ns8="urn:core_2_5.platform.webservices.netsuite.com"/>
  <record internalId="-5" xsi:type="ns9:Employee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns9="urn:employees_2_5.lists.webservices.netsuite.com">
   <ns9:entityId>A Wolfe</ns9:entityId>
   <ns9:isInactive>false</ns9:isInactive>
   .
   ...[more fields]
   .
  </record>
 </readResponse>
</readResponseList>
</getListResponse>
 </soapenv:Body>
```

C#

```
private int getCustomerList()
{
 // This operation requires a valid session
 this.login( true );

 // Prompt for list of nsKeys and put in an array
 _out.write( "\nnsKeys for records to retrieved (separated by commas): " );
 String reqKeys = _out.readLn();
 string [] nsKeys = reqKeys.Split( new Char[] {','} );

 return getCustomerList( nsKeys, false );
}
```

Java

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

 // Prompt for list of nsKeys and put in an array
 _console
 .write("\nnsKeys for records to retrieved (separated by commas): ");
 String reqKeys = _console.readLn();
 String[] nsKeys = reqKeys.split(",");

 return getCustomerList(nsKeys, false);
}
```

getPostingTransactionSummary

The getPostingTransactionSummary operation allows you to retrieve a summary of the actual data that posted to the general ledger in an account. You can use available filters/fields to generate reports that are similar to what you see when you run financial reports such as a Trial Balance, Balance Sheet, or an Income Statement.

Note: For information about NetSuite financial reports and financial statements, see the following topics in the NetSuite Help Center:

- *Financial Reports*
- *Financial Statements Overview*

The getPostingTransactionSummary operation returns the fields defined in PostingTransactionSummary. You can query by any filter defined in PostingTransactionSummaryFilter and group the results by any field defined in PostingTransactionSummaryField.

The first call to the operation returns the first page, total number of hits (totalRecords), and the number of pages. You can then retrieve subsequent pages by giving the page number. NetSuite caches the results after the first call to getPostingTransactionSummary as subsequent pages are being retrieved. The cache is reset if the session expires, or if you make another call to this operation with a page index of 1.

Also note the following:

- This operation can only be executed in a role that has the Financial Statements permission assigned. To enable this permission for a role, a NetSuite administrator must go to Setup > User Roles > Manage Roles > click the Reports tab > select **Financial Statments** from the Permission drop-down > click Save.
- Because the same operation is used for every page, make sure the fields and filters supplied are the same for every page requested, otherwise an error will be thrown.
- To search for null, specify -1. If the return is null, the element will be skipped (not -1).
- A maximum of 10,000 expressions is supported. A request with more than this number of expressions returns an error.
- For very large reports (for example, you have chosen all the columns), the query will take a very long time on first request, but subsequent requests will be fast. Make sure your timeout limit is set high.

Important: The information in the following paragraph pertains to **NetSuite OneWorld** accounts **only**.

The amounts returned from getPostingTransactionSummary are in the currency of the **subsidiary**, not the parent. If you want the amounts in the currency of the parent, you must programmatically apply the appropriate exchange rate. To obtain exchange rates, you must call getConsolidatedExchangeRate, which reads data from NetSuite's Consolidated Exchange Rate table. In your code, you must then multiply amount values returned by getPostingTransactionSummary by the exchange rate values returned by getConsolidatedExchangeRate.

Although the NetSuite UI automatically consolidates all amounts, you must perform your own exchange rate calculations in Web services.

Note: You may want to do planning in your local currency, in which case there is no need for exchange rate conversions.

Request

The GetPostingTransactionSummaryRequest type is used for the request.

Element Name	XSD Type	Notes
fields	PostingTransactionSummaryField	Specify how you want your data grouped.
filters	PostingTransactionSummaryFilter	Specify your filtering criteria.
pageIndex	xsd:int	Specify the page to be returned.

PostingTransactionSummaryField

Element Name	XSD Type	Notes
period	RecordRef	Specify to group data by period.
account	RecordRef	Specify to group data by account.
parentItem	RecordRef	Specify to group data by parent item.
item	RecordRef	Specify to group data by item.
customer	RecordRef	Specify to group data by customer.
department	RecordRef	Specify to group data by department.
class	RecordRef	Specify to group data by class.
location	RecordRef	Specify to group data by location.
subsidiary	RecordRef	Specify to group data by subsidiary.

PostingTransactionSummaryFilter

Element Name	XSD Type	Notes
period	RecordRef	Filter your request by period.
account	RecordRef	Filter your request by account.
parentItem	RecordRef	Filter your request by parent item.
item	RecordRef	Filter your request by item.
customer	RecordRef	Filter your request by customer.
department	RecordRef	Filter your request by department.

Element Name	XSD Type	Notes
class	RecordRef	Filter your request by class.
location	RecordRef	Filter your request by location.
subsidiary	RecordRef	Filter your request by subsidiary.

Response

The GetPostingTransactionSummaryResult type is used for the response.

Element Name	XSD Type	Notes
status	Status	The status for this operation. All applicable errors or warnings are listed within this type.
totalRecords	xsd:int	The total number of records for this search. Depending on the pageSize value, some or all the records may be returned in this response.
pageSize	xsd:int	The page size for this search.
totalPages	xsd:int	The total number of pages that are part of this search.
pageIndex	xsd:int	The page index for the current set of results.
postingTransactionSummaryList	PostingTransactionSummaryList	Returns a list of available transaction summary results based on filters defined in your request.

PostingTransactionSummaryList

Element Name	XSD Type	Notes
postingTransactionSummary	PostingTransactionSummary	Returns a list of available transactions that are filtered by and grouped by values specified in PostingTransactionSummaryField and PostingTransactionSummaryFilter.

PostingTransactionSummary

Element Name	XSD Type	Notes
period	RecordRef	Returns a summary based on period.
account	RecordRef	Returns a summary based on account.
parentItem	RecordRef	Returns a summary based on parent item.
item	RecordRef	Returns a summary based on item.
customer	RecordRef	Returns a summary based on customer.
department	RecordRef	Returns a summary based on department.
class	RecordRef	Returns a summary based on class.
location	RecordRef	Returns a summary based on location.
subsidiary	RecordRef	Returns a summary based on subsidiary.

Element Name	XSD Type	Notes
amount	double	Important: This argument is required.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

Example 1

The sample provided shows how to get and print account values using the getPostingTransactionSummary operation.

Java

```
/* Make Record Ref out of an internalId */
 public static RecordRef mrr(String internalId)
 {
  RecordRef toRet = new RecordRef();
  toRet.setInternalId(internalId);
  return toRet;
 }

public void testPostingActivity() throws Exception
{
//c.setHttpPort(80);
c.login();
PostingTransactionSummaryField pagb = new PostingTransactionSummaryField();
pagb.setParentItem(Boolean.TRUE);

/*
pagb.set_class(Boolean.FALSE);
pagb.setItem(Boolean.FALSE);
pagb.setCustomer(Boolean.FALSE);
pagb.setLocation(Boolean.FALSE);
*/
```

```
PostingTransactionSummaryFilter paf = new PostingTransactionSummaryFilter();
//paf.setAccount(new RecordRefList(new RecordRef[]{mrr("5"),mrr("12")}));
long start = System.currentTimeMillis();
GetPostingTransactionSummaryResult res = c.getPort().getPostingTransactionSummary(pagb, paf, 1)
;

for (int i=0; i<10; i++)
{
if (res.getPostingTransactionSummaryList().getPostingTransactionSummary(i).getAccount() != null
)
System.out.println("Account:" + res.getPostingTransactionSummaryList().getPostingTransactionSum
mary(i).getAccount().getInternalId());
if (res.getPostingTransactionSummaryList().getPostingTransactionSummary(i).getPeriod() != null)

System.out.println("Period:" + res.getPostingTransactionSummaryList().getPostingTransactionSumm
ary(i).getPeriod().getInternalId());
if (res.getPostingTransactionSummaryList().getPostingTransactionSummary(i).getCustomer() != nul
l)
System.out.println("Customer:" + res.getPostingTransactionSummaryList().getPostingTransactionSu
mmary(i).getCustomer().getInternalId());
if (res.getPostingTransactionSummaryList().getPostingTransactionSummary(i).get_class() != null)

System.out.println("_class:" + res.getPostingTransactionSummaryList().getPostingTransactionSumm
ary(i).get_class().getInternalId());
if (res.getPostingTransactionSummaryList().getPostingTransactionSummary(i).getDepartment() != n
ull)
System.out.println("Department:" + res.getPostingTransactionSummaryList().getPostingTransaction
Summary(i).getDepartment().getInternalId());
if (res.getPostingTransactionSummaryList().getPostingTransactionSummary(i).getLocation() != nul
l)
System.out.println("Location:" + res.getPostingTransactionSummaryList().getPostingTransactionSu
mmary(i).getLocation().getInternalId());
if (res.getPostingTransactionSummaryList().getPostingTransactionSummary(i).getSubsidiary() != n
ull)
System.out.println("Subsidiary:" + res.getPostingTransactionSummaryList().getPostingTransaction
Summary(i).getSubsidiary().getInternalId());
if (res.getPostingTransactionSummaryList().getPostingTransactionSummary(i).getItem() != null)
System.out.println("Item:" + res.getPostingTransactionSummaryList().getPostingTransactionSummar
y(i).getItem().getInternalId());
System.out.println("Amount:" + res.getPostingTransactionSummaryList().getPostingTransactionSumm
ary(i).getAmount());
}
for (int i=res.getPageIndex(); i<res.getTotalPages(); i++)
{
System.err.println("Elapsed Time : " + (System.currentTimeMillis() - start));
res = c.getPort().getPostingTransactionSummary(pagb,paf,i+1);
}
System.err.println("Elapsed Time : " + (System.currentTimeMillis() - start));
GetPostingTransactionSummaryResult p = c.getPort().getPostingTransactionSummary(pagb, paf, res.
getTotalPages()+2);
assertFalse(p.getStatus().isIsSuccess());
}
```

Example 2

The following SOAP and Java show how to get the posting activity for posting period 109. The sample prints the unconsolidated and consolidated amounts in the parent subsidiary by child subsidiary and account – for example, if subsidiary **2** is in Euros and subsidiary **1**, the parent company, is in dollars, show the amount in Euros and the amount in dollars for that account in period 109.

SOAP Request

```
  <soapenv:Body>
         <getPostingTransactionSummary xmlns="urn:messages_2009_1.platform.webservices.netsuite
.com">
            <fields>
               <ns6:period xmlns:ns6="urn:core_2009_1.platform.webservices.netsuite.com">true</
ns6:period>
               <ns7:account xmlns:ns7="urn:core_2009_1.platform.webservices.netsuite.com">true<
/ns7:account>
               <ns8:subsidiary xmlns:ns8="urn:core_2009_1.platform.webservices.netsuite.com">tr
ue<
   ns8:subsidiary>
            </fields>
            <filters>
               <ns9:period xmlns:ns9="urn:core_2009_1.platform.webservices.netsuite.com">
                  <ns9:recordRef internalId="109"/>
               </ns9:period>
            </filters>
            <pageIndex>1</pageIndex>
         </getPostingTransactionSummary>
      </soapenv:Body>
```

SOAP Reponse

```
<soapenv:Body>
         <getPostingTransactionSummaryResponse
  xmlns="urn:messages_2009_1.platform.webservices.netsuite.com">
            <platformCore:getPostingTransactionSummaryResult
  xmlns:platformCore="urn:core_2009_1.platform.webservices.netsuite.com">
               <platformCore:status isSuccess="true"/>
               <platformCore:totalRecords>19</platformCore:totalRecords>
               <platformCore:pageSize>1000</platformCore:pageSize>
               <platformCore:totalPages>0</platformCore:totalPages>
               <platformCore:pageIndex>1</platformCore:pageIndex>
               <platformCore:postingTransactionSummaryList>
                  <platformCore:postingTransactionSummary>
                     <platformCore:period internalId="109"/>
                     <platformCore:account internalId="2"/>
                     <platformCore:subsidiary internalId="1"/>
                     <platformCore:amount>500.0</platformCore:amount>
                  </platformCore:postingTransactionSummary>
                  <platformCore:postingTransactionSummary>
                     <platformCore:period internalId="109"/>
                     <platformCore:account internalId="6"/>
                     <platformCore:subsidiary internalId="1"/>
                     <platformCore:amount>19.0</platformCore:amount>
                  </platformCore:postingTransactionSummary>
                  <platformCore:postingTransactionSummary>
                     <platformCore:period internalId="109"/>
                     <platformCore:account internalId="6"/>
                     <platformCore:subsidiary internalId="2"/>
                     <platformCore:amount>5.0</platformCore:amount>
                  </platformCore:postingTransactionSummary>
                  <platformCore:postingTransactionSummary>
```

```
                <platformCore:period internalId="109"/>
                <platformCore:account internalId="6"/>
                <platformCore:subsidiary internalId="4"/>
                <platformCore:amount>6.0</platformCore:amount>
             </platformCore:postingTransactionSummary>
             <platformCore:postingTransactionSummary>
                <platformCore:period internalId="109"/>
                <platformCore:account internalId="6"/>
                <platformCore:subsidiary internalId="8"/>
                <platformCore:amount>-3.0</platformCore:amount>
             </platformCore:postingTransactionSummary>
             <platformCore:postingTransactionSummary>
                <platformCore:period internalId="109"/>
                <platformCore:account internalId="16"/>
                <platformCore:subsidiary internalId="6"/>
                <platformCore:amount>-4.0</platformCore:amount>
             </platformCore:postingTransactionSummary>
             <platformCore:postingTransactionSummary>
                <platformCore:period internalId="109"/>
                <platformCore:account internalId="24"/>
                <platformCore:subsidiary internalId="8"/>
                <platformCore:amount>-100.0</platformCore:amount>
             </platformCore:postingTransactionSummary>
             <platformCore:postingTransactionSummary>
                <platformCore:period internalId="109"/>
                <platformCore:account internalId="58"/>
                <platformCore:subsidiary internalId="8"/>
                <platformCore:amount>100.0</platformCore:amount>
             </platformCore:postingTransactionSummary>
             <platformCore:postingTransactionSummary>
                <platformCore:period internalId="109"/>
                <platformCore:account internalId="112"/>
                <platformCore:subsidiary internalId="1"/>
                <platformCore:amount>-496.0</platformCore:amount>
             </platformCore:postingTransactionSummary>
             <platformCore:postingTransactionSummary>
                <platformCore:period internalId="109"/>
                <platformCore:account internalId="112"/>
                <platformCore:subsidiary internalId="5"/>
                <platformCore:amount>-1.0</platformCore:amount>
             </platformCore:postingTransactionSummary>
             <platformCore:postingTransactionSummary>
                <platformCore:period internalId="109"/>
                <platformCore:account internalId="139"/>
                <platformCore:subsidiary internalId="4"/>
                <platformCore:amount>-9.0</platformCore:amount>
             </platformCore:postingTransactionSummary>
             <platformCore:postingTransactionSummary>
                <platformCore:period internalId="109"/>
                <platformCore:account internalId="142"/>
                <platformCore:subsidiary internalId="8"/>
                <platformCore:amount>3.0</platformCore:amount>
             </platformCore:postingTransactionSummary>
             <platformCore:postingTransactionSummary>
                <platformCore:period internalId="109"/>
                <platformCore:account internalId="145"/>
                <platformCore:subsidiary internalId="5"/>
                <platformCore:amount>1.0</platformCore:amount>
             </platformCore:postingTransactionSummary>
             <platformCore:postingTransactionSummary>
```

```
                    <platformCore:period internalId="109"/>
                    <platformCore:account internalId="148"/>
                    <platformCore:subsidiary internalId="2"/>
                    <platformCore:amount>-5.0</platformCore:amount>
                 </platformCore:postingTransactionSummary>
                 <platformCore:postingTransactionSummary>
                    <platformCore:period internalId="109"/>
                    <platformCore:account internalId="153"/>
                    <platformCore:subsidiary internalId="6"/>
                    <platformCore:amount>4.0</platformCore:amount>
                 </platformCore:postingTransactionSummary>
                 <platformCore:postingTransactionSummary>
                    <platformCore:period internalId="109"/>
                    <platformCore:account internalId="155"/>
                    <platformCore:subsidiary internalId="1"/>
                    <platformCore:amount>-23.0</platformCore:amount>
                 </platformCore:postingTransactionSummary>
                 <platformCore:postingTransactionSummary>
                    <platformCore:period internalId="109"/>
                    <platformCore:account internalId="175"/>
                    <platformCore:subsidiary internalId="4"/>
                    <platformCore:amount>3.0</platformCore:amount>
                 </platformCore:postingTransactionSummary>
                 <platformCore:postingTransactionSummary>
                    <platformCore:period internalId="109"/>
                    <platformCore:account internalId="186"/>
                    <platformCore:subsidiary internalId="8"/>
                    <platformCore:amount>0.0</platformCore:amount>
                 </platformCore:postingTransactionSummary>
                 <platformCore:postingTransactionSummary>
                    <platformCore:period internalId="109"/>
                    <platformCore:account internalId="187"/>
                    <platformCore:subsidiary internalId="8"/>
                    <platformCore:amount>0.0</platformCore:amount>
                 </platformCore:postingTransactionSummary>
              </platformCore:postingTransactionSummaryList>
           </platformCore:getPostingTransactionSummaryResult>
        </getPostingTransactionSummaryResponse>
     </soapenv:Body>
```

Java

```
/* Make Record Ref out of an internalId */
 public static RecordRef mrr(String internalId)
 {
  RecordRef toRet = new RecordRef();
  toRet.setInternalId(internalId);
  return toRet;
 }

 public void testPostingWorkflow() throws Exception
 {
  c.setCredentials(CRED_DB96_SIC);
  c.useRequestLevelCredentials();

  // Show and group by subsidiary, period and account.
  // These are the very basic columns. If you don't include account,
  // all amounts will be 0.
```

```
  PostingTransactionSummaryField pagb = new PostingTransactionSummaryField();
  pagb.setSubsidiary(Boolean.TRUE);
  pagb.setPeriod(Boolean.TRUE);
  pagb.setAccount(Boolean.TRUE);

  PostingTransactionSummaryFilter paf = new PostingTransactionSummaryFilter();
  paf.setPeriod(new RecordRefList(new RecordRef[]{mrr("109")}));
  GetPostingTransactionSummaryResult gestalt =
   c.getPort().getPostingTransactionSummary(pagb,paf,1);
  ConsolidatedExchangeRateFilter f = new ConsolidatedExchangeRateFilter();
  f.setPeriod(mrr("109"));
  f.setToSubsidiary(mrr("1"));
  GetConsolidatedExchangeRateResult cerr = c.getPort().getConsolidatedExchangeRate(f);

  for (PostingTransactionSummary unConsolidated :
   gestalt.getPostingTransactionSummaryList().getPostingTransactionSummary())
  {
   ConsolidatedExchangeRate rate = null;
   for (ConsolidatedExchangeRate testRate :
    cerr.getConsolidatedExchangeRateList().getConsolidatedExchangeRate())
   {
    if
     (testRate.getFromSubsidiary().getInternalId().equals
     (unConsolidated.getSubsidiary().getInternalId()))
    {
     rate = testRate;
     break;
    }
   }
   if (rate == null)
    continue;  // the Target Subsidiary will not have a conversion rate to itself.
   System.out.println("\n\nFor Subsidiary: " + unConsolidated.getSubsidiary().getInternalId()
+
    " to Consolidated Parent (1).\nPeriod: " + unConsolidated.getPeriod().getInternalId() +
      "\nAccount Id: " + unConsolidated.getAccount().getInternalId() +
      "\nUnconsolidated Amount is: " + unConsolidated.getAmount() +
      "\nConsolidated Amounts are (Avg/Historical/Current): "
         + (unConsolidated.getAmount()*rate.getAverageRate()) + " / " +
       (unConsolidated.getAmount()*rate.getHistoricalRate()) + " / " +
       (unConsolidated.getAmount()*rate.getCurrentRate()));
  }
 }
```

getSavedSearch

This operation allows users to retrieve a list of existing saved search IDs on a per-record-type basis (for example, all saved search IDs for every Customer saved search). Note that once you retrieve the list of saved search IDs, you may need to look in the NetSuite UI to see the criteria defined for the saved search. To navigate to the list of saved searches in the NetSuite UI, go to Lists > Search > Saved Searches.

This API takes a search record type as a request argument and returns a list of record references of the saved search. For each saved search, the response includes the internalId, scriptId, and name (if you are using the 2013.2 WSDL or later — with previous WSDLs, the scriptId is not returned).

For use cases explaining why you would want to get a list of saved search IDs and then reference a specific ID in your code, see Reference Existing Saved Searches.

Note: There is no async equivalent for this operation.

Request

The GetSavedSearchRequest type is used for the request. It contains the following fields.

Element Name	XSD Type	Notes
record	Record	Contains an array of record objects. The record type is an abstract type so an instance of a type that extends record must be used—such as Customer or Event.

Response

The GetSavedSearchResponse type is used for the response. It contains the following fields.

Element Name	XSD Type	Notes
status	Status	The status for this operation. All applicable errors or warnings are listed within this type.
totalRecords	xsd:int	The total number of records for this search. Depending on the pageSize value, some or all the records may be returned in this response
recordRefList	Record[]	A list of records that correspond to the specified ids. The actual records returned need to be of a type that extends the abstract type Record.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

C#

```
GetSavedSearchRecord record = new GetSavedSearchRecord();
record.searchTypeSpecified = true;
record.searchType = SearchRecordType.transaction;

_service.getSavedSearch(record);
;
```

Java

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

  GetSavedSearchRecord record = new GetSavedSearchRecord();
  record.setSearchType(SearchRecordType.transaction);

  GetSavedSearchResult result = _port.getSavedSearch(record);
}
```

SOAP Request

```
   <soap:Body>
        <getSavedSearch xmlns="urn:messages_2013_2.platform.webservices.netsuite.com">
            <record searchType="transaction"/>
        </getSavedSearch>
    </soap:Body>
```

SOAP Response

```
<soapenv:Body>
 <getSavedSearchResponse xmlns="urn:messages_2013_2.platform.webservices.netsuite.com">
  <platformCore:getSavedSearchResult xmlns:platformCore="urn:core_2013_2.platform.webservices.n
etsuite.com">
   <platformCore:status isSuccess="true"/>
      <platformCore:totalRecords>5</platformCore:totalRecords>
      <platformCore:recordRefList>
       <platformCore:recordRef xsi:type="platformCore:CustomizationRef" internalId="26" scriptI
d="customsearch26">
        <platformCore:name>Custom Transaction Search</platformCore:name>
        </platformCore:recordRef>
      <platformCore:recordRef xsi:type="platformCore:CustomizationRef" internalId="27" scriptId
="customsearch27">
     <platformCore:name>CC Transaction Search</platformCore:name>
    </platformCore:recordRef>
    <platformCore:recordRef xsi:type="platformCore:CustomizationRef" internalId="42" scriptId="
customsearch42">
          <platformCore:name>Returned Items</platformCore:name>
         </platformCore:recordRef>
         <platformCore:recordRef xsi:type="platformCore:CustomizationRef" internalId="44" scrip
tId="customsearch44">
```

```
     <platformCore:name>Ordered More than One Item</platformCore:name>
       </platformCore:recordRef>
    <platformCore:recordRef xsi:type="platformCore:CustomizationRef" internalId="46" scriptId="
customsearch46">
     <platformCore:name>Spending Account Average</platformCore:name>
    </platformCore:recordRef>
   </platformCore:recordRefList>
  </platformCore:getSavedSearchResult>
 </getSavedSearchResponse>
</soapenv:Body>
```

getSelectValue

Use the getSelectValue operation to retrieve valid select options for a given RecordRef, CustomRecordRef, or enumerated static field. This is useful if you are writing an application UI that needs to mimic NetSuite UI logic, if the referenced record type is not yet exposed in SuiteTalk, or when the logged-in user's role does not have permission to the instances of the referenced record type. A call to getSelectValue may return different results for the same field for different roles.

The getSelectValue operation can be used on standard body fields and custom body fields. It can also be used on sublist fields that appear on both standard and custom records.

These topics describe each aspect of the operation:

- GetSelectValue Overview
- Paginating Select Values
- Filtering Select Value Text
- Getting Dependent Select Values
- Sample Code

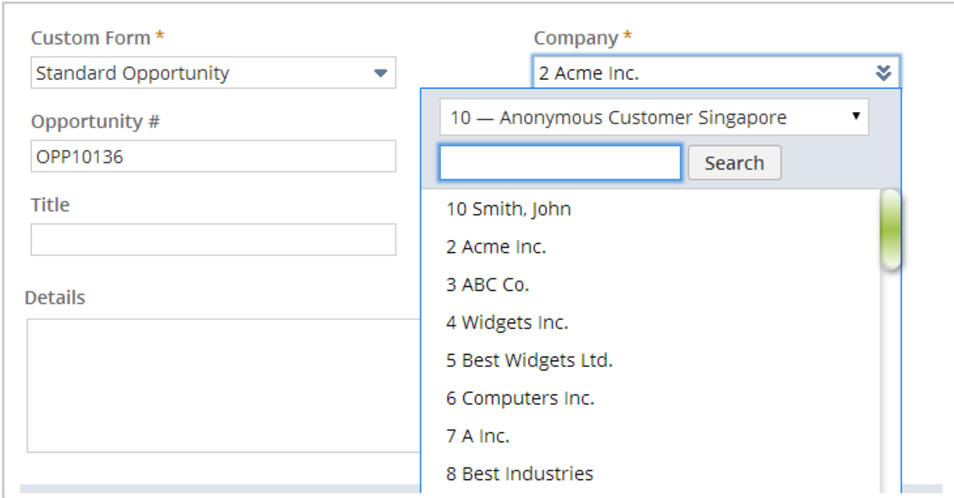
Important: If you reference a field or a select value that is renamed in future versions of NetSuite, your requests will still be handled, however, a warning will be returned. Also, when working with this operation be aware of any special permissions applied to a field. For example, a permission error will be thrown if you attempt to get select values on a field that has been disabled on a form.

Note: The getSelectValue operation will not return the following values: "" , (blank), -1,

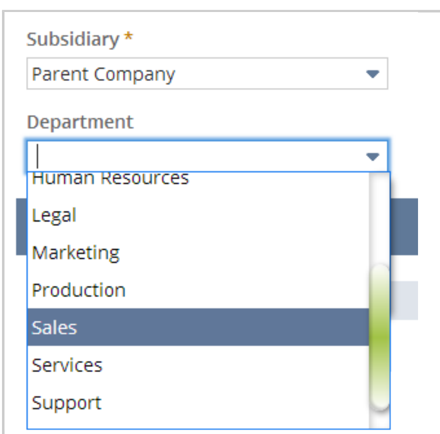
-New-, -2, -Custom-.

GetSelectValue Overview

The following figure shows a RecordRef field (Company) as displayed in the user interface, along with a list of select values for this field. You can use getSelectValue to return the entire list of values or just a subset of values.



In accounts where there are levels of field dependencies, such as OneWorld accounts, you can use getSelectValue to get select values for field "B" based on the value of field "A". The following screenshot shows an employee record in a OneWorld account. The select values associated with the Department field are based on the value specified in the Subsidiary field. In this scenario, you can use getSelectValue to get the values on the Department field by specifying the internal ID of its filterBy ("master") field, which is Subsidiary (internal ID —**subsidiary**). For more information, see Getting Dependent Select Values.



Running getSelectValue against records that are created from other records may result in an INSUFFICIENT_PERMISSION error.

Paginating Select Values

The first call to getSelectValue returns the total number of select values for the specified field. If you choose, you can return a subset of those values by specifying a pageIndex number in the request.

```
<complexType name="getSelectValueRequest">
       <sequence>
           <element name="fieldDescription" type="platformCore:GetSelectValueFieldDescription"

 minOccurs="1" maxOccurs="1"/>
   <element name=" pageIndex " type="xsd:int" minOccurs="1" maxOccurs="1"/>
 </sequence>
   </complexType>
```

Note: To define a page size, set the pageSize element in the SearchPreference type (see Setting Search Preferences). The value must be greater than 10 and less than the system-defined maximum of 1000. If the number of select values exceeds the page size, the remaining results must be retrieved in subsequent calls getSelectValue with a new pageIndex value.

Filtering Select Value Text

To help end users more easily pick from a long list of select values, you can use the **contains, startsWith**, or **is** search operators to filter the results returned by getSelectValue. The **filter** element in the GetSelectValueFieldDescription object allows you to set the operator type in your request.

```
<complexType name=" GetSelectValueFieldDescription ">
 <sequence>
    <element name="recordType" type="platformCoreTyp:RecordType" minOccurs="0" maxOccurs="1"/>
    <element name="customRecordType" type="platformCore:RecordRef" minOccurs="0"
  maxOccurs="1"/>
                <element name="sublist" type="xsd:string" minOccurs="0" maxOccurs="1"/>
       <element name="field" type="xsd:string" minOccurs="1" maxOccurs="1"/>
       <element name="customForm" type="platformCore:RecordRef" minOccurs="0" maxOccurs="1"/>
       <element name=" filter " type="platformCore: GetSelectValueFilter " minOccurs="0" maxOcc
urs="1"/>
       <element name="filterByValueList" type="platformCore:GetSelectFilterByFieldValueList"
  minOccurs="0" maxOccurs="1"/>
        </sequence>
</complexType>
```

Getting Dependent Select Values

The getSelectValue operation can also be used to get select values that are available on the condition of other field values. For example, in OneWorld accounts the values that appear in many dropdown fields are based on the values specified for either Customer or Subsidiary. As another example, on the Item sublist of a Sales Order, the values for the Tax Code field depend on both customer and item values.

In your getSelectValue call, you will use the **GetSelectFilterByFieldValueList** and **GetSelectFilterByFieldValue** objects to specify the filterBy field of a dependent field. (The dependent field is the field you want to get the values for. The filterBy field is the field that controls which values are available for the dependent field.)

```
<complexType name=" GetSelectFilterByFieldValueList ">
    <sequence>
    <element name="filterBy" type="platformCore:GetSelectFilterByFieldValue" minOccurs="1"
     maxOccurs="unbounded"/>
    </sequence>
 </complexType>
 <complexType name=" GetSelectFilterByFieldValue ">
    <sequence>
    <element name=" sublist " type="xsd:string" minOccurs="0" maxOccurs="1"/>
    <element name=" field " type="xsd:string" minOccurs="1" maxOccurs="1"/>
    <element name=" internalId " type="xsd:string" minOccurs="1" maxOccurs="1"/>
    </sequence>
 </complexType>
```

The sublist, field, and internalId arguments will contain:

- the name of the sublist the filterBy field appears on (*if* it appears on a sublist)
- the schema name of the filterBy field (for example, “entity”)
- the internalId of the specific entity record (for example, “87”)

A getSelectValue call for a dependent field with no filterBy (“master”) field specified returns 0 records. This type of call also returns a warning that notes the filterBy field, so the call can be corrected.

Important: Currently there is no programmatic way to discover what the filterBy field is for another field. Note, however, on transaction records the Customer (**entity**) field is always the filterBy field for any dependent RecordRef field. For all other record types, you must use the UI to see which field is the filterBy field.

Request

The getSelectValueRequest type is used for the request. It contains the following elements.

Element Name	XSD Type	Notes
fieldDescription	GetSelectValueFieldDescript	Use to specify all characteristics of the field containing the select values. For example, depending on the field and the values you want returned, you will specify the names or internalIds for the record type, sublist, and field. You may also

Element Name	XSD Type	Notes
		specify filtering criteria to refine the select options returned in the response.
pageIndex	xsd: int	For select values that span multiple pages, use this argument to specify which page to return in your response.

GetSelectValueFieldDescription

Element Name	XSD Type	Notes
recordType	RecordType	Specify a record defined in coreTypes.xsd
customRecordType	RecordRef	If you are getting select values for a field appearing on a custom record, specify the internal or external ID of the custom record, as well as the custom record type.
sublist	xsd:string	If getting select values for a field on a sublist, you must specify the sublist name (as it is defined in the schema). Sublist names are those names that appear toward the bottom of a record schema and are appended with **List**, for example itemList, saleTeamList, timeList.
field	xsd:string	Specify a field name, as defined in the schema for that record. The field value can represent either a body field or a sublist field.
customForm	RecordRef	If the RecordRef or CustomRecordRef field is associated with a custom form, specify the internal or external ID of the custom form, as well as the custom form type.
filter	GetSelectValueFilter	If you choose, you can filter select options using the **contains, is**, or **startsWith** operator. Use any of these operators to return a subset of the options associated with the RecordRef or CustomRecordRef field. For example, to get a specific list of customers on an opportunity record, you can search for “Adam” with the **contains** operator to get only customers whose name contains Adam. See also Filtering Select Value Text.
filterByValueList	GetSelectFilterByFieldValueList	This will contain a reference to the filterBy field (or fields) in which you specify the following: • **sublist** – If the filterBy field is on a sublist, you will specify the name of the sublist containing this field. Sublist names are appended with List (for example item **List**). • **field** – The name of the filterBy field (for example, *entity* or *subsidiary*). • **internalId** – The internalId value of the filterBy field (for example, “87” or “112”). If the filterBy field on an Opportunity record is Customer, you would specify: sublist = null field = “entity”

Element Name	XSD Type	Notes
		internalId = "87" (87 being the internalId of a specific customer record, such as the Abe Simpson customer record) See also Getting Dependent Select Values.

Response

The getSelectValueResponse type is used for the response. It contains the following elements.

Element Name	XSD Type	Notes
status	Status	The status for this operation. All applicable errors or warnings are listed within this type.
totalRecords	xsd:int	The total number of record references for this search. Depending on the pageSize value, some or all the references may be returned in the response.
totalPages	xsd:int	The total number of pages for this search. Depending on the pageSize value, some or all the pages may be returned in this response.
baseRefList	BaseRef[]	An array of baseRefs that references existing NetSuite records, including custom records. These baseRefs represent the valid values available for the current field.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.o
rg
```

```
soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001
XMLSchema-instance">
 <soapenv:Header>
  <ns1:passport soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next" soapenv:mustUndersta
nd="0"
     xmlns:ns1="urn:messages_2009_2.platform.webservices.netsuite.com">
  <ns2:email xmlns:ns2="urn:core_2009_2.platform.webservices.netsuite.com">kwolfe@netsuite.com<

    ns2:email>
 <ns3:password xmlns:ns3="urn:core_2009_2.platform.webservices.netsuite.com">mypassword<
 ns3:password>
<ns4:account xmlns:ns4="urn:core_2009_2.platform.webservices.netsuite.com">000071</ns4:account
<ns5:role internalId="3" xmlns:ns5="urn:core_2009_2.platform.webservices.netsuite.com"/>
  </ns1:passport>
 </soapenv:Header>
 <soapenv:Body>
  <getSelectValue xmlns="urn:messages_2009_2.platform.webservices.netsuite.com">
   <fieldDescription>
    <ns6:recordType xmlns:ns6="urn:core_2009_2.platform.webservices.netsuite.com">salesOrder<
       ns6:recordType>
<ns7:sublist xmlns:ns7="urn:core_2009_2.platform.webservices.netsuite.com">itemList</ns7:sublis
t>
<ns8:field xmlns:ns8="urn:core_2009_2.platform.webservices.netsuite.com">item</ns8:field>
<ns9:filterByValueList xmlns:ns9="urn:core_2009_2.platform.webservices.netsuite.com">
     <ns9:filterBy>
      <ns9:field>entity</ns9:field>
      <ns9:internalId>8</ns9:internalId>
     </ns9:filterBy>
    </ns9:filterByValueList>
   </fieldDescription>
   <pageIndex>1</pageIndex>
  </getSelectValue>
 </soapenv:Body>
</soapenv:Envelope>
```

Java

This sample shows how get select values for the Item field that appears on the Item sublist of a Sales Order record.

```
GetSelectFilterByFieldValueList myFilterByList = new GetSelectFilterByFieldValueList(new
 GetSelectFilterByFieldValue[]{new GetSelectFilterByFieldValue(null,"entity","8")});

GetSelectValueFieldDescription myGSVField = new GetSelectValueFieldDescription(RecordType.sales
Order,
         null, "itemList", "item", null, null, myFilterByList);

BaseRef[] br = c.getSelectValue(myGSVField);
```

getServerTime

This operation takes no arguments and returns the NetSuite server time in GMT, regardless of a user's time zone. Developers do not have to rely on client time when writing synchronization procedures because the client time may not be in synch with the NetSuite server time.

If you choose, you can write client code that takes the GMT returned by NetSuite and convert the time to your local time zone. The format of the dateTime value returned by getServerTime contains integer-valued year, month, day, hour and minute properties, a decimal-valued second property and a boolean time-zoned property (timestamp) – for example, 2005-09-21T15:24:00.000-07:00, where 2005-09-21 is the date, 15:24:00.000 is the time and -07:00 is your own time zone offset, if you choose to set one.

Any user with an active session can call getServerTime. There are no explicit permissions for this operation. For example, if you need to synchronize customer data with NetSuite, you can call getServerTime to initialize your synchronization process. To synchronize new or changed customers records, you can search customers and filter by **lastModifiedDate** using the value returned in getServerTime. To synchronize deleted customer records, you can call the getDeleted operation and use the value returned in getServerTime. This makes your synchronization process independent of your client time,which may not be in sync with the NetSuite server time and hence the timestamp we use to track changes.

Request

The GetServerTimeRequest type is used for the request. It takes no arguments.

Response

The GetServerTimeResponse type is used for the response. It contains the following elements.

Element Name	XSD Type	Notes
response	GetServerTimeResult	See GetServerTimeResult, below.

GetServerTimeResult

The GetServerTimeResult type contains the following elements.

Element Name	XSD Type	Notes
status	Status	The status for this request. All applicable errors or warnings will be listed within this type, which is defined in the platformCore XSD.
serverTime	dateTime	dateTime value returned is in GMT.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault

- ExceededUsageLimitFault
- UnexpectedErrorFault

Sample Code

Java

```
public void testGetServerTime() throws Exception
{
   c.login();

   GetServerTimeResult rs = c.getPort().getServerTime();
   System.out.println("Welcome to NetSuite. At the sound of the tone the NetSuite time will be
: " + new
   SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(rs.getServerTime().getTime()));

   System.out.println("This compares with a client time of " + new SimpleDateFormat("yyyy-MM-dd

   HH:mm:ss").format(Calendar.getInstance().getTime() ));
   System.out.println("This represents a skew of  "  + (Calendar.getInstance().getTimeInMillis(
)
 rs.getServerTime().getTimeInMillis() )/1000 + " seconds from netsuite
 (where positive time means the client clock is comparatively fast).");
}
```

initialize / initializeList

Use the initialize operation to emulate the UI workflow by prepopulating fields on transaction line items with values from a related record. Your Web services application can then modify only the values it needs to before submitting the record.

For example, in the UI clicking Bill from a Sales Order record loads an Invoice record where fields are populated with values from the Sales Order. When loading an invoice record in Web services, you can reference the related Sales Order record to initialize fields with values from that sales order.

Note: An asynchronous equivalent is available for the initializeList operation, **asyncInitializeList**. For information about asynchronous request processing, see Synchronous Versus Asynchronous Request Processing.

The following table lists all the transaction types that can be used with the initialize operation and the valid reference types they can use.

(Target Record) Transaction Type	Initialize Reference	Notes
Assembly Unbuild	Assembly Build	
Cash Refund	Cash Sale	
Cash Refund	Return Authorizat ion	

(Target Record) Transaction Type	Initialize Reference	Notes
Cash Sale	Customer	Populates Billable tabs
Cash Sale	Estimate	
Cash Sale	Opportunity	
Cash Sale	Sales Order	
Credit Memo	Customer	
Credit Memo	Invoice	
Credit Memo	Return Authorizat ion	
Customer Payment	Customer	Use the arAccount parameter to specify the accounts receivable (AR) account in a Customer to Customer Payment initialization. (This parameter is define in InitializeAuxRefType.)
Customer Payment	Invoice	
Customer Refund	Credit Memo	
Customer Refund	Customer	
Deposit Application	Customer Deposit	
Estimate	Opportunity	
Invoice	Customer	Defaults billable time, expense, and items
Invoice	Estimate	
Invoice	Opportunity	
Invoice	Sales Order	Defaults the line items as well as billable time, expense, and items
Item Fulfillment	Intercompany Transfer Order	
Item Fulfillment	Sales Order	Defaults the line items for fulfillment
Item Fulfillment	Transfer Order	
Item Fulfillment	Vendor Return Authorization	
Item Receipt	Intercompany Transfer Order	
Item Receipt	Purchase Order	Initializing an item receipt from a **dropship** purchase order is not supported. The recommended workflow is to initialize an item fulfillment from the sales order that has the dropship item. The accounting impact will be the same and more desirable because inventory levels will not be affected.
Item Receipt	Return Authorizat ion	
Item Receipt	Transfer Order	
Return Authorization	Cash Sale	
Return Authorization	Invoice	

(Target Record) Transaction Type	Initialize Reference	Notes
Return Authorization	Sales Order	
Sales Order	Estimate	
Sales Order	Opportunity	
Vendor Bill	Purchase Order	You can initialize or link a single purchase order to a vendor bill, or a list of purchase orders to a vendor bill. Note, however, when linking multiple purchase orders to a single vendor bill, all purchase orders must be for the same vendor. For more information, see the help topic *Linking Purchase Orders to a Vendor Bill*.
Vendor Bill	Vendor	Use the apAccount parameter to specify the accounts payable (AP) account in a Vendor to Vendor Bill initialization. (This parameter is define in InitializeAuxRefType.)
Vendor Credit	Vendor	
Vendor Credit	Vendor Bill	
Vendor Credit	Vendor Return Authorization	
Vendor Payment	Employee	
Vendor Payment	Vendor	
Vendor Payment	Vendor Bill	
Vendor Return Authorization	Vendor	
Vendor Return Authorization	Vendor Bill	
Work Order Close	Work Order	The work order must be configured to use WIP.
Work Order Completion	Work Order	The work order must be configured to use WIP.
Work Order Issue	Work Order	The work order must be configured to use WIP.

Ignore Read-Only Preference

In order to submit an initialized record without having to remove read-only fields populated during the initialization, set the Ignore Read-Only header preference to TRUE.

Important: When this preference is set to TRUE, read-only fields are simply ignored during the Web services request. The fields still cannot be set.

Request

The InitializeRequest type is used for this request. It contains the following fields:

Element Name	XSD Type	Notes
initializeRecord	InitializeRecord	

The InitializeRecord type takes the following fields:

- type
- reference
- auxReference
- referenceList

Response

The InitializeResponse type is used for the response. It contains the following fields:

Element Name	XSD Type	Notes
status	Status	The status for this operation. All applicable errors or warnings are listed within this type.
record	Record	The record type of the resulting initialization.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

SOAP Request (initialize)

```
<soapenv:Body>
<platformMsgs:initialize xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="h
ttp://www.w3.org/2001/XMLSchema-instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:pla
```

```
tformCoreTyp="urn:types.core_2_6.platform.webservices.netsuite.com" xmlns:platformCore="urn:cor
e_2_6.platform.webservices.netsuite.com" xmlns:platformMsgs="urn:messages_2_6.platform.webservi
ces.netsuite.com">
 <platformMsgs:initializeRecord>
  <platformCore:type>invoice</platformCore:type>
  <platformCore:reference internalId="1513" type="salesOrder">
   <platformCore:name>1511</platformCore:name>
  </platformCore:reference>
 </platformMsgs:initializeRecord>
</platformMsgs:initialize>
</soapenv:Body>
```

SOAP Request (initializeList)

```
<initializeList xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
  <initializeRecord>
    <ns1:type
   xmlns:ns1="urn:core_2_6.platform.webservices.netsuite.com">customerPayment
  </ns1:type>
    <ns2:reference internalId="176" type="customer"
   xmlns:ns2="urn:core_2_6.platform.webservices.netsuite.com"/>
  </initializeRecord>
  <initializeRecord>
    <ns3:type xmlns:ns3="urn:core_2_6.platform.webservices.netsuite.com">
   invoice</ns3:type>
        <ns4:reference internalId="176" type="customer"
     xmlns:ns4="urn:core_2_6.platform.webservices.netsuite.com"/>
   </initializeRecord>
</initializeList>
```

Java (initializeList)

```
InitializeRef iRef1 = new InitializeRef();
iRef1.setInternalId("176");
iRef1.setType(InitializeRefType.customer);

InitializeRecord ir1 = new InitializeRecord();
ir1.setReference(iRef1);
ir1.setType(InitializeType.customerPayment);

InitializeRecord ir2 = new InitializeRecord();
ir2.setReference(iRef1);
ir2.setType(InitializeType.invoice);

sessMgr.getPort().getNetSuitePortTypePort().initializeList(new InitializeRecord[]{ir1, ir2});
```

C# (initialize)

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

            InitializeRef ref1 = new InitializeRef();
            ref1.type = InitializeRefType.salesOrder;
```

```
    //internal id of the sales order to be converted to cash sale
           ref1.internalId = "792";
           ref1.typeSpecified = true;

           InitializeRecord rec = new InitializeRecord();
           rec.type = InitializeType.cashSale;
           rec.reference = ref1;

           ReadResponse read1 = _service.initialize(rec);
               }
```

login

The login operation is used to authenticate a user and start a new Web services session in NetSuite. The login operation is similar to the login for the UI. This operation provides a passport that includes a username, password, account and role. On success, the NetSuite server sets a cookie and establishes a session.

Important: Users can also authenticate to NetSuite by providing their user credentials in the SOAP header of each request; they do not need to invoke the login operation. With user credentials provided in each request, the need for session management and separate logins is eliminated. For more information, see Authentication Using Request Level Credentials.

All available Web services operations require that the user first be logged in. Once successfully completed, a login creates a session that allows subsequent operations to be performed without having to log in again until the session expires or the logout operation is invoked. If the session times out, the next operation fails. Web services requests initiated through a client application must have the ability to execute the login operation when a session has timed out and then submit the original request again.

Note: For information on session management, refer to Session Management.

If an operation is executed before a login is performed, it fails and the InvalidSessionFault is returned. Also note that after login, only one Web services request may be in flight at a time for a given session. Any attempt to violate this will result in a SOAP fault.

NetSuite Web services will use the last logged-in role when users do not specify a role in the login request and no default role is set. This makes the Web services login behavior consistent with the UI login behavior. Partner applications that rely on a specific role should be careful to specify that role in the login request, otherwise their users might be logged in with a role that is not appropriate for that application.

The login operation also verifies that the specified account has the Web Services feature enabled. If Web services is not enabled in your account, the SOAP fault InvalidCredentialsFault is returned with a code of WEB_SERVICES_NOT_ENABLED. (See Enabling the Web Services Feature for steps on enabling this feature.)

Request

The LoginRequest type is used for this request. It contains the following fields:

Element Name	XSD Type	Notes
passport	Passport	Contains all the required credentials including username, password, account and role to authenticate the user and create a new session.

The Passport type includes the following elements:

- email
- password
- account
- role

Note: You can confirm your accountID in the NetSuite UI. As administrator, go to Support > Customer Service > Contact Support by Phone. Your account number is displayed in a pop-up box. Also, Role is not a required parameter in the WS login. However, if you don't specify a role, the user's default role must have WS permissions.

Response

The LoginResponse type is used for the response. This references the SessionResponse type, which includes the status and wsRoleList elements.

The wsRoleList element returns a list of roles available for the user specified in the passport. You can then use this list of roles to execute different routines depending on available roles or to re-login with a different role.

Element Name	XSD Type	Notes
userID	RecordRef	References an existing user record.
role	RecordRef	References an existing role record.
isDefault	boolean	If true, the role is set as the default role for this user.
isInactive	boolean	If true, this role is currently unavailable for the current user.
isLoggedInRole	boolean	If true, the role of the logged in user is used.

Retrieving userID in Axis 1.1

Following is sample code that illustrates how you can retrieve the userID from the header when using Axis 1.1 — where **port** is the service port for a given endpoint.

```
String userid = getHeader(port, "sessionInfo");
```

```
public static String getHeader(NetSuitePortType port, String headerName) {
 com.netsuite.webservices.platform.NetSuiteBindingStub stub =
 (com.netsuite.webservices.platform.NetSuiteBindingStub) port;
 SOAPHeaderElement [] headers = stub.getResponseHeaders();
 for (int i=0; i< headers.length; i++)
 {
  SOAPHeaderElement header = headers[i];
  if (header.getName().equals(headerName))
  {
   Iterator childElements = header.getChildElements();
   while (childElements.hasNext())
   {
    SOAPElement el = (SOAPElement) childElements.next();
    return el.getValue();
   }
  }
 }
 return null;
}
```

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InsufficientPermissionFault
- InvalidAccountFault
- InvalidSessionFault
- InvalidCredentialsFault
- UnexpectedErrorFault

Sample Code

SOAP Request

In the following example a user is logged in with the Administrator role as indicated by the internalId of 3. For a list of available role IDs see Internal IDs Associated with Roles.

```
<soap:Body>
<platformMsgs:login>
 <platformMsgs:passport>
  <platformCore:email>jsmith@yahoo.com</platformCore:email>
  <platformCore:password>password123</platformCore:password>
  <platformCore:account>121511</platformCore:account>
  <platformCore:role internalId="3"/>
</platformMsgs:passport>
</platformMsgs:login>
</soap:Body>
```

SOAP Response

```
<soapenv:Body>
<loginResponse xmlns="urn:messages_1_2.platform.webservices.netsuite.com">
<sessionResponse xmlns="urn:messages_1_2.platform.webservices.netsuite.com">
 <ns1:status isSuccess="true" xmlns:ns1="urn:core_1_2.platform.webservices.netsuite.com"/>
 <ns2:wsRoleList xmlns:ns2="urn:core_1_2.platform.webservices.netsuite.com">
  <ns2:wsRole>
   <ns2:role internalId="3">
    <ns2:name>Administrator</ns2:name>
   </ns2:role>
   <ns2:isDefault>false</ns2:isDefault>
   <ns2:isInactive>false</ns2:isInactive>
  </ns2:wsRole>
  <ns2:wsRole>
   <ns2:role internalId="15">
    <ns2:name>Employee Center</ns2:name>
   </ns2:role>
   <ns2:isDefault>false</ns2:isDefault>
   <ns2:isInactive>false</ns2:isInactive>
  </ns2:wsRole>
  <ns2:wsRole>
   <ns2:role internalId="22">
    <ns2:name>Intranet Manager</ns2:name>
   </ns2:role>
   <ns2:isDefault>false</ns2:isDefault>
   <ns2:isInactive>false</ns2:isInactive>
  </ns2:wsRole>
  <ns2:wsRole>
   <ns2:role internalId="25">
    <ns2:name>System Administrator</ns2:name>
   </ns2:role>
   <ns2:isDefault>false</ns2:isDefault>
   <ns2:isInactive>false</ns2:isInactive>
  </ns2:wsRole>
 </ns2:wsRoleList>
</sessionResponse>
</loginResponse>
</soapenv:Body>
```

C#

```
private void login( bool isAuto )
{
if ( !_isAuthenticated )
 {
 // Check whether this is a forced login as part of another operation
  if ( isAuto )
  _out.writeLn( "\nYou need to first login before invoking this operation ..." );

  // Enable client cookie management. This is required.
  _service.CookieContainer = new CookieContainer();

  // Populate Passport object with all login information
  Passport passport = new Passport();
  RecordRef role = new RecordRef();

  // Determine whether to get login information from config
```

```
   // file or prompt for it
   if ( "true".Equals( _dataCollection["promptForLogin"] ) )
   {
    _out.writeLn( "\nPlease enter your login information: " );
    _out.write( "E-mail: " );
    passport.email = _out.readLn();
    _out.write( "Password: " );
    passport.password = _out.readLn();
    _out.write( "Role nsKey (press enter for default administrator role): " );
    role.internalId = _out.readLn();
    passport.role =role;
    _out.write( "Account: " );
    passport.account = _out.readLn();
   }
   else
   {
    passport.email = _dataCollection["login.email"];
    passport.password = _dataCollection["login.password"];
    role.internalId = _dataCollection["login.roleNSkey"];
    passport.role =role;
    passport.account= _dataCollection["login.acct"];
   }

   // Login to NetSuite
   _out.info( "\nLogging into NetSuite" );
   _out.info( "Username: " + passport.email );
   _out.info( "Account: " + passport.account );
   Status status = (_service.login( passport )).status;

   // Process response
   if ( status.isSuccess == true )
   {
    _isAuthenticated = true;
    _out.info( "\nThe user with nsKey=" + _service.sessionInfo.userId + " was logged
    in successful and a new session has been created." );
   }
   else
   {
    // Should never get here since any problems with the
    // login should have resulted in a SOAP fault
    _out.error( getStatusDetails( status ) );
   }
  }
}
```

Java

```
public void login(boolean isAuto) throws RemoteException {
if (!_isAuthenticated) {
 // Check whether this is a forced login as part of another operation
 if (isAuto)
  _console
  .writeLn("\nYou need to first login before invoking this operation ...");

  // Populate Passport object with all login information
  Passport passport = new Passport();
  RecordRef role = new RecordRef();

  // Determine whether to get login information from config
```

```
  // file or prompt for it
  if ("true".equals(_properties.getProperty("promptForLogin"))) {
   _console.writeLn("\nPlease enter your login information: ");
   System.out.print("E-mail: ");
   passport.setEmail(_console.readLn());
   System.out.print("Password: ");
   passport.setPassword(_console.readLn());
   System.out.print("Role nsKey (press enter for default administrator role): ");
   role.setInternalId(_console.readLn());
   passport.setRole(role);
   System.out.print("Account: ");
   passport.setAccount(_console.readLn());
  } else {
   passport.setEmail(_properties.getProperty("login.email"));
   passport.setPassword(_properties.getProperty("login.password"));
   role.setInternalId(_properties.getProperty("login.roleNSkey"));
   passport.setRole(role);
   passport.setAccount(_properties.getProperty("login.acct"));
  }

  // Login to NetSuite
  _console.info("\nLogging into NetSuite");
  _console.info("Username: " + passport.getEmail());
  _console.info("Account: " + passport.getAccount());

  Status status = (_port.login(passport)).getStatus();
  // Process the response
  if (status.isIsSuccess() == true) {
   _isAuthenticated = true;
   _console
   .info("\nThe login was successful and a new session has been created.");
  } else {
   // Should never get here since any problems with the
   // login should have resulted in a SOAP fault
   _console.error(getStatusDetails(status));
  }
 }
}
```

logout

The logout operation is used to terminate an active session.

Note: If you explicitly logout of a session, and then attempt to utilize the same session, the SESSION_TIMED_OUT error message is returned.

Request

The logoutRequest type is used for the request. It does not contain any fields.

Response

The status type is used for the response. It does not contain any fields.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- ExceededRequestLimitFault
- UnexpectedErrorFault

Sample Code

Following is an example that contains an excerpt of the SOAP body for both the request and response.

SOAP Request

```
<logout/>
```

SOAP Response

```
<logoutResponse>
 <status isSuccess=”true”/>
</logoutResponse>
```

C#

```
private void logout()
{
 if ( _isAuthenticated )
 {
  _out.info( "\nLogging out of NetSuite\n" );

  // Logout from NetSuite
  Status status = (_service.logout()).status;

  if ( status.isSuccess == true )
  {
   _isAuthenticated = false;
   _out.info( "Logout successful" );
  }
  else
  {
   // Should never get here since any problems with the
   // logout should have resulted in a SOAP fault
   _out.error( getStatusDetails( status ) );
  }
 }
 else
 {
```

```
  _out.info(
  "\nThe logout() operation cannot be invoked because there is no active session. " +
  "You must be first logged on before attempting to logout.\n" );
 }
}
```

Java

```
public void logout() throws RemoteException {
 if (_isAuthenticated) {
  _console.info("\nLogging out of NetSuite\n");

  // Logout from NetSuite
  Status status = (_port.logout()).getStatus();

  if (status.isIsSuccess() == true) {
   _isAuthenticated = false;
   _console.info("Logout successful");
  } else {
   // Should never get here since any problems with the
   // logout should have resulted in a SOAP fault
   _console.error(getStatusDetails(status));
  }
 } else {
  _console
  .info("\nThe logout() operation cannot be invoked because there is no active session. "
  + "You must be first logged on before attempting to logout.\n");
 }
}
```

mapSso

Single sign-on (SSO) refers to the procedure that allows a user of two or more user-authenticating Web applications to move between these applications using a browser, only presenting **authentication** information once per session.

The mapSso operation supports NetSuite's inbound single sign-on feature. This feature allows users to go directly from an external user-authenticating application to NetSuite, without having to separately log in to NetSuite. Validation is provided by the passing of an encrypted token, and user identification is provided by a mapping between external application credentials (company ID and user ID), and NetSuite credentials (email, password, account, and role used to log in to NetSuite).

The mapSso operation streamlines the provisioning of inbound single sign-on access to NetSuite accounts by automating the mapping of credentials between an external application and NetSuite. If this operation is not used, this mapping is created for each user by requiring them to log in to NetSuite on their first single sign-on access.

This operation allows users to access NetSuite through inbound single sign-on without knowing their NetSuite password. Use of this operation is required for inbound single sign-integrations to Web store.

Important: This operation provides mapping, not a login to NetSuite. This operation does NOT provide the ability to log in using a single sign-on token, cannot provisian a partner with the inbound single sign-on feature, or aid with public/private key exchange. Use of this operation implies that the account already has inbound single sign-on set up and has access to the associated partner ID and encryption keys needed to generate the token.

Be aware of the following:

- If a NetSuite role used for inbound single sign-on is deleted, the single sign-on mapping for any user with that role is automatically remapped to another role.
- If a user has a single sign-on mapping set up with a particular role and that role is removed from the user, the mapping is deleted. You can set up a new mapping for that user with a different role.
- Single sign-on mappings are not copied from a production account to a sandbox account when the sandbox is refreshed. These mappings must be recreated in the sandbox account for any users who require inbound single sign-on access to that account.

For more detailed information on NetSuite's inbound single sign-on feature, see the help topic *Inbound Single Sign-on*.

Important: Since September 2012, when NetSuite began hosting new customer accounts in multiple data centers, the URL to be used when running Web services operations depends upon the data center hosting each NetSuite account. For example, it could be webservices.netsuite.com or webservices.na1.netsuite.com. In the 2012.2 and later Web services endpoints, you can use the getDataCenterUrls operation to dynamically discover the correct URL for Web services access to a NetSuite account. A REST service is also available for this purpose. See the help topic *Using the REST roles Service to Get User Accounts, Roles, and Domains*. For more information, see the help topic *Understanding Multiple Data Centers*.

Request

The MapSsoRequest type is used for this request. It contains the following fields:

Element Name	XSD Type	Notes
ssoCredentials	SsoCredentials	Contains all the required credentials including username (email address), password, account, and role, to authenticate the user and create a new session.

The SsoCredentials type requires the following fields:

- email

- password
- account
- role
- authentication token: this is a string representing the encrypted token.
    - This token, prior to any hex-encoding, is of the form:

```
<companyID><space><userID><space><timestamp>
```

- Because spaces are used to delimit subtokens, companyID and userID values cannot include spaces.
- The timestamp string is a decimal representation of the number of milliseconds since January 1, 1970, 00:00:00 GMT.
- partner ID: this is the integration partner's affiliate ID as provided to the partner by NetSuite.

Response

The MapSsoResponse type is used for the response. It does not contain any fields.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidAccountFault
- InvalidCredentialsFault
- InvalidVersionFault
- ExceededRequestLimitFault
- UnexpectedErrorFault

Sample Code

SOAP Request

```
<soap:Body>
<platformMsgs:mapSso>
```

```
<platformMsgs:ssoCredentials>
<platformCore:email>bbrown@yahoo.com</platformCore:email>
<platformCore:password>mypassword</platformCore:password>
<platformCore:account>TSTDRV211915</platformCore:account>
<platformCore:authenticationToken>[enter token here]</platformCore:authenticationToken>
<platformCore:partnerId>[enter partner id here]</platformCore:partnerId>
</platformMsgs:ssoCredentials>
</platformMsgs:mapSso>
</soap:Body>
 <?xml version="1.0" encoding="UTF-8" ?>
- <Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:core_2008
_2.platform.webservices.netsuite.com" xmlns:ns2="urn:messages_2008_2.platform.webservices.netsu
ite.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <Header>
- <passport actor="http://schemas.xmlsoap.org/soap/actor/next">
  <email>bob@freidman.com</email>
  <password>[Content Removed for Security Reasons]</password>
  <account>TSTDRV216916</account>
  <role internalId="3" xsi:type="RecordRef" />
  </passport>
  </Header>
- <Body>
- <mapSso>
- <ssoCredentials>
  <email>bamber@sso.com</email>
  <password>[Content Removed for Security Reasons]</password>
  <account>TSTDRV216916</account>
<authenticationToken>73724D816695EF65D3AA2D391E206A77F8A7419EB52041CCF396024CD6E277A2E1F3249EBA
C5634A2D51BCE67D7AA2B796BADB4C2616DBB206E9439C5D310792E77E60D7FE7C4EF8B444C171A39622C29BC84D984
BEBE8BBD5D788E9E44DCF09F514F73175D86E89B925D3BF8D52FCEF314B716ED618B74F3E086AA9A1AE048E</authen
ticationToken>
  <partnerId>195863</partnerId>
  </ssoCredentials>
  </mapSso>
  </Body>
  </Envelope>
```

SOAP Response

```
<?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:Header>
- <platformMsgs:sessionInfo xmlns:platformMsgs="urn:messages_2008_2.platform.webservices.netsui
te.com">
  <platformMsgs:userId>698</platformMsgs:userId>
  </platformMsgs:sessionInfo>
- <platformMsgs:documentInfo xmlns:platformMsgs="urn:messages_2008_2.platform.webservices.netsu
ite.com">
  <platformMsgs:nsId>WEBSERVICES_TSTDRV216916_112320088757457221199759391_773a9c1</platformMsgs
:nsId>
  </platformMsgs:documentInfo>
  </soapenv:Header>
- <soapenv:Body>
- <mapSsoResponse xmlns="urn:messages_2008_2.platform.webservices.netsuite.com">
- <sessionResponse>
  <platformCore:status isSuccess="true" xmlns:platformCore="urn:core_2008_2.platform.webservice
s.netsuite.com" />
- <platformCore:wsRoleList xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.co
```

```
m">
- <platformCore:wsRole>
- <platformCore:role internalId="14">
  <platformCore:name>Customer Center</platformCore:name>
  </platformCore:role>
  <platformCore:isDefault>false</platformCore:isDefault>
  <platformCore:isInactive>false</platformCore:isInactive>
  </platformCore:wsRole>
  </platformCore:wsRoleList>
  </sessionResponse>
  </mapSsoResponse>
  </soapenv:Body>
  </soapenv:Envelope>
```

Java

```
 /* Generate a NetSuitePort */
 NetSuiteServiceLocator nss = new NetSuiteServiceLocator();
 NetSutePortType myNetSuitePort = nss.getNetSuitePort();

/* Here your code needs to generate a valid Netsuite Sso token */
 String mySsoToken = MySSOManager.getTokenFromIds
 ("myCompanyID","thisUsersUIDinMySystem");
 RecordRef rr = new RecordRef();
 rr.setid("23");

 /* Setup the Credential */
 SsoCredentials sc = new SsoCredentials();
 sc.setAccount("TSTDRV00000");
 sc.setEmail("bob@happycow.com");
 sc.setPassword("fr0mCA");
 sc.setRole(rr);
 sc.setAuthenticationToken(mySsoToken);

 /* Now initiate the mapping*/
 try{
  SessionResponse sr = myNetSuitePort.mapSso(sc);
  if (!sr.getStatus().isIsSuccess())
  throw new Exception("Mapping Failed: " + sr.getStatus().getStatusDetail(0).getMessage());
}
```

search

The *search* operation is used to execute a search on a specific record type based on a set of criteria. You can search by defining search filter fields on the record, joined fields on an associated record, search return columns, or joined search return columns from an associated record. The results of the search can be complete records, or a select set of fields specified through search return columns.

Note that you can also use the *search* operation to return an existing saved search. You cannot use the search operation to retrieve state values. You must use the getAll operation to retrieve all state values in the system. The getAll 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.

Important: Be aware that the search preferences you set in SearchPreferences object affect the search request and response. See Setting Web Services Preferences for details.

Use the *search* operation to execute the following types of searches:

- **Basic search** — Execute a search on a record type based on search filter fields that are specific to that type. See Basic Searches in Web Services.
- **Joined search** — Execute a search on a record type based on search filter fields on an associated record type. See Joined Searches in Web Services
- **Advanced search** — Execute a search on a record type in which you specify search filter fields **and/or** search return columns or joined search columns. Using advanced search, you can also return an existing saved search. See Advanced Searches in Web Services.

Note: An asynchronous equivalent is available for the search operation, **asyncSearch**. For information about asynchronous request processing, see Synchronous Versus Asynchronous Request Processing.

Also see the following sections for information on setting additional search filtering values and searching for custom fields:

- Setting Valid Search Values
- Setting the anyof, mine, or myteam Filtering Values
- Searching by lastModifiedDate
- Understanding Sorting in Advanced Search
- Search-Related Sample Code
- Searching for a Multi-select Custom Field

Important: To go directly to search-related code samples, see Search-Related Sample Code.

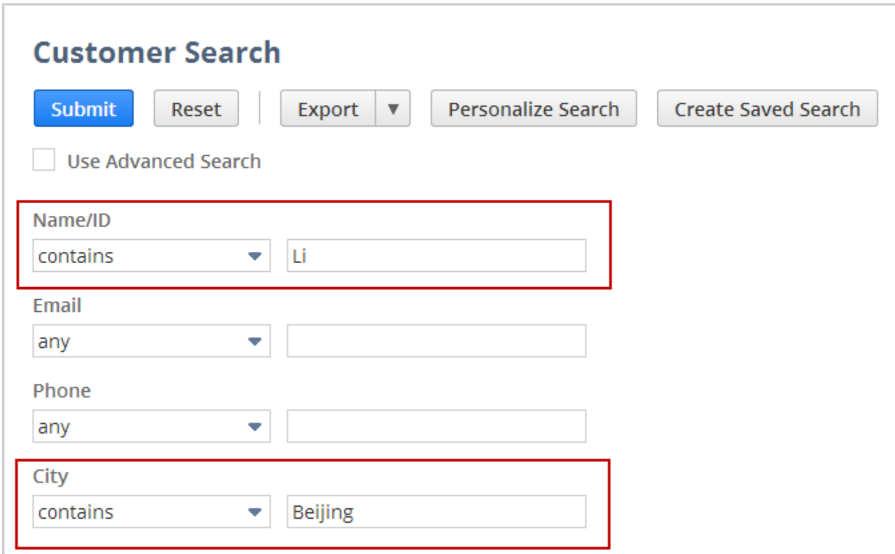
Basic Searches in Web Services

See the following topics to learn how to execute a basic search in NetSuite using Web services:

- What is a basic search?
- Which SuiteTalk objects are used in a basic search?
- Basic Search Code Sample

What is a basic search?

A basic search allows you to search records of a specific type using the fields on that record as search filters. The following figure shows the UI equivalent of a basic Customer search. Start by going to Lists > Relationships > Customers > Search, and ensure that the Use Advanced Search box is not checked. You can specify one or more field values to use as filters for search results.



In a basic search, field criteria are the **only** values you set. You cannot specify search return columns . In Web services, specifying search return columns is the equivalent of performing an advanced search. (See Advanced Searches in Web Services for details.)

In the example of a basic Customer search (see previous figure), the results returned include the record ID of every customer that has the Category field set to *From Advertisement* and the Status field set to *Customer-Closed Won*. Note that **ALL** the other data associated with these specific customer records are returned as well. As a consequence, in Web services a basic search tends to increase the search response time.

Which SuiteTalk objects are used in a basic search?

To perform a basic search in which you specify search filter criteria only, use:

1. < *Record* > **Search**
2. < *Record* > **SearchBasic**

For more details, see Basic Search Objects Explained and Basic Search Code Sample.

Basic Search Objects Explained

In SuiteTalk, any record that supports search has a corresponding < *Record* > **Search** object, which contains a *basic* element. The *basic* element references a < *Record* > **SearchBasic** object, which defines all available search criteria (filter fields) specific to that record type.

The XSD snippet below shows the CustomerSearch object. The *basic* element references the CustomerSearchBasic object, which defines all available search criteria for the Customer record.

```
<complexType name=" CustomerSearch ">
        <complexContent>
            <extension base="platformCore:SearchRecord">
                <sequence>
                    <element name=" basic " type="platformCommon: CustomerSearchBasic " minOccu
rs="0"/>
                    <element name="callJoin" type="platformCommon:PhoneCallSearchBasic" minOccu
rs="0"/>
                    <element name="campaignResponseJoin" type="platformCommon:CampaignSearchBas
ic"
   minOccurs="0"/>
           .....

  </sequence>
            </extension>
        </complexContent>
    </complexType>
```

Note: < *Record* > **Search** objects reside in the same XSD as their corresponding record objects. In this case, both the Customer and CustomerSearch objects reside in the listRelXSD. Also note that the CustomerSearch object, like all < *Record* > **Search** objects, provide available search joins for that record type. For information on joined searches, see Joined Searches in Web Services.

This snippet shows the CustomerSearchBasic object. All < *Record* > **SearchBasic** objects are defined in the platformCommon XSD. This sample shows four of the available fields (accountNumber, address, addressee, addressLabel) that can be used as search criteria on the Customer record.

```
<complexType name=" CustomerSearchBasic ">
      <complexContent>
 <extension base="platformCore:SearchRecord">
    <sequence>
    <element name=" accountNumber " type="platformCore:SearchStringField" minOccurs="0" />
    <element name=" address " type="platformCore:SearchStringField" minOccurs="0" />
   <element name=" addressee " type="platformCore:SearchStringField" minOccurs="0" />
    <element name=" addressLabel " type="platformCore:SearchStringField" minOccurs="0" />
       ...
   </sequence>
         </extension>
   </complexContent>
</complexType>
```

For a code sample of a basic search, see Basic Search Code Sample.

Joined Searches in Web Services

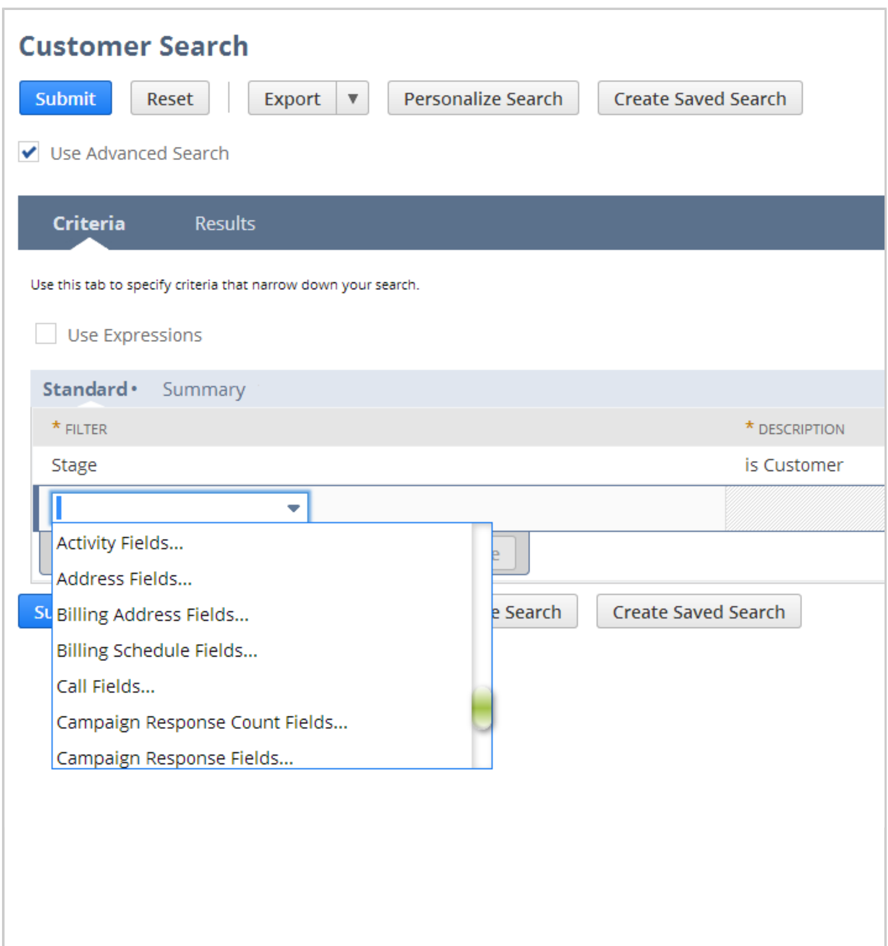
See the following topics to learn how to execute a joined search in NetSuite using Web services:

- What is a joined search?
- Which SuiteTalk objects are used in a joined search?
- Joined Search Code Samples

What is a joined search?

A joined search allows you search against a specific record type using the fields on an associated record as search filters. In the UI, you can identify which associated records provide joined filter criteria by first navigating to a record's search interface. For example, for the Customer search interface, go to Lists > Relationships > Customers > Search.

Check the Use Advanced Search box and wait for the page to update. Then scroll through the Filter dropdown list. Joined search records are indicated by the record name followed by an ellipsis (....). Search fields from any of the records listed (that are also currently exposed in SuiteTalk) can be included in the search criteria.



Note: For a list of SuiteTalk-supported records, see the help topic *Web Services Supported Records*.

Which SuiteTalk objects are used in a joined search?

To perform a joined search in which you use search filter criteria from an associated record, use:

1. < *Record* > **Search**
2. < *Record* > **SearchBasic**

For more details, see Joined Search Objects Explained and Joined Search Code Samples.

Joined Search Objects Explained

In SuiteTalk, all search joins are listed in the < *Record* > **Search** object. For example, to find available search joins for the Contact or Employee record, see the ContactSearch and EmployeeSearch XSDs, respectively.

The snippet below shows the CustomerSearch object, which includes < *xxx* > **Join** elements. These elements reference search criteria available from other SuiteTalk-supported records.

```
<complexType name=" CustomerSearch ">
        <complexContent>
            <extension base="platformCore:SearchRecord">
                <sequence>
                    <element name="basic" type="platformCommon:CustomerSearchBasic" minOccurs="
0"/>
                    <element name=" callJoin " type="platformCommon: PhoneCallSearchBasic " min
Occurs="0"/>
                    <element name=" campaignResponseJoin " type="platformCommon: CampaignSearch
Basic "
   minOccurs="0"/>
                    <element name=" caseJoin " type="platformCommon: SupportCaseSearchBasic " m
inOccurs="0"/>
  .....

  </sequence>
            </extension>
        </complexContent>
    </complexType>
```

In this case, all search filter criteria from the PhoneCallSearchBasic, CampaignSearchBasic, and SupportCaseSearchBasic objects are available to the Customer record as joined search filters. Note that all < *Record* > **SearchBasic** objects in NetSuite Web services are defined in the platformCommon XSD.

For a code sample of a joined search, see Joined Search Code Samples.

Important: Only fields on **SuiteTalk-supported** records can be specified as filter criteria for a joined search request. For a list of SuiteTalk-supported records, see the help topic *Web Services Supported Records*.

Returning an Associated Joined List of Records

Using a combination of joined search and the internalId list on each record, you can retrieve a list of records for an associated list of records. For example, you can retrieve a list of contacts for a given list of customers. In order to do this, you must first retrieve the desired list of internalIds for the record you need to retrieve by, and then submit that list in a joined search query to retrieve the associated list.

See Joined Search Code Samples for an example of a joined search which uses a list of records.

Advanced Searches in Web Services

Advanced searching in SuiteTalk provides users with the ability to:

- Perform a search that references an existing saved search. See Reference Existing Saved Searches.
- Perform a search that references an existing saved search, and then overrides existing search return columns with new search return columns. See Specify Search Criteria and Search Return Columns.
- Perform a search that references an existing saved search, and then provides additional search filter criteria (on top of the criteria already specified in the saved search). See Specify Search Criteria and Search Return Columns.
- Perform a search that specifies search criteria and search result columns. See Specify Search Criteria and Search Return Columns.

The SuiteTalk API includes advanced search objects for all records that have an existing search interface. To see which objects are used to execute advanced searches, see Which SuiteTalk objects are used in advanced search?

For advanced search code samples, see Advanced Search Code Samples.

Important: Note that advanced search functionality will not work in endpoints prior to 2008.2. Also note that searchMore, searchNext, and respective asynchronous operations can be used in advanced search, as well as all search preferences and pagination functionality.

The following is an XSD snippet of the listRel XSD. This sample provides a high-level overview of advanced search objects and their relationship to the basic search object < *Record* > **Search**. In this sample, the objects Customer **SearchAdvanced**, Customer **SearchRow**, and Customer

SearchRowBasic (not shown in sample) are considered to be advanced search objects in the SuiteTalk API.

```
<complexType name="CustomerSearch">
    <complexContent>
        <extension base="platformCore:SearchRecord">
            <sequence>
                <element name="basic" type="platformCommon:CustomerSearchBasic" minOccurs="0"/>
                <element name="callJoin" type="platformCommon:PhoneCallSearchBasic" minOccurs="0"/>
                .....
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="CustomerSearchAdvanced">
    <complexContent>
        <extension base="platformCore:SearchRecord">
            <sequence>
                <element name="criteria" type="listRel:CustomerSearch" minOccurs="0"/>
                <element name="columns" type="listRel:CustomerSearchRow" minOccurs="0"/>
            </sequence>
            <attribute name="savedSearchId" type="xsd:string"/>
            <attribute name="savedSearchScriptId" type="xsd:string"/>
        </extension>
    </complexContent>
</complexType>
<complexType name="CustomerSearchRow">
    <complexContent>
        <extension base="platformCore:SearchRow">
            <sequence>
                <element name="basic" type="platformCommon:CustomerSearchRowBasic" minOccurs="0"/>
                <element name="callJoin" type="platformCommon:PhoneCallSearchRowBasic" minOccurs="0"/>
                ....
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

Use to search by existing fields on the record or joined fields on associated records. See Basic Searches in Web Services and Joined Searches in Web Services.

Use to search by existing fields on the record **and/or** available search return columns. Can also be used to return an existing saved search. See: Specify Search Criteria and Search Return Columns and Reference Existing Saved Searches.

Use to specify search return columns or search column joins to return in a search response. See Specify Search Criteria and Search Return Columns.

Reference Existing Saved Searches

Advanced search in Web services allows you to reference an existing saved search. Returning saved searches provides you with access to data that otherwise could not be returned in SuiteTalk. For example, advanced search allows you to return saved searches that include formula filters (in the search criteria) or expressions.

When working with saved searches, two questions come to mind for Web services users:

- Why reference an existing saved search?
- How do I reference an existing saved search?

Why reference an existing saved search?

The following use cases illustrate possible scenarios for referencing a saved search:

- You want to return only a subset of data from a record (in other words, data that is specified through the saved search's return columns).
- Your integration application processes a set of records that are identified in a saved search. Periodically, users change the criteria of the search. By referencing a saved search ID, your code can just reference their saved search. Developers do not have to change and re-test code every time the search criteria changes.

- You have a complex search that compiles data from many different records. You can create a saved search in the NetSuite UI, and then reference this search in Web services rather than try to code the search in Web services.
- You want to reference an existing saved search based on Leads, for example. You can return all the data provided in this search, and then define additional criteria for the search response. For example, you can return a Leads saved search and then provide additional criteria that returns the leads from this search created with today's date. In other words, if you reference a saved search and add any filter criteria to the search request, the additional criteria will be conjunctive with the saved search criteria.

How do I reference an existing saved search?

First you must obtain the saved search ID. You can do so through the UI by going to Lists > Search > Saved Searches. The saved search ID appears in the ID column.

You can also use the getSavedSearch operation to programmatically retrieve a list of saved search IDs for a specific record type. Note that this operation does nothing more than return a list of saved search IDs on a per-record-type basis (for example, all saved search IDs for the Customer record type).

You can then use the search() operation, along with the < *Record* > **SearchAdvanced** object to return the details of the saved search. The following is a simple example that shows how to instantiate the CustomerSearchAdvanced object and specify a savedSearchId to return.

```
// create search object
CustomerSearchAdvanced customerSearch = new CustomerSearchAdvanced();

//set saved search id
customerSearch.savedSearchId="100";

// perform the search
NetSuiteService nss = new NetSuiteService();
SearchResult result = nss.search(customerSearch);
```

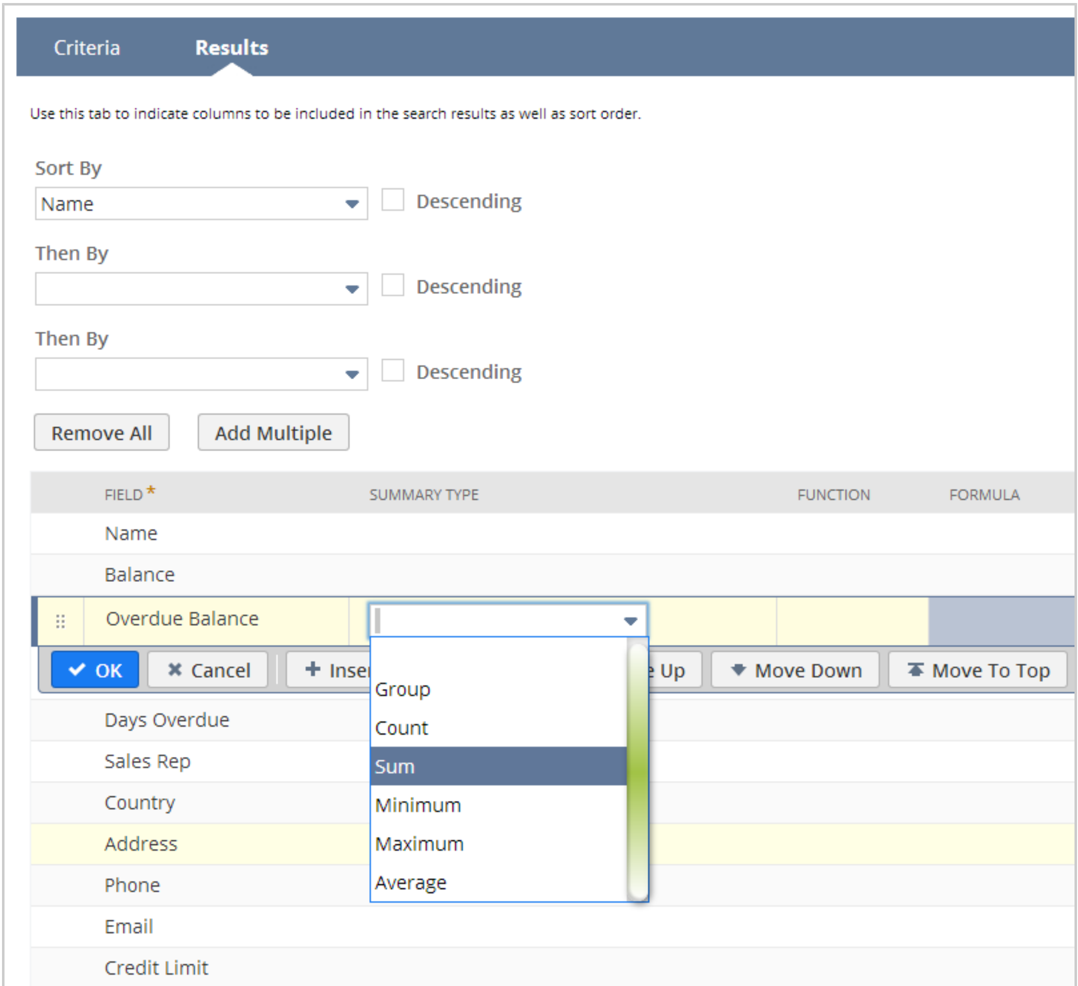
For more detailed samples, see Advanced Search Code Samples.

Usage Notes

- Only one saved search can be referenced as part of the search call.
- If you reference a saved search that contains search functions, you *will* get the results back. However, you cannot *create* an advanced search that uses search functions. Creating a search in Web services that uses search functions is not currently supported.
- If you reference a saved search that contains summary results, you will get the following error:

  We cannot return search columns for summary saved search < *saved search ID* >

The following figure shows the UI equivalent of setting a summary type on a search return column. If a value is set, the saved search cannot be returned.



Specify Search Criteria and Search Return Columns

Similar to a basic search, an advanced search allows you search against a specific record type using the fields on that record as search filters. In addition, you can **also** specify a set of search return columns or joined columns to return.

This kind of advanced search is useful for retrieving only the record data you need rather than the contents of an entire record.

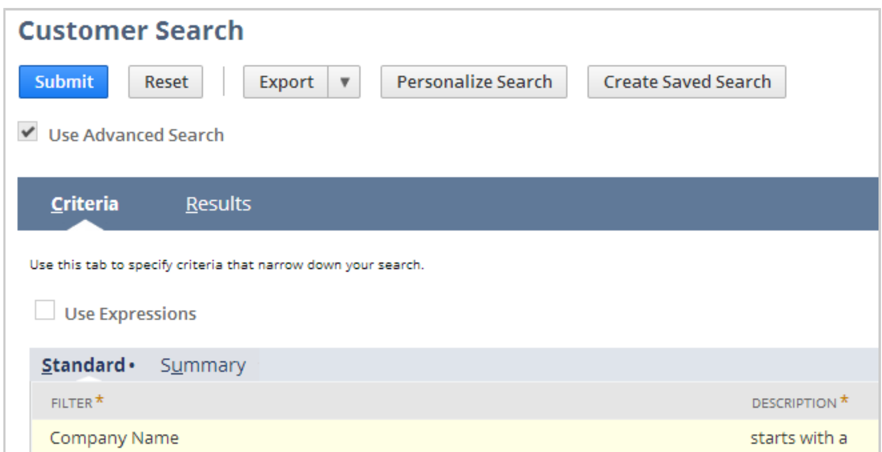
Advanced searches work well, for example, for users who may have a mobile client that needs to display only the name, phone number, and email address of their sales rep contact. Rather than returning all of the data on a Contact record, users can create an advanced search that pulls only the relevant information. They are no longer required to download the entire record.

For advanced search code samples that show how to specify both search criteria and search return columns, see Advanced Search Code Samples.

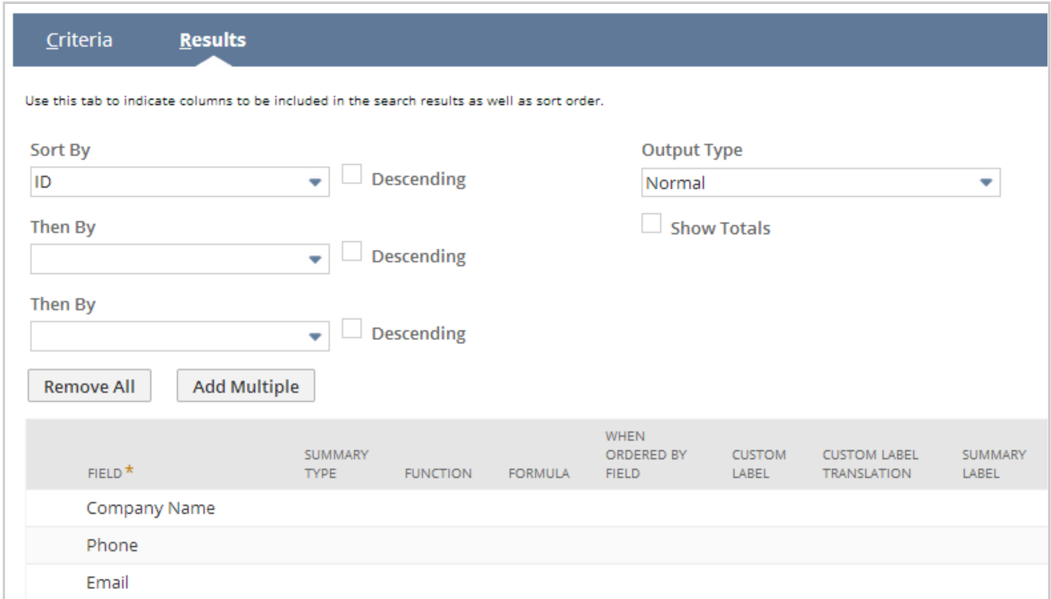
What are search return columns?

The following figures show the UI equivalent of an advanced search that includes search return columns.

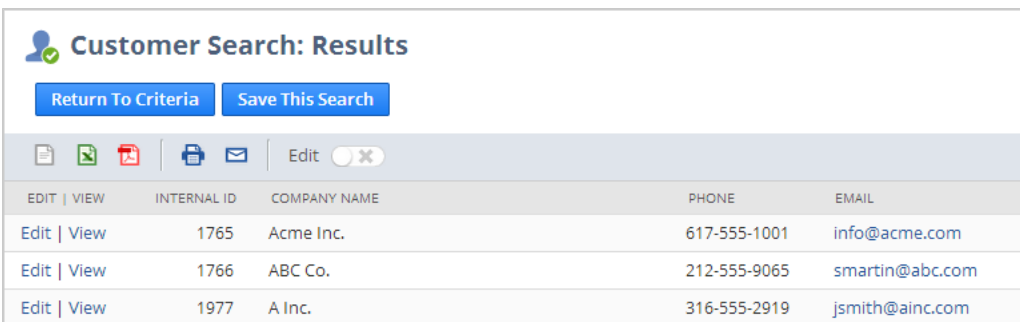
1. First, specify the basic search criteria on the Criteria subtab (in this example all customers that have a company name starting with the letter A).



2. Next, click the Results subtab to define the search return columns. When your search is executed, your search response will return only the company name, phone, and contact name of all customers whose company starts with the letter A.



The following figure shows the results of the search. Only the relevant data are returned.



Which SuiteTalk objects are used in advanced search?

The following summarizes the objects used in different types of advanced searches. For additional details, see Advanced Search Objects Explained and Advanced Search Code Samples.

- To perform a search that specifies search filter criteria and search result columns, use:
    1. < *Record* > **Search**
    2. < *Record* > **SearchBasic**
    3. < *Record* > **SearchRow**
    4. < *Record* > **SearchRowBasic**

- To perform a search that references an existing saved search, use:
  1. < *Record* > **SearchAdvanced**
- To perform a search that references an existing saved search, and then overrides existing search return columns with new search return columns, use:
  1. < *Record* > **SearchAdvanced**
  2. < *Record* > **SearchRow**
  3. < *Record* > **SearchRowBasic**
- To perform a search that references an existing saved search, and then provides additional search filter criteria (that is in *addition* to the criteria already specified in the saved search), use:
  1. < *Record* > **SearchAdvanced**
  2. < *Record* > **Search**
  3. < *Record* > **SearchBasic**

Advanced Search Objects Explained

The < Record >SearchAdvanced object contains:

- a *criteria* element - references the < *Record* > **Search** object through which you specify standard search field criteria. (See Basic Searches in Web Services for details on the < *Record* >Search object.)
- *a columns* element - references the < *Record* > **SearchRow** object through which you specify a set of search result columns to return in the response.
- a *savedSearchId* attribute - references the saved search internal ID (for example, 57, 99, 63).

Note: If the Show Internal ID preference is enabled in your NetSuite account, the saved search internal ID appears in the **Internal ID** column in a saved search list. Programmatically, you can use getSavedSearch to obtain a list of saved search internal IDs for a specific record type.

- a *savedSearchScriptId* attribute - references a custom saved search ID (for example, customsearch_mySpecialSearch).

Note: If the Show Internal ID preference is enabled in your NetSuite account, the saved search ID appears in the **ID** column in a saved search list.

Important: You cannot use getSavedSearch to return a list of custom saved search IDs; only the system-defined internal IDs (57, 99, 63, etc.) will be returned.

The XSD sample below shows the CustomerSearchAdvanced object:

```
<complexType name=" CustomerSearchAdvanced ">
 <complexContent>
  <extension base="platformCore:SearchRecord">
   <sequence>
    <element name=" criteria " type="listRel:CustomerSearch" minOccurs="0"/>
    <element name=" columns " type="listRel:CustomerSearchRow" minOccurs="0"/>
    </sequence>
    <attribute name=" savedSearchId " type="xsd:string"/>
    <attribute name=" savedSearchScriptId " type="xsd:string"/>
   </extension>
```

The < Record >SearchRow object contains:

- a *basic* element - references the < *Record* > **SearchRowBasic** object, which specifies available search return columns and column joins for *that* record type.
- <xxx>Join elements - references the < *Record* > **SearchRowBasic** object, which specifies search return columns and column joins for the *associated* record type.

The XSD below shows a snippet of the CustomerSearchRow object:

```
<complexType name=" CustomerSearchRow ">
 <complexContent>
  <extension base="platformCore:SearchRow">
       <sequence>
         <element name=" basic " type="platformCommon:CustomerSearchRowBasic"
     minOccurs="0"/>
         <element name=" callJoin " type="platformCommon:PhoneCallSearchRowBasic"
    minOccurs="0"/>
         <element name=" campaignResponseJoin "type="platformCommon:CampaignSearchRowBasic"
    minOccurs="0"/>
   <element name=" caseJoin " type="platformCommon:SupportCaseSearchRowBasic"
    minOccurs="0"/>
    ......
      </sequence>
  </extension>
 </complexContent>
</complexType>
```

The < Record >SearchRowBasic object contains:

- search result column elements - use to define specific column names in your response.

The next snippet shows the CustomerSearchRowBasic object, which, like ALL < *Record* > **SearchRowBasic** objects, resides in the platformCommon XSD. This object lists all available search return columns for the Customer record.

```
<complexType name=" CustomerSearchRowBasic ">
        <sequence>
            <element name=" accountNumber " type="platformCore: SearchColumnStringField " minOc
curs="0"
  maxOccurs="unbounded"/>
            <element name=" address " type="platformCore: SearchColumnStringField " minOccurs="
0"
```

```
  maxOccurs="unbounded"/>
            <element name=" addressee " type="platformCore: SearchColumnStringField " minOccurs
="0"
  maxOccurs="unbounded"/>
            <element name=" addressLabel " type="platformCore: SearchColumnStringField " minOcc
urs="0"
  maxOccurs="unbounded"/>
           ...

        </sequence>
    </complexType>
```

Note that all search return columns reference SearchColumn< *xxx* >Field objects. See Search Column Custom XML Schema Types for definitions of each search column object.

Important: When executing an advanced search, you can set the SearchPreferences. *returnSearchColumns* preference to TRUE to ensure that only search return columns are returned in a search. An error is thrown if *returnSearchColumns* is set to TRUE and you have not specified search return columns in your request. (Note that you will **not** receive an error if you are using advanced search functionality to return a **saved search** that already includes search return columns.)

Note: Also note that in an advanced search, the *bodyFieldsOnly* preference is ignored.

The default value for *returnSearchColumns* is TRUE.

```
 <complexType name="SearchPreferences">
 <sequence>
  <element name="bodyFieldsOnly" minOccurs="0" type="xsd:boolean" default="true"/>
  <element name=" returnSearchColumns " minOccurs="0" type="xsd:boolean"
  default="true"/>
  <element name="pageSize" minOccurs="0" type="xsd:int"/>
 </sequence>
</complexType>
```

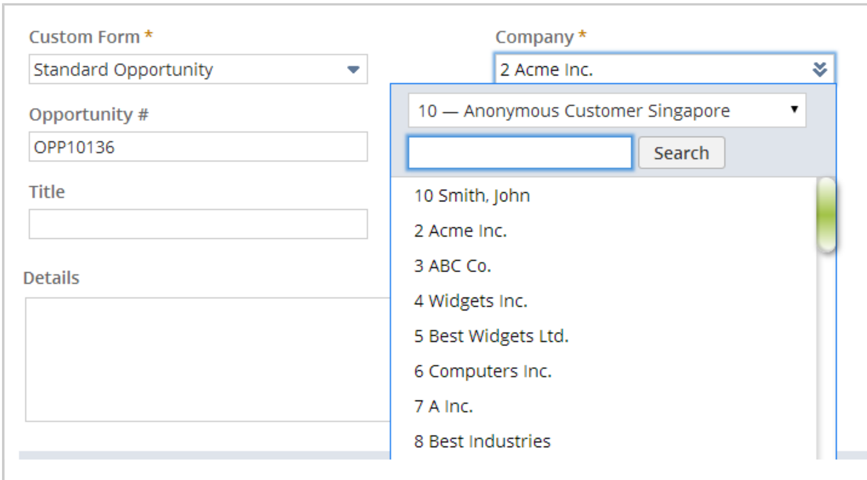
For detailed information on setting search preferences, see Setting Search Preferences.

How Custom Fields Are Identified in Search Results

When executing an advanced search, you might choose to include as part of your search results values from custom fields. In some cases, the way that the system identifies these values varies depending on which endpoint you are using. For many users, this difference might not be of great consequence. However, you may want to review this information if you are upgrading from a previous endpoint to the 2013.2 WSDL or later, and if your integration depends on these values.

Specifically, this difference applies to scenarios in which you execute an advanced search, join to a record that has a custom field, and include the custom field in your search results. With 2013.1 and earlier endpoints, the system in its response adds a prefix to the string used to identify the custom field. In 2013.2 and later, this prefix is not added.

For example, suppose you had a custom item field on your inventory item record, and this custom field had an ID of custitem_style.



First, note that this value is referred to as internalId when using 2013.1 or an earlier endpoint. With 2013.2 and later endpoints, the value is called scriptId.

Second, with older endpoints, in some cases, the system modifies this value slightly. For example: Suppose you executed a TransactionSearchAdvanced and joined to the item record. If you included in your search results the custom field pictured above, the 2013.1 endpoint, or an earlier one, would identify the field's internalId as having a value of "item_custitem_style." That is, in this particular scenario, the system would add the "item_" prefix. With a different type of join, a different prefix would be added.

If you are using the 2013.2 endpoint or later, the system does **not** add a prefix. The scriptId returned in the response is the same as in the request ("custitem_style," in this example). Therefore, if you upgrade to 2013.2 or a later WSDL and if you have integrations that depend on this value, you must modify these integrations.

Example — Behavior with 2013.1 and Earlier Endpoints

In the following example, the user executes a TransactionSearchAdvanced operation with a join to the item record. The user is searching for estimates that are associated with inventory items. In the results, the user wants to include the value of the custom field with the internalId "custitem_color."

```
<soapenv:Body>
 <platformMsgs:search>
  <platformMsgs:searchRecord xsi:type="s0:TransactionSearchAdvanced">
   <s0:criteria>
    <s0:basic>
     <platformCommon:internalId operator="anyOf">
      <platformCore:searchValue internalId="11962" type="estimate" />
     </platformCommon:internalId>
```

```
      <platformCommon:type operator="anyOf">
       <platformCore:searchValue>_estimate</platformCore:searchValue>
      </platformCommon:type>
     </s0:basic>
     <s0:itemJoin>
      <platformCommon:type operator="anyOf">
       <platformCore:searchValue>_inventoryItem</platformCore:searchValue>
      </platformCommon:type>
     </s0:itemJoin>
    </s0:criteria>
    <s0:columns>
     <s0:itemJoin>
      <platformCommon:customFieldList>
       <platformCore:customField xsi:type="platformCore:SearchColumnSelectCustomField" internalI
d="custitem_color" />
         </platformCommon:customFieldList>
     </s0:itemJoin>
    </s0:columns>
   </platformMsgs:searchRecord>
  </platformMsgs:search>
</soapenv:Body>
```

The response might look like the following. Note that the internalId value is now “item_custitem_color.”

```
<platformCore:searchRowList>
   <platformCore:searchRow xsi:type="tranSales:TransactionSearchRow" xmlns:tranSales="urn:sales
_2013_1.transactions.webservices.netsuite.com">
      <tranSales:itemJoin xmlns:platformCommon="urn:common_2013_1.platform.webservices.netsuite
.com">
         <platformCommon:customFieldList>
            <platformCore:customField internalId="item_custitem_color" xsi:type="platformCore:S
earchColumnSelectCustomField">
               <platformCore:searchValue internalId="1" typeId="6" />
            </platformCore:customField>
         </platformCommon:customFieldList>
      </tranSales:itemJoin>
   </platformCore:searchRow>
</platformCore:searchRowList>
```

Example — Behavior with 2013.2 and Later Endpoints

The SOAP request below represents another TransactionSearchAdvanced that joins to the item record. (In this example, because the 2013.2 WSDL is being used, the “custitem_color” value is identified as the scriptId, rather than internalId.)

```
<soapenv:Body>
 <search xmlns="urn:messages_2013_2.platform.webservices.netsuite.com">
  <searchRecord xsi:type="ns6:TransactionSearchAdvanced" xmlns:ns6="urn:sales_2013_2.transactio
ns.webservices.netsuite.com">
   <ns6:criteria xsi:type="ns6:TransactionSearch">
    <ns6:basic xsi:type="ns7:TransactionSearchBasic" xmlns:ns7="urn:common_2013_2.platform.webs
ervices.netsuite.com">
     <ns7:internalId operator="anyOf" xsi:type="ns8:SearchMultiSelectField" xmlns:ns8="urn:core
_2013_2.platform.webservices.netsuite.com">
      <ns8:searchValue internalId="67" xsi:type="ns8:RecordRef"/>
```

```
     </ns7:internalId>
           <ns7:type operator="anyOf" xsi:type="ns9:SearchEnumMultiSelectField" xmlns:ns9="urn:
core_2013_2.platform.webservices.netsuite.com">
      <ns9:searchValue xsi:type="xsd:string">_estimate</ns9:searchValue>
           </ns7:type>
    </ns6:basic>
    </ns6:criteria>
    <ns6:columns xsi:type="ns6:TransactionSearchRow">
     <ns6:itemJoin xsi:type="ns10:ItemSearchRowBasic" xmlns:ns10="urn:common_2013_2.platform.we
bservices.netsuite.com">
      <ns10:customFieldList xsi:type="ns11:SearchColumnCustomFieldList" xmlns:ns11="urn:core_20
13_2.platform.webservices.netsuite.com">
       <ns11:customField scriptId="custitem_color" xsi:type="ns11:SearchColumnSelectCustomField
"/>
      </ns10:customFieldList>
     </ns6:itemJoin>
    </ns6:columns>
   </searchRecord>
  </search>
</soapenv:Body>
```

In the SOAP response, the scriptId value – “custitem_color” – is unchanged. No prefix has been added.

```
<platformCore:searchRowList>
  <platformCore:searchRow xsi:type="tranSales:TransactionSearchRow" xmlns:tranSales="urn:sales_
2013_2.transactions.webservices.netsuite.com">
     <tranSales:itemJoin xmlns:platformCommon="urn:common_2013_2.platform.webservices.netsuite.
com">
        <platformCommon:customFieldList>
           <platformCore:customField internalId="211" scriptId="custitem_color" xsi:type="platf
ormCore:SearchColumnSelectCustomField">
              <platformCore:searchValue internalId="1" typeId="28"/>
           </platformCore:customField>
        </platformCommon:customFieldList>
     </tranSales:itemJoin>
  </platformCore:searchRow>
</platformCore:searchRowList>
```

Joining Through Custom Fields

If you are using 2013.2 or a later WSDL, you may notice that many *RecordType* Search objects include CustomSearchJoin. This object allows you to join to records that are associated with the primary record through a custom field.

That is, if Record A includes a custom field of type list/record or multiple select, and this custom field references another record that is exposed to SuiteTalk, the 2013.2 WSDL and later WSDLs allow your search of Record A to include a join to the SearchBasic of the referenced record.

Similarly, many *RecordType* SearchRow objects now include CustomSearchRowBasic. You can use this object to include columns from the referenced record in the results of an advanced search.

See the following for more details on executing searches using CustomSearchJoin and CustomSearchRowBasic:

- Overview of CustomSearchJoin and CustomSearchRowBasic
- CustomSearch Objects Defined
- Example – Using CustomSearchJoin and CustomSearchRowBasic
- CustomSearchJoin Usage Notes

Overview of CustomSearchJoin and CustomSearchRowBasic

This topic describes some of the basics of using CustomSearchJoin and CustomSearchRowBasic.

Executing a Search

To use CustomSearchJoin, you can execute either a joined search or an advanced search.

For example, consider an employee record that has been altered to include a custom field pointing to the account record. In this case, you can search the employee record and use CustomSearchJoin to include data from the account record as part of your search criteria. You would execute your search using either of the following two objects:

- EmployeeSearch — Results returned would include the entire employee record for each employee that meets your criteria. (For general details on this type of search, see Joined Searches in Web Services.)
- EmployeeSearchAdvanced — Results returned could include any columns you specify from the employee record or the account record. (For general details on this type of search, see Advanced Searches in Web Services.) In this scenario, you would use CustomSearchRowBasic as well as CustomSearchJoin.

Sample Use Cases for CustomSearchJoin

You can use CustomSearchJoin in a couple of different scenarios.

A common case is one in which you have altered a standard record type to include a custom field that references another standard record type. This scenario includes the example where your employee record has been altered to include a custom field that points to the account record.

Additionally, you can join to and from custom record types. For example, suppose your organization uses custom record types called Customer Survey and Customer Suggestion. If Customer Survey and Customer Suggestion both include fields that reference each other, a search of Customer Survey could include a join to Customer Suggestion and vice versa.

Similarly, you can join between the following:

- A customized record type that references a standard record type.
- A standard record type that references a custom record type.

Sample Use Cases for CustomSearchRowBasic

CustomSearchRowBasic can be used in conjunction with all of the custom field usage described above for CustomSearchJoin.

Additionally, your search does not have to include the use of CustomSearchJoin in order for you to use CustomSearchRowBasic. For example, you could use EmployeeSearchAdvanced to search using criteria that is defined entirely on the employee record (and not use any joins). But in your results, you might still want to include data from the record referenced by a custom field on the employee record. You do this using CustomSearchRowBasic.

Note that CustomSearchRowBasic is available only when you are executing an advanced search.

Note: CustomSearchJoin and CustomSearchRowBasic are available only for accounts that have upgraded to 2013.2 or a later endpoint.

CustomSearch Objects Defined

For accounts that have upgraded to 2013.2 or a later endpoint, common.xsd includes two elements that support joining to and gathering result data from records that are referenced by a custom field. These objects are CustomSearchJoin and CustomSearchRowBasic. The main purpose of these objects is described in Overview of CustomSearchJoin and CustomSearchRowBasic. This topic describes how the objects are defined in the schema.

CustomSearchJoin is included in many *RecordType* Search objects. This element is defined as follows:

```
<complexType name="CustomSearchJoin">
    <sequence>
        <element name="customizationRef" type="platformCore:CustomizationRef"/>
        <element name="searchRecordBasic" type="platformCore:SearchRecordBasic"/>
    </sequence>
</complexType>
```

CustomSearchRowBasic is included in many *RecordType* SearchRow objects. This element is defined as follows:

```
<complexType name="CustomSearchRowBasic">
    <sequence>
        <element name="customizationRef" type="platformCore:CustomizationRef"/>
        <element name="searchRowBasic" type="platformCore:SearchRowBasic"/>
    </sequence>
</complexType>
```

Notice that both of these elements use CustomizationRef, which you use to identify the custom field. CustomizationRef includes five possible attributes. Of these, you can use only internal ID or script ID to identify a custom field. The other attributes (external ID, type, and name) are not applicable to custom fields.

The following example shows how you can use script ID to reference a custom field and join to AccountSearchBasic:

```
CustomizationRef field_account = new CustomizationRef(null,null,null,null,"custentity_account")
;
CustomSearchJoin customSearchJoin = new CustomSearchJoin(field_account,accountSearchBasic);
```

For a more extensive example, see Example – Using CustomSearchJoin and CustomSearchRowBasic.

Note: CustomSearchJoin and CustomSearchRowBasic are available only for accounts that have upgraded to the 2013.2 or a later endpoint.

Example – Using CustomSearchJoin and CustomSearchRowBasic

If you are using the 2013.2 WSDL or later, you have access to CustomSearchJoin and CustomSearchRowBasic. This topic shows how you might use these objects in your code. For details on the general purpose of these objects and sample use cases, see Overview of CustomSearchJoin and CustomSearchRowBasic.

The example below assumes that your employee record has been altered to include a custom field that references the account record. This sample shows how you might use EmployeeSearchAdvanced to find employee records associated with an account of a certain type and value.

```
public  void testSearch1() throws Exception{


// We are searching for employees associated with a certain type of account. Here we specify th
at account.

AccountSearchBasic accountSearchBasic = new AccountSearchBasic();
accountSearchBasic.setType(new SearchEnumMultiSelectField(new String[]{AccountType.__bank}, Sea
rchEnumMultiSelectFieldOperator.anyOf));


// Here we specify the field on the employee record that references the account record.

CustomizationRef field_account = new CustomizationRef();
field_account.setScriptId("custentity_account");


// Here we create a custom search join object.
CustomSearchJoin customSearchJoin = CustomSearchJoin(field_account, accountSearchBasic);


// Next, we pass the custom search join that we just created into an EmployeeSearch instance.

EmployeeSearch employeeSearch = new EmployeeSearch();
```

```
employeeSearch.setCustomSearchJoin(new CustomSearchJoin[] {customSearchJoin});


// Next, we specify that we want the results to include the balance of the account.

AccountSearchRowBasic accountSearchRowBasic = new AccountSearchRowBasic();
accountSearchRowBasic.setBalance(new SearchColumnDoubleField[]{new SearchColumnDoubleField()});

CustomSearchRowBasic customSearchRowBasic =new CustomSearchRowBasic(field_account,accountSearch
RowBasic);


// Here, we specify that we want the employee's internal ID included in the results.

EmployeeSearchRowBasic employeeSearchRowBasic = new EmployeeSearchRowBasic();
employeeSearchRowBasic.setInternalId(new SearchColumnSelectField[]{new SearchColumnSelectField(
)});


// In the following snippet, we pass the specified search columns into an EmployeeSearchRow ins
tance.

EmployeeSearchRow employeeSearchRow = new EmployeeSearchRow();
employeeSearchRow.setCustomSearchJoin(new CustomSearchRowBasic[]{customSearchRowBasic});
employeeSearchRow.setBasic(employeeSearchRowBasic);


// Create the advanced search.

EmployeeSearchAdvanced employeeSearchAdvanced = new EmployeeSearchAdvanced();
employeeSearchAdvanced.setCriteria(employeeSearch);
employeeSearchAdvanced.setColumns(employeeSearchRow);

c.search(employeeSearchAdvanced);

}
```

Note: CustomSearchJoin and CustomSearchRowBasic are available only for accounts that have upgraded to the 2013.2 or a later endpoint.

CustomSearchJoin Usage Notes

If you are using the 2013.2 WSDL or later, you have access to CustomSearchJoin. Refer to the following sections to better understand the capabilities of this search object:

- Use CustomSearchJoin Only With Custom Fields
- CustomSearchJoin Connects Only With SearchBasics
- Nested Joins Are Not Allowed

Use CustomSearchJoin Only With Custom Fields

You cannot use CustomSearchJoin in conjunction with a standard field that references another record. For example, the standard employee record includes the Department field. If you decide to use CustomSearchJoin, and you identify the Department field as your custom field, the system returns an error reading “Invalid custom field reference.” You should use the standard departmentJoin that is already provided as part of the EmployeeSearch object.

CustomSearchJoin Connects Only With SearchBasics

CustomSearchJoin allows only for a basic search of the referenced record. For example, if the employee record has been altered to include a custom field that points to the account record, you can join to AccountSearchBasic – but not AccountSearch or AccountSearchAdvanced. This limitation is consistent with the capabilities of searching in the UI.

Nested Joins Are Not Allowed

You cannot use nested joins. For example, suppose your contact record includes a custom field referencing the employee record, which includes a custom field that references the account record. A search of the contact record could include a join to the employee record, but the same search could not *also* join to the account record.

Note: CustomSearchJoin and CustomSearchRowBasic are available only for accounts that have upgraded to the 2013.2 or a later endpoint.

Setting Valid Search Values

Prior to the 2008.2 endpoint, if you performed a search that included an invalid search enum filter value, you would generally still get records returned in your search. For example, if you performed a search for all customers with the Country enum search value set to "United States" (rather than the supported enum value _unitedStates), you would still get results from your search. Although the value "United States" was not recognized, you continued to get customer records. Note, however, the search results returned ALL customers in the system, since the value “United States” was invalid.

Starting with the 2008.2 endpoint, instead of simply ignoring any invalid search enum values, and still returning search results, NetSuite now returns 0 records and a no-match warning. Therefore, when setting search values, be sure to use the values defined in the schema.

The following is an example of what is now returned if invalid values are specified:

```
<platformCore:searchResult
xmlns:platformCore="urn:core_2008_2.platform.webservices.netsuite.com">
<platformCore:status isSuccess="true">
<platformCore:statusDetail type="WARN">
<platformCore:code>WARNING</platformCore:code>
<platformCore:message>The field country's enum value <United States> is
invalid for this search.</platformCore:message>
</platformCore:statusDetail>
</platformCore:status>
<platformCore:totalRecords>0</platformCore:totalRecords>
<platformCore:totalPages>0</platformCore:totalPages>
<platformCore:searchId>WEBSERVICES_MSTRWLF_102120085637216058968423l6_60315faa132ad</platformCo
re:searchId>
<platformCore:searchRowList/>
```

Setting the anyof, mine, or myteam Filtering Values

In SuiteTalk you can further define your search using the following filtering values:

Filter	Note
@NONE@	Equates to *anyof* (see Filtering Lists that Contain Null Values) or *unassigned* depending on the field.
@CURRENT@	Equates to *mine*. For example, use this filter to return all of your own events.
@HIERARCHY@	Equates to *my team*. For example, use this filter on a salesRep field for Customer records. If you have previously defined the members of your sales team, using the @HIERARCHY@ filter will return only the customers that have worked with members of your sales team.

The following provides a SOAP sample for finding *my* events.

```
<search xmlns="urn:messages_2_6.platform.webservices.netsuite.com">
  <searchRecord xsi:type="ns1:CalendarEventSearchBasic"
xmlns:ns1="urn:common_2_6.platform.webservices.netsuite.com">
     <ns1:attendee operator="anyOf" xsi:type="ns2:SearchMultiSelectField"
xmlns:ns2="urn:core_2_6.platform.webservices.netsuite.com">
        <ns2:searchValue internalId="@CURRENT@" xsi:type="ns2:RecordRef"/>
     </ns1:attendee>
 </searchRecord>
 </search>
```

Filtering Lists that Contain Null Values

For select lists or multi selects which can have a null value, the UI supports the search criteria on these list type fields as “None Of” “-None-“, which essentially means not “Any Of” all list options. Such a search would result in all records which do NOT have this list type field as null. In order to accomplish this “None Of” “-None-“ search you need to set the internalId of the search key to “@None@”.

Example

To search for Customers which have a Partner associated with them in NetSuite, the SOAP would look as below for the “Partner field not null” part:

```
<ns3:partner operator="noneOf">
<ns8:searchValue internalId="@NONE@" xmlns:ns8="urn:core_2_5.platform.webservices.netsuite.com"
/>
</ns3:partner>
<ns3:customFieldList>
```

Java code to generate this SOAP would be:

```
// Adding search criteria where Partner is not null
RecordRef[] noPartner = new RecordRef[1];
noPartner[0] = new RecordRef();
noPartner[0].setInternalId("@NONE@");
SearchMultiSelectField partner = new SearchMultiSelectField();
partner.setOperator(SearchMultiSelectFieldOperator.noneOf);
```

```
partner.setSearchValue(noPartner);
custSearch.setPartner(partner);
```

C# code to generate this SOAP would be:

```
// Adding search criteria where Partner is not null
RecordRef[] noPartner = new RecordRef[1];
noPartner[0] = new RecordRef();
noPartner[0].internalId = "@NONE@";
SearchMultiSelectField partner = new SearchMultiSelectField();
partner.@operator = SearchMultiSelectFieldOperator.noneOf;
partner.searchValue = noPartner;
custSearch.partner = partner;
```

Searching by lastModifiedDate

This sample shows how to create a customer, and then search for the customer that was created based on a given time frame. In this case, the sample uses the lastModifiedDate field to search “within” a couple of seconds before the customer was created and then a minute after. The search returns the record that was previously created.

Java

```
Customer c = (Customer) new TestCustomer().createMinimalRecord();
Calendar timeFrom = Calendar.getInstance();
WriteResponse wr = sessMgr.getPort().add(c);
outputResult(wr.getStatus().isIsSuccess());

Calendar timeTo = Calendar.getInstance();
timeTo.setTimeInMillis(timeTo.getTimeInMillis() + 60000);

CustomerSearch cs = new CustomerSearch();
CustomerSearchBasic csb = new CustomerSearchBasic();
SearchDateField sdf = new SearchDateField();
sdf.setOperator(SearchDateFieldOperator.within);
sdf.setSearchValue(timeFrom);
sdf.setSearchValue2(timeTo);
csb.setLastModifiedDate(sdf);

cs.setBasic(csb);

SearchResult sr = sessMgr.getWrappedPort().search(cs, this);
```

SOAP

```
<searchRecord xsi:type="ns1:CustomerSearch"
 xmlns:ns1="urn:relationships_2_5.lists.webservices.netsuite.com">
              <ns1:basic xsi:type="ns2:CustomerSearchBasic"
 xmlns:ns2="urn:common_2_5.platform.webservices.netsuite.com">
                 <ns2:lastModifiedDate operator="within" xsi:type="ns3:SearchDateField"
 xmlns:ns3="urn:core_2_5.platform.webservices.netsuite.com">
                    <ns3:searchValue xsi:type="xsd:dateTime">2007-02-10T00:16:17.750Z</ns3:sea
rchValue>
```

```
                    <ns3:searchValue2 xsi:type="xsd:dateTime">2007-02-10T00:17:53.015Z</ns3:se
archValue2>
                 </ns2:lastModifiedDate>
              </ns1:basic>
           </searchRecord>
```

Understanding Sorting in Advanced Search

In Web services, when users return a saved search that has sorting criteria specified in the saved search, the records are returned according to the specified *sorted by* order. To see the *sorted by* criteria that has been applied to a saved search, users can look at the saved search criteria in the UI.

In an ad-hoc Web services search (a search in which *sorted by* criteria have not been set), users should be aware of the implicit *sorted by* order in which records are returned. This order is based on record type. Record types and the default *sort by* order for each record type are listed in the following table:

Note: For a list of all records associated with each type, see the help topic *Web Services Supported Records*.

Record Type	Default "Sorted by" Order
Entities	Name (the name of the entity)
Actvities	Event (the title of the Event)
Marketing	Campaign ID **Note:** Promotion Code searches are sorted and returned by promotion code name.
Transactions	Date (the transaction date)
Support	Number **Note:** Topic searches are sorted and returned by topic title.
File Cabinet	Name (the name of the file or folder)
Items	Name (the name of the item)
Communications (includes the Note and the Message records)	Note - sorted/returned by Author Message - sorted/returned by Message Internal ID
Website	Name
Lists	Name **Note:** Gift Certificate searches are sorted and returned by Name (From).

Request

The SearchRequest type is used for the request. It contains the following field.

Element Name	XSD Type	Notes
searchRecord	SearchRecord	The SearchRecord type is an abstract type. An instance of a type that extends SearchRecord must be used—such as CustomerSearchBasic or EventSearchBasic.

Response

The SearchResponse type is used for the response. It contains the following fields.

Element Name	XSD Type	Notes
status	Status	The status for this search. All applicable errors or warnings will be listed within this type.
totalRecords	xsd:int	The total number of records for this search. Depending on the pageSize value, some or all the records may be returned in this response
pageSize	xsd:int	The page size for this search.
totalPages	xsd:int	The total number of pages that are part of this search.
pageIndex	xsd:int	The page index for the current set of results.
searchId	string	Returns a specific search based on its search ID.
recordList	Record[]	A list of records that meet the criteria for this search. The actual records returned need to be of a type that extends the abstract type of record.
searchRowList	SearchRowList	A list of return columns that meet the criteria for this search.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Search-Related Sample Code

See the following search-related code samples:

- Basic Search Code Sample
- Joined Search Code Samples
- Advanced Search Code Samples
- Searching for a Multi-select Custom Field

Basic Search Code Sample

SOAP Request

In the following example, customer records that have an email that contains shutterfly.com are searched for. Note that you can limit the search page size at the request level (see Pagination).

```
<soap:Body>
<platformMsgs:search>
<searchRecord xsi:type="ContactSearch">
 <customerJoin xsi:type="CustomerSearchBasic">
  <email operator=”contains” xsi:type="platformCore:SearchStringField">
  <platformCore:searchValue>shutterfly.com</platformCore:searchValue>
  <email>
 <customerJoin>
</searchRecord>
</search>
</soap:Body>
```

SOAP Response

Notice that in this example, only one matching record was found. You can see that the page size was set such that if multiple records were found, only 10 would be returned at a time. The searchMore or searchNext operation could then be performed to return additional results.

```
<soapenv:Body>
<searchResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
<searchResult xmlns="urn:core_2_5.platform.webservices.netsuite.com">
<status isSuccess="true"/>
<totalRecords>1</totalRecords>
<pageSize>10</pageSize>
<totalPages>1</totalPages>
<pageIndex>1</pageIndex>
<recordList>
 <record internalId="983" xsi:type="ns1:Customer"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:ns1="urn:relationships_2_5.lists.webservices.netsuite.com">
  <ns1:entityId>Shutter Fly</ns1:entityId>
  <ns1:isInactive>false</ns1:isInactive>
  <ns1:companyName>Shutter Fly, Inc</ns1:companyName>
  <ns1:entityStatus internalId="6"><name>LEAD-New</name>
  .
  .
  .
  <ns1:customFieldList>
   <customField internalId="custentity_map" xsi:type="StringCustomFieldRef">
    <value>http://maps.google.com</value>
   </customField>
```

```
    <customField internalId="custentity_had_order_problems"
    xsi:type="BooleanCustomFieldRef"><value>false</value></customField>
   </ns1:customFieldList>
  </record>
</recordList>
</searchResult>
</searchResponse>
</soapenv:Body>
```

C#

```
 private void searchCustomer()
{
 // This operation requires a valid session
 this.login( true );

 _out.writeLn( "\nEnter search parameters" );

 // Instantiate a search object for customers. Note that the search
 // object is different from the regular record used for add and update.
 CustomerSearch custSearch = new CustomerSearch();

 // Search the customer entity id which is a string field
 _out.write( "Entity ID (press enter to skip): " );
 String nameValue = _out.readLn();
 SearchStringField entityId = null;
 if ( !nameValue.Equals( "" ) )
 {
  entityId = new SearchStringField();
  entityId.@operator = SearchStringFieldOperator.contains;
  entityId.operatorSpecified = true;
  entityId.searchValue = nameValue;
  custSearch.basic.entityId = entityId; //see note below for an alternative
 }
//Note:  You could also use CustomerSearchBasic to access entityId directly, for example:
  CustomerSearchBasic custSearchBasic = new CustomerSearchBasic();
  custSearchBasic.entityId = entityId;
 // Search the customer stage which is a list field
 _out.write( "Customer Stage (one or more nsKeys separated by commas, press enter to skip): " )
;
 String stageKeysValue = _out.readLn();
 SearchMultiSelectField stage = null;
 if ( !stageKeysValue.Equals( "" ) )
 {
  stage = new SearchMultiSelectField();
  stage.@operator = SearchMultiSelectFieldOperator.anyOf;
  stage.operatorSpecified = true;

  string [] nskeys = stageKeysValue.Split( new Char[] {','} );

  RecordRef[] recordRefs = new RecordRef[ stageKeysValue.Length ];
  for (int i=0; i<nskeys.Length; i++ )
  {
   RecordRef recordRef = new RecordRef();
   recordRef.internalId = nskeys[i];
   recordRefs[i] = recordRef;
  }
  stage.searchValue = recordRefs;
  custSearch.stage = stage;
```

```
}

// Search by isActive field which is a boolean
/*
SearchBooleanField isActive = new SearchBooleanField();
while ( true )
{
 _out.write( "Is active [T/F] (default is T): " );
 String upcomingStr = _out.readLn();

 if ( "T".Equals( upcomingStr.ToUpper() ) || "".Equals( upcomingStr.ToUpper() ) )
 {
  isActive.searchValue = true;
  isActive.searchValueSpecified = true;
  break;
 }
 else if ( upcomingStr.ToUpper().Equals( "F" ) )
 {
  isActive.searchValue = false;
  isActive.searchValueSpecified = true;
  break;
 }
 else
 {
  _out.writeLn( "Invalid selection" );
 }
}
custSearch.active = isActive;
*/
if ( custSearch.entityId == null && custSearch.stage == null )
{
 _out.info( "\nNo search criteria was specified. Searching for all records." );
}
else
{
 _out.info(
 "\nSearching for customers with the following criteria: " +
 (entityId==null ? "" : ("\nentityID=" + custSearch.entityId.searchValue) + ",
 Operator=" + entityId.@operator.ToString()) +
 (stage==null ? "" : ("\nstage nsKeys='" + stageKeysValue + "',
 Operator=" + stage.@operator.ToString())) );
}


// Invoke search() web services operation
SearchResult response = _service.search( custSearch );

// Process response
if ( response.status.isSuccess )
{
 // Process the records returned in the response and print to console
 processCustomerSearchResponse( response );

 // Since pagination controls what is returned, check to see
 // if there are anymore pages to retrieve.
 searchMore( response );
}
else
{
 _out.error( getStatusDetails( response.status ) );
```

```
 }
}
```

Java

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

 _console.writeLn("\nEnter search parameters");

 // Instantiate a search object for customers. Note that the search
 // object is different from the regular record used for add and update.
 CustomerSearch custSearch = new CustomerSearch();

 // Search the customer entity id which is a string field
 _console.write("Entity ID (press enter to skip): ");
 String nameValue = _console.readLn();
 SearchStringField entityId = null;
 if (!nameValue.equals("")) {
  entityId = new SearchStringField();
  entityId.setOperator(SearchStringFieldOperator.contains);
  entityId.setSearchValue(nameValue);
  custSearch.setEntityId(entityId);
 }

 // Search the customer stage which is a list field
 _console
 .write("Customer Stage (one or more nsKeys separated by commas, press enter to skip): ");
 String stageKeysValue = _console.readLn();
 SearchMultiSelectField stage = null;
 if (!stageKeysValue.equals("")) {
  stage = new SearchMultiSelectField();
  stage.setOperator(SearchMultiSelectFieldOperator.anyOf);

  String[] nskeys = stageKeysValue.split(",");

  RecordRef[] recordRefs = new RecordRef[stageKeysValue.length()];
  for (int i = 0; i < nskeys.length; i++) {
   RecordRef recordRef = new RecordRef();
   recordRef.setInternalId(nskeys[i]);
   recordRefs[i] = recordRef;
  }
  stage.setSearchValue(recordRefs);
  custSearch.setStage(stage);
 }
 if (custSearch.getEntityId() == null && custSearch.getStage() == null) {
  _console
  .info("\nNo search criteria was specified. Searching for all records.");
 } else {
  _console
  .info("\nSearching for customers with the following criteria: "
  + (entityId == null ? ""
  : ("\nentityId=" + custSearch
  .getEntityId().getSearchValue())
  + ", Operator="
  + entityId.getOperator().toString())
```

```
  + (stage == null ? "" : ("\nstage nsKeys='"
  + stageKeysValue + "', Operator=" + stage
  .getOperator().toString())));
 }

 // Set page size for number of records to be returned in search
 // response

 // Invoke search() web services operation
 SearchResult result = _port.search(custSearch);

 // Process result
 if (result.getStatus().isIsSuccess()) {
  // Process the records returned in the result and print to console
  processCustomerSearchResponse(result);

  // Since pagination controls what is returned, check to see
  // if there are anymore pages to retrieve.
  searchMore(result);
 } else {
  _console.error(getStatusDetails(result.getStatus()));
 }
}
```

Joined Search Code Samples

Sample 1 — Customer Search With Contact Join

This sample shows how to execute a customer search in which the contact email address is specified as the search criteria.

Java

```
public void contactSearch_with_CustomerJoin() throws Exception {
    ContactSearch cs = new ContactSearch();
    ContactSearchBasic contactSearchBasic = new ContactSearchBasic();
    contactSearchBasic.setEmail(new SearchStringField("contact@email.com", SearchStringFieldOpe
rator.is));
    CustomerSearchBasic customerSearchBasic = new CustomerSearchBasic();
    customerSearchBasic.setEntityId(new SearchStringField("My Customer", SearchStringFieldOpera
tor.is));
    cs.setBasic(contactSearchBasic);
    cs.setCustomerJoin(customerSearchBasic);
    sessMgr.getWrappedPort().search(cs, this);
}
```

SOAP

```
<search xmlns="urn:messages_2008_1.platform.webservices.netsuite.com">
       <searchRecord xsi:type="ns4:ContactSearch"
  xmlns:ns4="urn:relationships_2008_1.lists.webservices.netsuite.com">
               <ns4:basic xsi:type="ns5:ContactSearchBasic"
  xmlns:ns5="urn:common_2008_1.platform.webservices.netsuite.com">
                  <ns5:email operator="is" xsi:type="ns6:SearchStringField"
  xmlns:ns6="urn:core_2008_1.platform.webservices.netsuite.com">
```

```
                  <ns6:searchValue xsi:type="xsd:string">contact@email.com</ns6:searchValue>

                </ns5:email>
              </ns4:basic>
              <ns4:customerJoin xsi:type="ns7:CustomerSearchBasic"
  xmlns:ns7="urn:common_2008_1.platform.webservices.netsuite.com">
                <ns7:entityId operator="is" xsi:type="ns8:SearchStringField"
  xmlns:ns8="urn:core_2008_1.platform.webservices.netsuite.com">
                  <ns8:searchValue xsi:type="xsd:string">My Customer</ns8:searchValue>
                </ns7:entityId>
              </ns4:customerJoin>
            </searchRecord>
          </search>
```

Sample 2 — Contact Search With Customer Join

The following sample shows how to return an associated joined list of records. In this case, all contacts associated with customers of internalId 1, 2 and 3 are returned.

Java

```
RecordRef[] rr = new RecordRef[]{new RecordRef("1", RecordType.customer),
new RecordRef("2", RecordType.customer), new RecordRef("3", RecordType.customer)};
CustomerSearchBasic customerSearchBasic = new CustomerSearchBasic();
customerSearchBasic.setInternalId(new SearchMultiSelectField(rr,
SearchMultiSelectFieldOperator.anyOf));
ContactSearch contactSearch = new ContactSearch();
contactSearch.setCustomerJoin(customerSearchBasic);
```

SOAP

```
<soapenv:Body>
 <search xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
 <searchRecord xsi:type="ns1:ContactSearch"
  xmlns:ns1="urn:relationships_2_5.lists.webservices.netsuite.com">
  <ns1:customerJoin xsi:type="ns2:CustomerSearchBasic"
   xmlns:ns2="urn:common_2_5.platform.webservices.netsuite.com">
   <ns2:internalId operator="anyOf" xsi:type="ns3:SearchMultiSelectField"
    xmlns:ns3="urn:core_2_5.platform.webservices.netsuite.com">
    <ns3:searchValue internalId="1" type="customer"
     xsi:type="ns3:RecordRef"/>
    <ns3:searchValue internalId="2" type="customer"
     xsi:type="ns3:RecordRef"/>
    <ns3:searchValue internalId="3" type="customer"
     xsi:type="ns3:RecordRef"/>
   </ns2:internalId>
  </ns1:customerJoin>
 </searchRecord>
 </search>
</soapenv:Body>
```

Sample 3 — Item Search With Pricing Join

The following sample shows how to search for all items that have a price level of 10.00.

C#

```
private void myItemSearch()
{

 ItemSearchBasic myItemSearchBasic = new ItemSearchBasic();

 SearchEnumMultiSelectField myEnum = new SearchEnumMultiSelectField();
 myEnum.@operator = SearchEnumMultiSelectFieldOperator.anyOf;
 myEnum.operatorSpecified = true;
 String[] searchStringArray = new String[1];
 searchStringArray[0] = "_inventoryItem";
 myEnum.operatorSpecified = true;
 myEnum.searchValue = searchStringArray;

 myItemSearchBasic.type = myEnum;


 PricingSearchBasic myPricingSearchBasic = new PricingSearchBasic();

 SearchDoubleField rateValue = new SearchDoubleField();
 rateValue.@operator = SearchDoubleFieldOperator.equalTo;
 rateValue.operatorSpecified = true;
 rateValue.searchValue = 10.00;
 rateValue.searchValueSpecified = true;

 myPricingSearchBasic.rate = rateValue;


 ItemSearch myItemSearch = new ItemSearch();
 myItemSearch.basic = myItemSearchBasic;
 myItemSearch.pricingJoin = myPricingSearchBasic;

 SearchResult searchResult = _service.search(myItemSearch);
}
```

SOAP Request

```
<soap:Body>
 <search xmlns="urn:messages_2014_1.platform.webservices.netsuite.com">
  <searchRecord xmlns:q1="urn:accounting_2014_1.lists.webservices.netsuite.com" xsi:type="q1:It
emSearch">
   <q1:basic>
    <type operator="anyOf" xmlns="urn:common_2014_1.platform.webservices.netsuite.com">
     <searchValue xmlns="urn:core_2014_1.platform.webservices.netsuite.com">_inventoryItem</sea
rchValue>
    </type>
   </q1:basic>
    <q1:pricingJoin>
    <rate operator="equalTo" xmlns="urn:common_2014_1.platform.webservices.netsuite.com">
     <searchValue xmlns="urn:core_2014_1.platform.webservices.netsuite.com">10</searchValue>
    </rate>
   </q1:pricingJoin>
  </searchRecord>
 </search>
</soap:Body>
```

Advanced Search Code Samples

The following advanced search samples are provided.

1. Perform a search that includes a saved search ID.
2. Perform a search with a defined page size.
3. Perform a search in which you override search columns in a saved search.
4. Perform a search in which you specify additional saved search criteria.
5. Programmatically manipulate data returned by a search

Perform a search that includes a saved search ID.

C#

```
// Create a service
NetSuiteService nss = new NetSuiteService();

// Perform the search. Note that you can get the saved search ID using
// either getSavedSearch() or through the UI
TransactionSearchAdvanced tsa1 = new TransactionSearchAdvanced();
tsa1.savedSearchId="57";  //substitute your own saved search internal ID
nss.search(tsa1);
```

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" savedSearchId="57" />
</search>
```

SOAP Response

```
// The default is to return search rows
<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:account>
<platformCore:searchValue internalId="125"/>
</platformCommon:account>
<platformCommon:amount>
<platformCore:searchValue>12481.9</platformCore:searchValue>
</platformCommon:amount>
<platformCommon:entity>
<platformCore:searchValue internalId="78"/>
</platformCommon:entity>
<platformCommon:mainline>
<platformCore:searchValue>false</platformCore:searchValue>
</platformCommon:mainline>
<platformCommon:number>
<platformCore:searchValue>102</platformCore:searchValue>
</platformCommon:number>
```

```
<platformCommon:postingPeriod>
<platformCore:searchValue internalId="72"/>
</platformCommon:postingPeriod>
<platformCommon:tranDate>
<platformCore:searchValue>2005-02-10T00:00:00.000-08:00</platformCore:searchValue>
</platformCommon:tranDate>
<platformCommon:tranId>
<platformCore:searchValue>102</platformCore:searchValue>
</platformCommon:tranId>
<platformCommon:type>
<platformCore:searchValue internalId="SalesOrd"/>
</platformCommon:type>
</tranSales:basic>
</platformCore:searchRow>
```

Perform a search with a defined page size.

This sample performs the same search as the previous sample, however, you are specifying the number of results that are returned per page.

C#

```
TransactionSearchAdvanced tsa2 = new TransactionSearchAdvanced();
tsa2.savedSearchId="57";  //substitute your own saved search internal ID

// Set search preference to return search columns
SearchPreferences sp = new SearchPreferences();
sp.pageSizeSpecified = true;
sp.pageSize = 10;
nss.searchPreferences = sp;
nss.search(tsa2);
```

Perform a search in which you override search columns in a saved search.

In this sample you are returning a specific saved search (57), and then overriding the search return columns in the saved search and specifying *your own* search return columns.

C#

```
TransactionSearchAdvanced tsa3 = new TransactionSearchAdvanced();
tsa3.savedSearchId="57";  //substitute your own saved search internal ID

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

// Instantiate SearchColumn 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;

SearchColumnDateField [] tranDateCols = new SearchColumnDateField[1];
SearchColumnDateField tranDateCol = new SearchColumnDateField();
tranDateCols[0] = tranDateCol;
tsrb.tranDate = tranDateCols;
SearchColumnBooleanField [] isFulfilledCols = new SearchColumnBooleanField[1];
SearchColumnBooleanField isFulfilledCol = new SearchColumnBooleanField();
isFulfilledCol.customLabel = "Order Fulfilled"; // Define a custom label
isFulfilledCols[0] = isFulfilledCol;
tsrb.shipRecvStatus = isFulfilledCols;

tsr.basic = tsrb;
tsa3.columns = tsr;
nss.search(tsa3);
```

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" savedSearchId="57">
                  <q1:columns>
                     <q1:basic>
                        <internalId xmlns="urn:common_2008_2.platform.webservices.netsuite.com"
>
                           <customLabel xmlns="urn:core_2008_2.platform.webservices.netsuite.co
m">Sales Order ID
   </customLabel>
                        </internalId>
                        <shipRecvStatus xmlns="urn:common_2008_2.platform.webservices.netsuite.
com">
                           <customLabel xmlns="urn:core_2008_2.platform.webservices.netsuite.co
m">Order Fulfilled
   </customLabel>
                        </shipRecvStatus>
                        <tranDate xmlns="urn:common_2008_2.platform.webservices.netsuite.com" /
>
                     </q1:basic>
                  </q1:columns>
               </searchRecord>
            </search>
```

SOAP Response

```
<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="117"/>
                    <platformCore:customLabel>Sales Order ID</platformCore:customLabel>
                    </platformCommon:internalId>
                    <platformCommon:tranDate>
                    <platformCore:searchValue>2005-04-23T00:00:00.000-07:00</platformCore:searc
hValue>
                    </platformCommon:tranDate>
```

```
                    </tranSales:basic>
                    </platformCore:searchRow>
```

Perform a search in which you specify additional saved search criteria.

The next sample shows how to take a saved search and specify additional search filter criteria. The additional search critera will *not* override the criteria already defined in the saved search.

C#

```
TransactionSearchAdvanced tsa4 = new TransactionSearchAdvanced();
tsa4.savedSearchId="57";

TransactionSearch ts = new TransactionSearch();
TransactionSearchBasic tsb = new TransactionSearchBasic();
// condition 1: 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;

// condition 2: tranId contains 182
SearchStringField sfTranId = new SearchStringField();
sfTranId.searchValue = "182"; // tranId contains 11
sfTranId.@operator = SearchStringFieldOperator.contains;
sfTranId.operatorSpecified = true;
tsb.tranId=sfTranId;

// condition 3: Is MultiShipping Routes enabled
SearchBooleanField sbfShipLineEnabled = new SearchBooleanField();
sbfShipLineEnabled.searchValue = true;
sbfShipLineEnabled.searchValueSpecified = true;
tsb.shipping = sbfShipLineEnabled;

// condition 4: Open SO
SearchBooleanField sbfTranStatus = new SearchBooleanField();
sbfTranStatus.searchValue = true;
sbfTranStatus.searchValueSpecified = true;
tsb.shipRecvStatusLine = sbfTranStatus;

ts.basic = tsb;
tsa4.criteria = ts;
tsa4.columns = tsr; //note - columns previously defined above.
nss.search(tsa4);
```

SOAP Request

```
<searchRecord xmlns:q1="urn:sales_2008_2.transactions.webservices.netsuite.com" xsi:type="q1:Tr
ansactionSearchAdvanced">
               <q1:criteria>
                  <q1:basic>
                     <tranId operator="contains" xmlns="urn:common_2008_2.platform.webservices.
```

```
netsuite.com">
                        <searchValue xmlns="urn:core_2008_2.platform.webservices.netsuite.com">
182</searchValue>
                     </tranId>
                     <type operator="anyOf" xmlns="urn:common_2008_2.platform.webservices.netsu
ite.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>
                     <shipRecvStatus xmlns="urn:common_2008_2.platform.webservices.netsuite.com
">
                        <customLabel xmlns="urn:core_2008_2.platform.webservices.netsuite.com">
Order Fulfilled
   </customLabel>
                     </shipRecvStatus>
                     <shipRecvStatusLine xmlns="urn:common_2008_2.platform.webservices.netsuite
.com">
                        <customLabel xmlns="urn:core_2008_2.platform.webservices.netsuite.com">

   Order Line 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>
```

SOAP Request (with Shipping and ShipRecvStatusLine criteria)

```
<searchRecord xmlns:q1="urn:sales_2008_2.transactions.webservices.netsuite.com"
 xsi:type="q1:TransactionSearchAdvanced">
               <q1:criteria>
                  <q1:basic>
                     <shipRecvStatus xmlns="urn:common_2008_2.platform.webservices.netsuite.com
">
                        <searchValue xmlns="urn:core_2008_2.platform.webservices.netsuite.com">
true</searchValue>
                     </shipRecvStatus>
                     <tranId operator="contains" xmlns="urn:common_2008_2.platform.webservices.
netsuite.com">
                        <searchValue xmlns="urn:core_2008_2.platform.webservices.netsuite.com">
11</searchValue>
                     </tranId>
                     <type operator="anyOf" xmlns="urn:common_2008_2.platform.webservices.netsu
ite.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>
                     <shipRecvStatus xmlns="urn:common_2008_2.platform.webservices.netsuite.com
">
                        <customLabel xmlns="urn:core_2008_2.platform.webservices.netsuite.com">
Order Fulfilled
   </customLabel>
                     </shipRecvStatus>
                     <tranDate xmlns="urn:common_2008_2.platform.webservices.netsuite.com" />
                  </q1:basic>
               </q1:columns>
            </searchRecord>
```

SOAP Response

```
 <platformCore:searchRow xsi:type="tranSales:TransactionSearchRow" xmlns:tranSales="urn:sales_2
008_2.transactions.webservices.netsuite.com">
    <tranSales:basic xmlns:platformCommon="urn:common_2008_2.platform.webservices.netsuite.com"
>
                        <platformCommon:internalId>
                           <platformCore:searchValue internalId="986"/>
                           <platformCore:customLabel>Sales Order ID</platformCore:customLabel>
                        </platformCommon:internalId>
                        <platformCommon:shipRecvStatusLine>
                           <platformCore:searchValue>true</platformCore:searchValue>
                           <platformCore:customLabel>Order Line Fulfilled</platformCore:customL
abel>
                        </platformCommon:shipRecvStatusLine>
                        <platformCommon:tranDate>
                           <platformCore:searchValue>2008-09-11T00:00:00.000-07:00</platformCor
e:searchValue>
                        </platformCommon:tranDate>
                     </tranSales:basic>
                  </platformCore:searchRow>
```

Programmatically manipulate data returned by a search

Java

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

        CustomerSearchAdvanced searchRecord = new CustomerSearchAdvanced();
        searchRecord.setSavedSearchId("26"); // A saved customer search

        SearchResult result = _port.search(searchRecord);

        if( result.getTotalRecords() > 0 ) {
                // retain the search ID in order to get more pages
                String sSearchId = result.getSearchId();
```

```
                SearchRowList rowList = result.getSearchRowList();

                processRowList(rowList);

                int iNumPages = result.getTotalPages();

                for ( int i=2; i<=iNumPages; i++) {
                        result = _port.searchMoreWithId(sSearchId, i);
                        rowList = result.getSearchRowList();

                        processRowList(rowList);
                }
                }
}

public void processRowList(SearchRowList rowList) {
        for (int i=0; i<rowList.getSearchRow().length; i++) {
                CustomerSearchRow row = (CustomerSearchRow) rowList.getSearchRow(i);
                CustomerSearchRowBasic basic = row.getBasic();

                SearchColumnStringField companyName = basic.getCompanyName(0);

                _console.write("Company Name: "+companyName.getSearchValue());
                if (basic.getEmail(0).getSearchValue() != null) {
                        String email = basic.getEmail(0).getSearchValue();
                        _console.write("\tEmail: "+email);
                }
                if (basic.getPhone(0).getSearchValue() != null) {
                        String phone = basic.getPhone(0).getSearchValue();
                        _console.write("\tPhone: "+phone);
                }
                _console.writeLn("\n");
        }
}
```

Searching for a Multi-select Custom Field

In the following code sample, the results for the custom transaction field *custcolcolumnname* are returned.

Java

```
 // transaction search by custom column field
 TransactionSearchBasic transactionSearch = new TransactionSearchBasic();

 SearchCustomFieldList searchCustomFieldList = new SearchCustomFieldList();
 transactionSearch.setCustomFieldList(searchCustomFieldList);

 // make the multiselectsearch
 SearchMultiSelectCustomField searchMultiSelectCustomField = new SearchMultiSelectCustomField()
;

 ListOrRecordRef listOrRecordRef = new ListOrRecordRef();
 listOrRecordRef.setInternalId("3"); // the internal id of the custom list value
 listOrRecordRef.setType("1"); // your custom list typeId
```

```
 searchCustomFieldList.setCustomField(new SearchCustomField[]{searchMultiSelectCustomField});

 // make the search expression
 searchMultiSelectCustomField.setInternalId("custcolcolumnname"); //the name of the tx custom c
olumn
 searchMultiSelectCustomField.setOperator(SearchMultiSelectFieldOperator.anyOf);

 searchMultiSelectCustomField.setSearchValue(new ListOrRecordRef[] {listOrRecordRef});

 SearchResult sr = _port.search(transactionSearch);
```

C#

```
private void searchForMultiSelectCustomField()

        {
            if (_isAuthenticated)
            {
                _out.info("\nExecuting  search ..... \n");
                // transaction search by custom column field
                TransactionSearch transactionSearch = new TransactionSearch();
                TransactionSearchBasic transactionSearchBasic = new TransactionSearchBasic();

                //Java - the SearchCustomFieldList is not used.
                //SearchCustomFieldList searchCustomFieldList = new SearchCustomFieldList();
                //transactionSearch.setCustomFieldList(searchCustomFieldList);
                SearchMultiSelectCustomField searchMultiSelectCustomField = new
          SearchMultiSelectCustomField();

                // make the search expression
               //the name of the transaction custom column
       searchMultiSelectCustomField.internalId = "custbody_multi_select";
                searchMultiSelectCustomField.@operator = SearchMultiSelectFieldOperator.anyOf;
                searchMultiSelectCustomField.operatorSpecified = true;
                //custom list called colors with typei id  1, values blue - internalid 1, green
 - id2 etc
                //we are looking for transactions which have transaction body field
        //of type multi select set to color blue
                ListOrRecordRef listOrRecordRef = new ListOrRecordRef();
                listOrRecordRef.internalId = "3";
                listOrRecordRef.typeId = "1";

                searchMultiSelectCustomField.searchValue = new ListOrRecordRef[] { listOrRecord
Ref };
                SearchCustomField[] searchCustomFieldList = new SearchCustomField[] {
  searchMultiSelectCustomField };

                //Java
                //searchCustomFieldList.setCustomField(new SearchCustomField[]{searchMultiSelec
tCustomField});
                transactionSearchBasic.customFieldList = searchCustomFieldList;
                transactionSearch.basic = transactionSearchBasic;
                SearchResult searchRes = _service.search(transactionSearch);
                _out.info("\nSearch Result contains " + searchRes.totalRecords + " record(s) \n
");
            }
            else
            {
```

```
            _out.info(
                "\nCannot call  search operation  because there is no active session. " +
                "You must be first logged on before attempting to call saved search.\n");
        }
    }
```

searchMore

The seachMore operation is used to retrieve more records after an initial search operation.

Important: Users who authenticate to NetSuite through login can use either searchMore *or* searchMoreWithId to paginate through search results. Users who authenticate to NetSuite by providing user credentials in the header of their SOAP requests can use **only** searchMoreWithId to paginate through search results, since searchMore requires an active session.

Note: (See Authentication Using Request Level Credentials for information on request-level-credential authentication.)

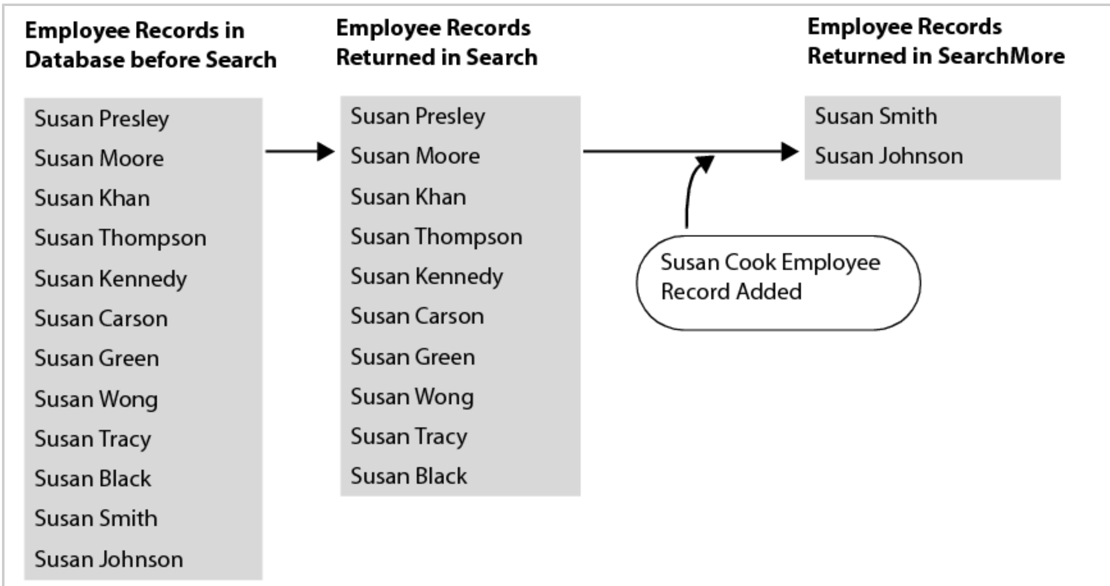
The results returned in a searchMore operation reflect the records in the next segment as defined in the **original** search operation. Therefore, if a record is deleted before all records have been returned, the total number of records may differ from the total record count returned in the search operation — the deleted record is not returned. If a record is **added**, that record is not returned in subsequent searchMore operations. However, the data in each record returned is current such that if a change to a record occurs in between the time of the original search operation and the searchMore operation, the updated data is returned.

Example

Suppose you submit the following request for all employees whose firstName contains ‘Susan' and set the pagesize to 10 records:

```
<soap:Header>
 <platformMsgs:searchPreferences>
  <platformMsgs:bodyFieldsOnly>true</platformMsgs:bodyFieldsOnly>
  <platformMsgs:pageSize>10</platformMsgs:pageSize>
 </platformMsgs:searchPreferences>
</soap:Header>
<soap:Body>
 <platformMsgs:search>
  <platformMsgs:searchRecord xsi:type="listEmp:EmployeeSearch">
   <listEmp:firstName operator="contains" xsi:type="platformCore:SearchStringField">
    <platformCore:searchValue>Susan</platformCore:searchValue>
   </listEmp:firstName>
  </platformMsgs:searchRecord>
 </platformMsgs:search>
</soap:Body>
```

As illustrated in the following figure, if the database contains 12 matches at the time of the initial Search, the Search result would provide the first 10 records as defined in the pageSize preference. A subsequent SearchMore operation would return only the remaining two records from the original search result even if another employee record, Susan Cook, is added prior to the SearchMore request being submitted.



Request

The SearchMoreRequest type is used for the request. It contains the following fields.

Element Name	XSD Type	Notes
pageIndex	SearchRecord	An index that specifies which page in the search to return. If it is not provided, the next page is returned.

Response

The SearchMoreResponse type is used for the response. It contains the following fields.

Element Name	XSD Type	Notes
status	Status	The status for this search. All applicable errors or warnings will be listed within this type.
totalRecords	xsd:int	The total number of records for this search. Depending on the pageSize value, some or all the records may be returned in this response
pageSize	xsd:int	The page size for this search.
totalPages	xsd:int	The total number of pages that are part of this search.
pageIndex	xsd:int	The page index for the current set of results.

Element Name	XSD Type	Notes
recordList	Record[]	A list of records that meet the criteria for this search. The actual records returned need to be of a type that extends the abstract type of record.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

SOAP Request

In this example, page 2 of the search result set is requested.

```
<soap:Body>
<platformMsgs:searchMore>
 <platformMsgs:pageIndex>2</platformMsgs:pageIndex>
</platformMsgs:searchMore>
</soap:Body>
```

SOAP Response

```
<soapenv:Body>
<searchMoreResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
<searchResult xmlns="urn:core_2_5.platform.webservices.netsuite.com">
<status isSuccess="true"/>
<totalRecords>93</totalRecords>
<pageSize>10</pageSize>
<totalPages>10</totalPages>
<pageIndex>2</pageIndex>
<recordList>
 <record internalId="80" xsi:type="ns1:Customer"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:ns1="urn:relationships_2_5.lists.webservices.netsuite.com">
  <ns1:entityId>Jackson Alexander</ns1:entityId>
  <ns1:isInactive>false</ns1:isInactive>
  .
```

```
  ...[more fields]
  .
  <ns1:customFieldList>
   <customField internalId="custentity_map" xsi:type="StringCustomFieldRef">
    <value>http://maps.google.com</value>
   </customField>
  </ns1:customFieldList>
 </record>
 <record internalId="227" xsi:type="ns2:Customer"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:ns2="urn:relationships_2_5.lists.webservices.netsuite.com">
  <ns2:entityId>Seena Thomas</ns2:entityId>
  <ns2:isInactive>false</ns2:isInactive>
  <ns2:companyName>Seena Thom Inc.</ns2:companyName>
  .
  ...[more fields]
  .
  <ns2:customFieldList>
   <customField internalId="custentity_map" xsi:type="StringCustomFieldRef">
    <value>http://maps.google.com</value>
   </customField>
  </ns2:customFieldList>
 </record>
 .
 ...[more records]
 .
</recordList>
</searchResult>
</searchMoreResponse>
</soapenv:Body>
```

C#

```
private void searchMore( SearchResult response )
{
 //SearchResponse response;
 bool isGetAllPages = false;

 // Keep getting pages until there are no more pages to get
 while ( response.totalRecords > (response.pageSize * response.pageIndex) )
 {
  if ( !isGetAllPages )
  {
   // Only continue if user wants to get the next page
   _out.write( "\nThere are more search results. Would you like to get the next page for
   this search? (A/Y/N): " );
   String userResponse = _out.readLn().ToUpper();
   if ( String.Equals( userResponse, "N" ) )
   {
    break;
   }
   else if ( String.Equals( userResponse, "A" ) )
   {
    isGetAllPages = true;
   }
  }

  // Invoke searchMore() operation
  response = _service.searchMore( response.pageIndex + 1 );
```

```
  // Process response
  if ( response.status.isSuccess )
  {
   processCustomerSearchResponse( response );
  }
  else
  {
   _out.error( getStatusDetails( response.status ) );
  }
 }
}
```

Java

```
public void searchMore(SearchResult result) throws RemoteException,
ExceededUsageLimitFault, UnexpectedErrorFault, InvalidSessionFault,
ExceededRecordCountFault {
 // SearchResult response;
 boolean isGetAllPages = false;

 // Keep getting pages until there are no more pages to get
 while (result.getPageSize() != null
 && result.getPageIndex() != null
 && (result.getTotalRecords().intValue() > (result.getPageSize()
 .intValue() * result.getPageIndex().intValue()))) {
  if (!isGetAllPages) {
   // Only continue if user wants to get the next page
   _console
   .write("\nThere are more search results. Would you like to get the next page for
    this search? (A/Y/N):");
   String userResponse = _console.readLn().toUpperCase();
   if ("N".equals(userResponse)) {
    break;
   } else if ("A".equals(userResponse)) {
    isGetAllPages = true;
   }
  }

  // Invoke searchMore() operation
  result = _port.searchMore(result.getPageIndex().intValue() + 1);

  // Process result
  if (result.getStatus().isIsSuccess()) {
   processCustomerSearchResponse(result);
  } else {
   _console.error(getStatusDetails(result.getStatus()));
  }
 }
}
```

searchMoreWithId

Users who authenticate to NetSuite by providing their credentials in the SOAP header of their requests **must** use *searchMoreWithId* to retrieve search results that span multiple pages. They cannot use searchMore or searchNext, since both operations require an active session. (For information on request-level-credential authentication, see Authentication Using Request Level Credentials.)

Note: Users who authenticate to NetSuite through login can use searchMore *or* searchMoreWithId to paginate through search results.

The searchMoreWithId operation allows users to reference a specific search result set by its *searchId*, a parameter included in all search results. As with searchMore, users must set the *pageIndex* value to specify which page in the search to return.

Note that a given entity is allowed to have two search IDs alive at a time. The oldest ID is expunged if a third is created. For SuiteCloud Plus users, a maximum of 20 search IDs are stored for a single SuiteCloud Plus user (two IDs per session x ten current logins). (For information on the SuiteCloud Plus license, see Enabling Web Services Concurrent Users with SuiteCloud Plus.)

Search IDs expire if they have not been used within 15 minutes after their creation. Passing an expired or invalid *searchId* will return search results with a “failed” status and StatusDetailCode=INVALID_JOB_ID.

Note: There is no async equivalent for this operation.

Request

The SearchMoreWithIdRequest type is used for the request. It contains the following fields.

Element Name	XSD Type	Notes
searchId	string	The search result ID.
pageIndex	int	An index that specifies which page in the search to return.

Response

The SearchMoreWithIdResponse type is used for the response. It contains the following fields.

Element Name	XSD Type	Notes
status	Status	The status for this search. All applicable errors or warnings will be listed within this type.
totalRecords	xsd:int	The total number of records for this search. Depending on the pageSize value, some or all the records may be returned in this response
pageSize	xsd:int	The page size for this search.
totalPages	xsd:int	The total number of pages that are part of this search.

Element Name	XSD Type	Notes
pageIndex	xsd:int	The page index for the current set of results.
searchId	string	Returns a specific search based on its search ID.
recordList	Record[]	A list of records that meet the criteria for this search. The actual records returned need to be of a type that extends the abstract type of record.
searchRowList	SearchRowList	A list of return columns that meet the criteria for this search.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

The following samples show the SOAP for the initial search as well as the SOAP for subsequent searchMoreWithId operation used to specify the next page of the search. The ID returned in the initial search is specified as the *searchId* when executing searchMoreWithId.

Note: Prefix-to-namespace mappings have been omitted for readability.

SOAP Request for Initial Search

```
<soapenv:Envelope>
    <soapenv:Header>
        <platformMsgs:searchPreferences>
            <platformMsgs:bodyFieldsOnly>true</platformMsgs:bodyFieldsOnly>
            <platformMsgs:pageSize>100</platformMsgs:pageSize>
        </platformMsgs:searchPreferences>
        <platformMsgs:passport>
            <platformCore:email>jdoe@netsuite.com</platformCore:email>
            <platformCore:password>mypassword</platformCore:password>
            <platformCore:account>000034</platformCore:account>
            <platformCore:role internalId="3"/>
```

```
        </platformMsgs:passport>
    </soapenv:Header>
    <soapenv:Body>
        <platformMsgs:search>
            <searchRecord xsi:type="platformCommon:ContactSearchBasic">
                <platformCommon:city operator="is" xsi:type="platformCore:SearchStringField">
                    <platformCore:searchValue xsi:type="xsd:string">San Francisco</platformCore
:searchValue>
                </platformCommon:city>
            </searchRecord>
        </platformMsgs:search>
    </soapenv:Body>
</soapenv:Envelope>
```

SOAP Response for Initial Search

```
<soapenv:Envelope>
    <soapenv:Header>
        <platformMsgs:documentInfo>
            <platformMsgs:nsId>WEBSERVICES_528736_070120085439950063070492 33_d53ef4d2273b15<
  platformMsgs:nsId>
        </platformMsgs:documentInfo>
    </soapenv:Header>
    <soapenv:Body>
        <platformMsgs:searchResponse>
          <platformCore:searchResult>
          <platformCore:status isSuccess="true"/>
          <platformCore:totalRecords>231</platformCore:totalRecords>
          <platformCore:pageSize>100</platformCore:pageSize>
          <platformCore:totalPages>3</platformCore:totalPages>
          <platformCore:pageIndex>1</platformCore:pageIndex>
            <platformCore: searchId > WEBSERVICES_528736_070120085439950063070492 33_d53ef4d2273
b15
  <platformCore: searchId >
           <platformCore:recordList>
           <platformCore:recordList>
                 <platformCore:record internalId="4" externalId="entity-4" xsi:type="listRel:Co
ntact">
                      <listRel:entityId>john</listRel:entityId>
                      <listRel:firstName>John</listRel:firstName>
                        .....
                    </platformCore:record>
                </platformCore:recordList>
            </platformCore:searchResult>
        </searchMoreWithIdResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

SOAP Request for Getting the Next Page Using searchMoreWithId

```
<soapenv:Envelope>
    <soapenv:Header>
        <platformMsgs:searchPreferences>
            <platformMsgs:bodyFieldsOnly>true</platformMsgs:bodyFieldsOnly>
```

```
            <platformMsgs:pageSize>100</platformMsgs:pageSize>
        </platformMsgs:searchPreferences>
        <platformMsgs:passport>
            <platformCore:email>jdoe@netsuite.com</platformCore:email>
            <platformCore:password>mypassword</platformCore:password>
            <platformCore:account>000034</platformCore:account>
            <platformCore:role internalId="3"/>
        </platformMsgs:passport>
    </soapenv:Header>
    <soapenv:Body>
        <platformMsgs:searchMoreWithId>
            <searchId>WEBSERVICES_528736_070120085439950063070492333_d53ef4d2273b15</searchId>
            <pageIndex>2</pageIndex>
        </platformMsgs:searchMoreWithId>
    </soapenv:Body>
</soapenv:Envelope>
```

SOAP Response for Getting the Next Page Using searchMoreWithId

```
<soapenv:Envelope>
    <soapenv:Header>
        <platformMsgs:documentInfo>
            <platformMsgs:nsId>WEBSERVICES_528736_070120085439950063070492333_d53ef4d2273b15</pl
atformMsgs:nsId>
        </platformMsgs:documentInfo>
    </soapenv:Header>
    <soapenv:Body>
        <platformMsgs:searchMoreWithIdResponse>
            <platformCore:searchResult>
         <platformCore:status isSuccess="true"/>
         <platformCore:totalRecords>231</platformCore:totalRecords>
         <platformCore:pageSize>100</platformCore:pageSize>
         <platformCore:totalPages>3</platformCore:totalPages>
         <platformCore:pageIndex>1</platformCore:pageIndex>
        <platformCore: searchId > WEBSERVICES_528736_070120085439950063070492333_d53ef4d2273b15
             </platformCore: searchId >
                <platformCore:recordList>
                    <platformCore:record internalId="5" externalId="entity-5" xsi:type="listRel
:Contact">
                        <listRel:entityId>mike</listRel:entityId>
                        <listRel:firstName>Mike</listRel:firstName>
                        .....
                    </platformCore:record>
                </platformCore:recordList>
            </platformCore:searchResult>
        </platformMsgs:searchMoreWithIdResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

Java

```
public List<Contact> searchContactsByCity(NetSuitePortType port, String city) throws Exception
{
    SearchResult searchResult = null;         // search result for each page
    Record [] contacts = null;                    // contacts for each page
```

```
    List<Contact> consolidatedResults = new ArrayList<Contact>();   // to return

    // build search criteria
    ContactSearchBasic contactSearch = new ContactSearchBasic();
    contactSearch.setCity(new SearchStringField(city, SearchStringFieldOperator.is));

    // execute the search
    searchResult = port.search(contactSearch);

    if( searchResult.getTotalRecords() > 0 )
    {
        // retain the search ID in order to get more pages
        String sSearchId = searchResult.getSearchId();

        // process first page
        contacts = searchResult.getRecordList().getRecord();
        for (Record contact : contacts)
            consolidatedResults.add((Contact)contact);

        // process remaining pages using search ID
        int iNumPages = searchResult.getTotalPages();
        for ( int i=2; i<=iNumPages; i++)
        {
            // get the page
            searchResult = port.searchMoreWithId(sSearchId, i);

            // process the page
            contacts = searchResult.getRecordList().getRecord();
            for (Record contact : contacts)
                consolidatedResults.add((Contact)contact);
        }
    }
    return consolidatedResults;
}
```

searchNext

The searchNext operation is used to retrieve the next set of records after an initial search operation.

Important: Users who authenticate to NetSuite through login can use either searchNext *or* searchMoreWithId to paginate through search results. Users who authenticate to NetSuite by providing user credentials in the header of their SOAP requests can use **only** searchMoreWithId to paginate through search results, since searchNext requires an active session.

Note: (See Authentication Using Request Level Credentials for information on request-level-credential authentication.)

Request

The SearchNextRequest type is used for the request. It does not contain any fields.

Response

The SearchNextResponse type is used for the response. It contains the following fields.

Element Name	XSD Type	Notes
status	Status	The status for this search. All applicable errors or warnings will be listed within this type.
totalRecords	xsd:int	The total number of records for this search. Depending on the pageSize value, some or all the records may be returned in this response
pageSize	xsd:int	The page size for this search.
totalPages	xsd:int	The total number of pages that are part of this search.
pageIndex	xsd:int	The page index for the current set of results.
searchId	string	Returns a specific search based on its search ID.
recordList	Record[]	A list of records that meet the criteria for this search. The actual records returned need to be of a type that extends the abstract type of record.
searchRowList	SearchRowList	A list of return columns that meet the criteria for this search.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

SOAP Request

```
<soap:Body>
 <platformMsgs:searchNext/>
</soap:Body>
```

SOAP Response

```
<soapenv:Body>
<searchNextResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
<searchResult xmlns="urn:core_2_5.platform.webservices.netsuite.com">
<status isSuccess="true"/>
<totalRecords>93</totalRecords>
<pageSize>10</pageSize>
<totalPages>10</totalPages>
<pageIndex>3</pageIndex>
<recordList>
 <record internalId="865" xsi:type="ns1:Customer"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:ns1="urn:relationships_2_5.lists.webservices.netsuite.com">
  <ns1:entityId>John Bauer</ns1:entityId>
  <ns1:isInactive>false</ns1:isInactive>
  .
  ...[more fields]
  .
  <ns1:customFieldList>
   <customField internalId="custentity_map" xsi:type="StringCustomFieldRef">
    <value>http://maps.google.com</value>
   </customField>
  </ns1:customFieldList>
 </record>
 .
 ...[more records]
 .
</recordList>
</searchResult>
</searchNextResponse>
</soapenv:Body>
```

ssoLogin

The ssoLogin operation provides a mechanism for a partner application to log in on behalf of the user into NetSuite, without the user's credentials (such as the user's password) ever going through the partner servers, when the inbound single sign-on feature is enabled.

Users have to go through the following steps to set up inbound single sign-on. The external application must establish a single sign-on mapping for every user that accesses NetSuite through their site. The following steps outline the general workflow. For more details, see the help topic *Inbound Single Sign-on*.

1. User authenticates at the partner site.
2. Partner provides the single sign-on link to user.
3. Partner produces a token containing the company ID and user ID from the external system and the timestamp, that is encrypted using a private key, and redirects the user to NetSuite.
4. User logs in to NetSuite and the authentication link for the user is created.

5. After the single sign-on mapping is created, the partner invokes ssoLogin operation via Web services to access NetSuite on behalf of the user.

Important: Please note the important following details about this operation:

- Users who authenticate to NetSuite through the login operation can use ssoLogin. Users who authenticate to NetSuite by providing user credentials in the header of their SOAP requests cannot use this operation.
- Web services users who want to use inbound single sign-on need to purchase the "Single Sign On Configuration" line item. Contact your account manager for assistance.
- The SuiteTalk mapSso operation is not directly connected to the ssoLogin operation. The purpose of mapSso is to establish the mapping between a user's identity in an external application and the user's identity in NetSuite.
- The ssoLogin operation is for establishing **inbound** connections only, as described. To create an outbound connection that goes from NetSuite to a third-party application, see the help topic *SuiteSignOn (Outbound Single Sign-on)*.

Important: Since September 2012, when NetSuite began hosting new customer accounts in multiple data centers, the URL to be used when running Web services operations depends upon the data center hosting each NetSuite account. For example, it could be webservices.netsuite.com or webservices.na1.netsuite.com. In the 2012.2 and later Web services endpoints, you can use the getDataCenterUrls operation to dynamically discover the correct URL for Web services access to a NetSuite account. A REST service is also available for this purpose. See the help topic *Using the REST roles Service to Get User Accounts, Roles, and Domains*. For more information, see the help topic *Understanding Multiple Data Centers*.

Request

The SsoPassportRequest type is used for this request. It contains the following fields:

Element Name	XSD Type	Notes
SsoPassport	SsoPassport	Contains all the required credentials including username (email address), password, account, role, and authentication token, and partner Id to authenticate the user and create a new session.

The SsoPassport type includes the following elements:

- authenticationToken
- partnerId

Response

The SsoLoginResponse type is used for the response.

Element Name	XSD Type	Notes
status	Status	The status for this search. All applicable errors or warnings will be listed within this type.
wsRoleList	WsRoleList	

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidAccountFault
- InvalidCredentialsFault
- InvalidVersionFault
- ExceededRequestLimitFault
- UnexpectedErrorFault

Sample Code

SOAP Request

```
<ssoLogin xmlns="urn:messages_2012_1.platform.webservices.netsuite.com>
            <SsoPassport>
               <ns1:authenticationToken xmlns:ns1="urn:core_2012_1.platform.webservices.netsuit
e.com"><my
  token></ns1:authenticationToken>
               <ns2:partnerId xmlns:ns2="urn:core_2012_1.platform.webservices.netsuite.com">1</
ns2:partnerId>
            </SsoPassport>
         </ssoLogin>
```

Java

```
public void ssoLogin() throws Exception {
        SsoPassport sso = new SsoPassport();
        sso.setAuthenticationToken("<my token>");
        sso.setPartnerId("1");
        sessMgr.getPort().ssoLogin(sso);
     }
```

update

The update operation is used to update an instance of a record in NetSuite. It is similar to the updateList operation, which allows users to update more than one record at a time.

Only the fields that have been populated in each submitted record are updated in the system. If a field has NOT been populated, it is not updated in the system and it retains its previous value. If a field is set to an empty string, the previous value of the field is replaced with an empty string. Therefore, when updating records, it is recommended that you get the desired record, instantiate a new record of the same type, populate only the fields that require an update and then submit the updated record. This ensures that only the fields requiring an update are written on submission.

Important: Calculated and hidden fields in records are always updated by the system unless your service explicitly overrides the system values. For more information, see Hidden Fields. Also, custom fields can only be set to NULL by submitting the field in nullFieldList. For more information, see CustomFieldList.

To ensure that the most recent data for a given record is being modified, when a Web service request is received, the values for that record are retrieved at the time of the Update request rather than with the initial Get of the associated record. The record is then updated by the values submitted in the request. It is possible that between the time of the retrieval of the record field values and the submission of the updated fields that the record is altered from another source (for example from a UI submission). In this case an error message is returned to indicate that the fields have been modified since your service retrieved the record.

Although records of a particular type may be used in multiple integration scenarios, each record instance can only have a single external ID value. In order to maintain data integrity, only a single integrated application can set and update external ID values for each record type. External ID values for all records of a particular type must all come from the same external application.

Note: As of the 2011.2 endpoint, when a record is updated and the values that are sent are exactly the same as the existing record values, the record is not reset (nothing happens). In previous endpoints, sometimes records were reset in these cases.

Request

The UpdateRequest type is used for the request. It contains the following fields.

Element Name	XSD Type	Notes
record	Record	Contains an array of record objects. The record type is an abstract type so an instance of a type that extends record must be used—such as Customer or Event.

Response

The UpdateResponse type is used for the response. It contains the following fields.

Element Name	XSD Type	Notes
response	WriteResponse	Contains details on the status of the operation and a reference to the updated record.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

SOAP Request

In the following example, a customer's companyName is updated. The internal ID for the customer must be provided in the request.

```
<soap:Body>
<platformMsgs:update>
 <platformMsgs:record internalId="980" xsi:type="listRel:Customer">
  <listRel:companyName>Shutter Fly Corporation</listRel:companyName>
 </platformMsgs:record>
</platformMsgs:update>
</soap:Body>
```

SOAP Response

```
<soapenv:Body>
<updateResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
<writeResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
```

```
 <ns1:status isSuccess="true" xmlns:ns1="urn:core_2_5.platform.webservices.netsuite.com"/>
  <baseRef internalId="980" type="customer" xsi:type="ns2:RecordRef"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns2="urn:core_2_5.platform.webservices.netsuite.com"/>
</writeResponse>
</updateResponse>
</soapenv:Body>
```

C#

```
 private void updateCustomer()
{
 // This operation requires a valid session
 this.login( true );

 Customer customer = new Customer();

 // Get nsKey for update
 _out.write( "\nEnter nsKey for customer record to be updated : " );
 customer.internalId = _out.readLn().ToUpper();

 // Set name and email
 customer.entityId = "XYZ 2 Inc";
 customer.companyName = "XYZ 2, Inc.";
 customer.email = "bsanders@xyz.com";

 // Populate the address. Updating a list through WS results in the
 // entire contents of the previous list being replaced by the new
 // list.
 CustomerAddressbook address = new CustomerAddressbook();
 address.defaultBilling = true;
 address.defaultBillingSpecified = true;
 address.defaultBilling = false;
 address.defaultBillingSpecified = true;
 address.label = "Billing Address";
 address.addr1 = "4765 Sunset Blvd";
 address.city = "San Mateo";
 address.state = "CA";
 address.country = Country._unitedStates;

 // Attach the address to the customer
 CustomerAddressbookList addressList = new CustomerAddressbookList();
 CustomerAddressbook[] addresses = new CustomerAddressbook[1];
 addresses[0] = address;
 addressList.addressbook = addresses;
 customer.addressbookList = addressList;

 // Invoke add() operation
 WriteResponse response = _service.update( customer );

 // Process the response
 if ( response.status.isSuccess )
 {
  _out.info(
  "\nThe following customer was updated successfully:" +
  "\nkey=" + ((RecordRef) response.baseRef).internalId +
  "\nentityId=" + customer.entityId +
  "\ncompanyName=" + customer.companyName +
  "\nemail=" + customer.email +
```

```
  "\naddressbookList[0].label=" + customer.addressbookList.addressbook[0].label );
 }
 else
 {
  _out.error( getStatusDetails( response.status ) );
 }
}
```

Java

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

 Customer customer = new Customer();

 // Get nsKey for update
 _console.write("\nEnter nsKey for customer record to be updated : ");
 customer.setInternalId(_console.readLn().toUpperCase());

 // Set name and email
 customer.setEntityId("XYZ 2 Inc");
 customer.setCompanyName("XYZ 2, Inc.");
 customer.setEmail("bsanders@xyz.com");

 // Populate the address. Updating a list through WS results in the
 // entire contents of the previous list being replaced by the new
 // list.
 CustomerAddressbook address = new CustomerAddressbook();
 address.setDefaultBilling(Boolean.TRUE);
 address.setDefaultShipping(Boolean.FALSE);
 address.setLabel("Billing Address");
 address.setAddr1("4765 Sunset Blvd");
 address.setCity("San Mateo");
 address.setState("CA");
 address.setCountry(Country._unitedStates);

 // Attach the address to the customer
 CustomerAddressbookList addressList = new CustomerAddressbookList();
 CustomerAddressbook[] addresses = new CustomerAddressbook[1];
 addresses[0] = address;
 addressList.setAddressbook(addresses);
 customer.setAddressbookList(addressList);

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

 // Process the response
 if (response.getStatus().isIsSuccess()) {
  _console.info("\nThe following customer was updated successfully:"
  + "\nkey="
  + ((RecordRef) response.getBaseRef()).getInternalId()
  + "\nentityId="
  + customer.getEntityId()
  + "\ncompanyName="
  + customer.getCompanyName()
  + "\nemail="
```

```
  + customer.getEmail()
  + "\naddressbookList[0].label="
  + customer.getAddressbookList().getAddressbook(0)
  .getLabel());
 } else {
  _console.error(getStatusDetails(response.getStatus()));
 }
}
```

Updating Record Lists

When updating a list of records (a sublist) within a business record you CANNOT update a specific item in the list. Instead you must interact with the sublist as a whole. For details, see Working with Sublists in Web Services.

updateList

The updateList operation is used to update one or more instances of a record type in NetSuite.

If there are multiple records, they can either be of the same record type or different record types. For example, it's possible to update a customer and a contact within a single request using this operation.

Only the fields that have been populated in each submitted record are updated in the system. If a field has not been populated, it is not updated in the system and it retains its previous value. If a field is set to an empty string, the previous value of the field is replaced with an empty string.

Although records of a particular type may be used in multiple integration scenarios, each record instance can only have a single external ID value. In order to maintain data integrity, only a single integrated application can set and update external ID values for each record type. External ID values for all records of a particular type must all come from the same external application.

Important: Calculated and hidden fields in records are always updated by the system unless your service explicitly overrides the system values. For more information, refer to Hidden Fields.

Note: As of the 2011.2 endpoint, when a record is updated and the values that are sent are exactly the same as the existing record values, the record is not reset (nothing happens). In previous endpoints, sometimes records were reset in these cases.

Request

The UpdateListRequest type is used for the request. It contains the following fields.

Element Name	XSD Type	Notes
record[]	Record	Contains an array of record objects. The record type is an abstract type so an instance of a type that extends record must be used—such as Customer or Event.

Note: An asynchronous equivalent is available for this operation, **asyncUpdateList**. For information about asynchronous request processing, see Synchronous Versus Asynchronous Request Processing.

Response

The UpdateListResponse type is used for the response. It contains the following fields.

Element Name	XSD Type	Notes
response[]	WriteResponse	Contains an array of WriteResponse objects, each of which contains details on the status of that update operation and a reference to the created record.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

SOAP Request

In the following example, two customer records are updated. The first has the single field companyName updated, while the second updates two fields, entityID and companyName. The internalID for each record must be provided and the type of record (Customer) to be updated.

```
<soap:Body>
<platformMsgs:updateList>
```

```
 <platformMsgs:record internalId="980" xsi:type="listRel:Customer">
  <listRel:companyName>Shutter Fly Corporation</listRel:companyName>
 </platformMsgs:record>
 <platformMsgs:record internalId="981" xsi:type="listRel:Customer">
  <listRel:entityId>GNCC</listRel:entityId>
  <listRel:companyName>GNCC Corp</listRel:companyName>
 </platformMsgs:record>
</platformMsgs:updateList>
</soap:Body>
```

SOAP Response

```
<soapenv:Body>
<updateListResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
<writeResponseList xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
 <writeResponse>
  <ns1:status isSuccess="true" xmlns:ns1="urn:core_2_5.platform.webservices.netsuite.com"/>
  <baseRef internalId="980" type="customer" xsi:type="ns2:RecordRef"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns2="urn:core_2_5.platform.webservices.netsuite.com"/>
 </writeResponse>
 <writeResponse>
  <ns3:status isSuccess="true" xmlns:ns3="urn:core_2_5.platform.webservices.netsuite.com"/>
  <baseRef internalId="981" type="customer" xsi:type="ns4:RecordRef"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns4="urn:core_2_5.platform.webservices.netsuite.com"/>
 </writeResponse>
</writeResponseList>
</updateListResponse>
</soapenv:Body>
```

C#

```
private void updateCustomerList()
{
 // This operation requires a valid session
 this.login( true );

 // Prompt for list of nsKeys and put in an array
 _out.write( "\nEnter nsKeys for customer records to be updated (separated by commas): " );
 String reqKeys = _out.readLn();
 string [] nsKeys = reqKeys.Split( new Char[] {','} );

 // Create an array of Record objects to hold the customers
 Record[] records = new Record[nsKeys.Length];

 // For each submitted nsKey, populate a customer object
 for ( int i=0; i<nsKeys.Length; i++)
 {
  Customer customer = new Customer();

  // Update name
  customer.entityId = "XYZ Inc " + i;
  customer.companyName = "XYZ, Inc. " + i;

  customer.internalId = nsKeys[i].Trim();
  records[i] = customer;
```

```
 }

 // Invoke updateList() operation to update customers
 WriteResponse[] responses = _service.updateList( records );

 // Process responses for all successful updates
 _out.info( "\nThe following customers were updated successfully:" );
 bool hasFailures = false;
 for ( int i=0; i<responses.Length; i++ )
 {
  if ( (responses[i] != null) && (responses[i].status.isSuccess) )
  {
   _out.info( "\nCustomer[" + i + "]:" );
   _out.info(
   "key=" + ((RecordRef) responses[i].baseRef).internalId +
   "\nentityId=" + ((Customer) records[i]).entityId +
   "\ncompanyName=" + ((Customer) records[i]).companyName );
  }
  else
  {
   hasFailures = true;
  }
 }

 // Process responses for all unsuccessful updates
 if ( hasFailures )
 {
  _out.info( "\nThe following customers were not updated:\n" );
  for ( int i=0; i<responses.Length; i++ )
  {
   if ( (responses[i] != null) && (!responses[i].status.isSuccess) )
   {
    _out.info( "Customer[" + i + "]:" );
    _out.info( "key=" + ((RecordRef) responses[i].baseRef).internalId );
    _out.errorForRecord( getStatusDetails( responses[i].status ) );
   }
  }
 }
}
```

Java

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

 // Prompt for list of nsKeys and put in an array
 _console
 .write("\nEnter nsKeys for customer records to be updated (separated by commas): ");
 String reqKeys = _console.readLn();
 String[] nsKeys = reqKeys.split(",");

 // Create an array of Record objects to hold the customers
 Record[] records = new Record[nsKeys.length];

 // For each submitted nsKey, populate a customer object
 for (int i = 0; i < nsKeys.length; i++) {
```

```
  Customer customer = new Customer();

  // Update name
  customer.setEntityId("XYZ Inc " + i);
  customer.setCompanyName("XYZ, Inc. " + i);

  customer.setInternalId(nsKeys[i].trim());
  records[i] = customer;
 }

 // Invoke updateList() operation to update customers
 WriteResponseList responseList = _port.updateList(records);

 // Process responses for all successful updates
 WriteResponse[] responses = responseList.getWriteResponse();
 boolean hasFailures = false;
 _console.info("\nThe following customers were updated successfully:");
 for (int i = 0; i < responses.length; i++) {
  if ((responses[i] != null)
  && (responses[i].getStatus().isIsSuccess())) {
   _console.info("\nCustomer[" + i + "]:");
   _console.info("key="
   + ((RecordRef) responses[i].getBaseRef()).getInternalId()
   + "\nentityId="
   + ((Customer) records[i]).getEntityId()
   + "\ncompanyName="
   + ((Customer) records[i]).getCompanyName());
  } else {
   hasFailures = true;
  }
 }

 // Process responses for all unsuccessful updates
 if (hasFailures) {
  _console.info("\nThe following customers were not updated:\n");
  for (int i = 0; i < responses.length; i++) {
   if ((responses[i] != null)
   && (!responses[i].getStatus().isIsSuccess())) {
    _console.info("Customer[" + i + "]:");
    _console.info("key="
    + ((RecordRef) responses[i].getBaseRef()).getInternalId());
    _console.errorForRecord(getStatusDetails(responses[i]
    .getStatus()));
   }
  }
 }
}
```

updateInviteeStatus

The updateInviteeStatus operation lets users respond to NetSuite events that have been sent to them. This operation takes both the internal ID of the event as well as a calendar event status as arguments.

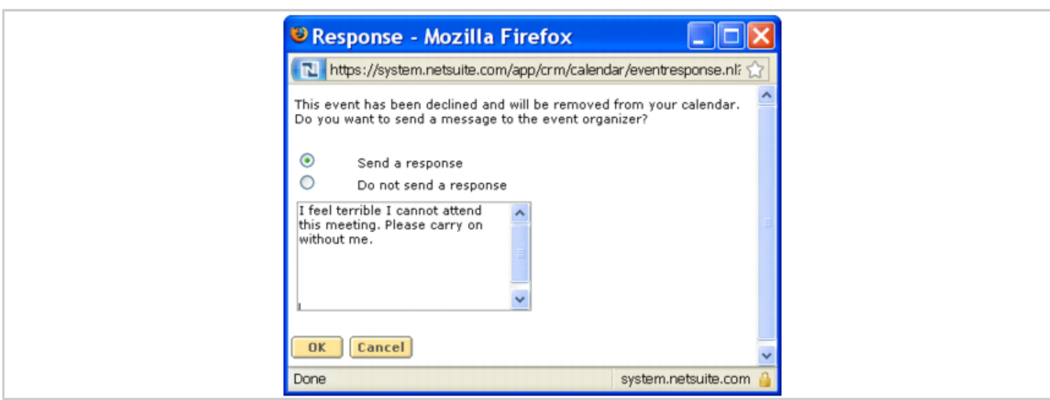
After invitees have responded to the event invitation, the Event record is updated with their response. Possible responses include *accepted, declined*, *tentative*, and *noResponse.*

Note the following:

- For information on the Event record in Web services, see the help topic *Events (CalendarEvent)*. For general information on scheduling events in NetSuite, see the help topic *Scheduling Events*.)
- There is no async equivalent for the updateInviteeStatus operation.

Note the following when using this operation:

- Users must have a valid session. This operation *does* include a **passport** header to support request level credentials. (For information on request level authentication, see Authentication Using Request Level Credentials.)
- To update other properties on the Event record, the event owner should still use the update operation. Note, however, event owners are not exempt from using updateInviteeStatus if they have to update their own event response status. There may be cases in which event owners must decline their own event and have another person run the event for them.
- Unlike in the NetSuite UI (see figure), invitees will not be able to send a message back to the organizer. Web services does not currently support messages attached to events. An event invitee's response can include only one of the following values: _accepted, _declined, _tentative, _noResponse.



Request

The UpdateInviteeStatusRequest type is used for the request.

Element Name	XSD Type	Notes
eventId	RecordRef	References an existing instance of an Event record.
responseCode	CalendarEventAttendeeResponse	The CalendarEventAttendeeResponse type includes the following enums:

Element Name	XSD Type	Notes
		_accepted _declined _tentative _noResponse

Response

The UpdateInviteeStatusResponse type is used for the response.

Element Name	XSD Type	Notes
response	WriteResponse	Contains details on the status of the operation and a reference to the updated record.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

C#

```
NetSuiteService nss = new NetSuiteService();
// login details omitted

UpdateInviteeStatusReference inviteeStatusRef = new UpdateInviteeStatusReference();

// Set the event id for the status update
RecordRef eventIdRef  = new RecordRef();
eventIdRef.internalId = "100";   // Substitute this with your own event id
inviteeStatusRef.eventId = eventIdRef;

// Set the event status
inviteeStatusRef.responseCode = CalendarEventAttendeeResponse._accepted;

// Update invitee event status
```

```
WriteResponse resp = nss.updateInviteeStatus(inviteeStatusRef);
```

SOAP Request

```
<updateInviteeStatus xmlns="urn:messages_2009_1.platform.webservices.netsuite.com">
            <updateInviteeStatusReference>
               <eventId internalId="100" xmlns="urn:core_2009_1.platform.webservices.netsuite.c
om" />
               <responseCode xmlns="urn:core_2009_1.platform.webservices.netsuite.com">_accepte
d
  </responseCode>
            </updateInviteeStatusReference>
         </updateInviteeStatus>
```

SOAP Response

```
  <updateInviteeStatusResponse xmlns="urn:messages_2009_1.platform.webservices.netsuite.com">
            <writeResponse>
               <platformCore:status isSuccess="true"
                  xmlns:platformCore="urn:core_2009_1.platform.webservices.netsuite.com"/>
               <baseRef internalId="100" type="calendarEvent" xsi:type="platformCore:RecordRef"

  xmlns:platformCore="urn:core_2009_1.platform.webservices.netsuite.com"/>
            </writeResponse>
         </updateInviteeStatusResponse>
```

updateInviteeStatusList

The updateInviteeStatusList operation is used to update one or more NetSuite events. For general details on the updateInviteeStatus operation, see updateInviteeStatus.

Request

The UpdateInviteeStatusListRequest type is used for the request.

Element Name	XSD Type	Notes
updateInviteeStatusReference[]	UpdateInviteeStatusReference	UpdateInviteeStatusReference uniquely identifies the event and the status you want to update the event with.

The UpdateInviteeStatusReference type includes the following elements:

- eventId
- responseCode

Element Name	XSD Type	Notes
eventId	RecordRef	References an existing instance of an Event record.

Element Name	XSD Type	Notes
responseCode	CalendarEventAttendeeResponse	The CalendarEventAttendeeResponse type includes the following enums: _accepted _declined _tentative _noResponse

Response

The UpdateInviteeStatusListResponse type is used for the response.

Element Name	XSD Type	Notes
response[]	WriteResponse	Contains an array of WriteResponse objects, each of which contains details on the status of that updateInviteeStatusList operation and a reference to each event record.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

Sample Code

C#

```
UpdateInviteeStatusReference [] inviteeStatusRefList = new UpdateInviteeStatusReference[2];
                  for (int i=0; i<2; i++)
                     {
                        RecordRef eventIdRef = new RecordRef();
                         if (i==0)
                           eventIdRef.internalId = "100";
                         else
                           eventIdRef.internalId = "101";
                           UpdateInviteeStatusReference statusRef = new UpdateInviteeStatusRefe
```

```
rence();
                            statusRef.eventId = eventIdRef;
                            statusRef.responseCode = CalendarEventAttendeeResponse._accepted;
                            inviteeStatusRefList[i] = statusRef;
                     }
                              nss.updateInviteeStatusList(inviteeStatusRefList);
```

SOAP Request

```
<updateInviteeStatusList xmlns="urn:messages_2009_1.platform.webservices.netsuite.com">
            <updateInviteeStatusReference>
               <eventId internalId="100" xmlns="urn:core_2009_1.platform.webservices.netsuite.c
om" />
               <responseCode xmlns="urn:core_2009_1.platform.webservices.netsuite.com">_accepte
d
  </responseCode>
            </updateInviteeStatusReference>
            <updateInviteeStatusReference>
               <eventId internalId="101" xmlns="urn:core_2009_1.platform.webservices.netsuite.c
om" />
               <responseCode xmlns="urn:core_2009_1.platform.webservices.netsuite.com">_accepte
d
  </responseCode>
            </updateInviteeStatusReference>
         </updateInviteeStatusList>
```

SOAP Response

```
 <updateInviteeStatusListResponse xmlns="urn:messages_2009_1.platform.webservices.netsuite.com"
>
            <writeResponseList>
               <writeResponse>
                  <platformCore:status isSuccess="true"
      xmlns:platformCore="urn:core_2009_1.platform.webservices.netsuite.com"/>
                  <baseRef internalId="100" type="calendarEvent" xsi:type="platformCore:RecordR
ef"
      xmlns:platformCore="urn:core_2009_1.platform.webservices.netsuite.com"/>
               </writeResponse>
               <writeResponse>
                  <platformCore:status isSuccess="true"
       xmlns:platformCore="urn:core_2009_1.platform.webservices.netsuite.com"/>
                  <baseRef internalId="101" type="calendarEvent" xsi:type="platformCore:RecordR
ef"
     xmlns:platformCore="urn:core_2009_1.platform.webservices.netsuite.com"/>
               </writeResponse>
            </writeResponseList>
         </updateInviteeStatusListResponse>
```

upsert

The upsert operation is used to add a new instance or to update an instance of a record in NetSuite. It is similar to the upsertList operation, which allows users to add or update more than one record at a time.

The upsert operation is similar to both the add and update operations, but upsert can be run without first determining whether a record exists in NetSuite. A record is identified by its external ID and its record type. If a record of the specified type with a matching external ID exists in the system, it is updated. If it does not exist, a new record is created.

Because external ID is mandatory for this operation, upsert is supported only for records that support the external ID field. For a list of these records, see the help topic *NetSuite Record Types that Support External ID*. Also, this operation prohibits the passing of internal ID values.

Limitations on upsert

- Although records of a particular type may be used in multiple integration scenarios, each record instance can only have a single external ID value. In order to maintain data integrity, only a single integrated application can set and update external ID values for each record type. External ID values for all records of a particular type must all come from the same external application.
- Upsert cannot complete updates during an initialize / initializeList operation, when a record is transformed into a record of another type. In this case, upsert only adds and does not update records.
- Updates through the upsert operation are also subject to the same limitations as updates through the update operation. For details of these limitations, see update.

Request

The UpsertRequest type is used for the request. It contains the following fields.

Element Name	XSD Type	Notes
record	Record	Contains an array of record objects. The record type is an abstract type so an instance of a type that extends record must be used—such as Customer or Event.

Response

The UpsertResponse type is used for the response. It contains the following fields.

Element Name	XSD Type	Notes
response	WriteResponse	Contains details on the status of the operation and a reference to thecreated or updated record.

Faults

- This operation can throw one of the following faults. See SOAP Fault Status Codes for more information.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault
- UnexpectedErrorFault

- This operation returns the following run-time error if the passed record type does not support the external ID field:

```
INVALID_RCRD_TYPE:  <record_type> does not support external ID and cannot be used with upsert
```

- This operation returns the following run-time error if passed data includes internal ID:

```
USER_ERROR:  You cannot set internalId with upsert.
```

- This operation returns the following run-time error if passed data does not include external ID:

```
USER_ERROR:  This operation requies a value for externalId.
```

For more information about error codes, see Error Status Codes.

Sample Code

SOAP Request

```
<soap:Body>
            <upsert xmlns="urn:messages_2011_2.platform.webservices.netsuite.com">
               <record xmlns:q1="urn:relationships_2011_2.lists.webservices.netsuite.com" xsi:t
ype="q1:Customer" externalId="THISISMYEXTID">
                  <q1:entityId>XYZ 2 Inc</q1:entityId>
                  <q1:companyName>XYZ 2, Inc.</q1:companyName>
                  <q1:email>bsanders@xyz.com</q1:email>
               </record>
            </upsert>
         </soap:Body>
```

SOAP Response

```
<soapenv:Body>
            <upsertResponse xmlns="urn:messages_2011_2.platform.webservices.netsuite.com">
```

```
                <writeResponse>
                   <platformCore:status isSuccess="true" xmlns:platformCore="urn:core_2011_2.pla
tform.webservices.netsuite.com"/>
                   <baseRef internalId="973" externalId="THISISMYEXTID" type="customer" xsi:type
="platformCore:RecordRef" xmlns:platformCore="urn:core_2011_2.platform.webservices.netsuite.com
"/>
                </writeResponse>
             </upsertResponse>
          </soapenv:Body>
```

C#

```
private void upsertCustomer()
        {
              // This operation requires a valid session
              this.login( true );

              Customer customer = new Customer();

              // Get externalId for upsert
              _out.write( "\nEnter externalId for customer record to be created or updated : "
);
              customer.externalId = _out.readLn().ToUpper();

              // Set name and email
              customer.entityId = "XYZ 2 Inc";
              customer.companyName = "XYZ 2, Inc.";
              customer.email = "bsanders@xyz.com";

              // Invoke upsert() operation
              WriteResponse response = _service.upsert( customer );

              // Process the response
              if (response.status.isSuccess )
              {
                    _out.info(
                    "\nThe upsert operation was successful :" +
                    "\ninternalId=" + ((RecordRef) response.baseRef).internalId +
                    "\nexternalId=" + ((RecordRef) response.baseRef).externalId +
                    "\nentityId=" + customer.entityId +
                    "\ncompanyName=" + customer.companyName);
              } else {
                    _out.error( getStatusDetails( response.status ) );
              }
        }
```

Java

```
public void upsertCustomer() throws RemoteException,ExceededUsageLimitFault,
            UnexpectedErrorFault, InvalidSessionFault,ExceededRecordCountFault
      {
            // This operation requires a valid session
            this.login(true);
            Customer customer = new Customer();

            // Get extenalId for add or update
            _console.write("\nEnter externalId for customer record to be updated : ");
```

```
            customer.setExternalId(_console.readLn());

            // Set name and email
            customer.setEntityId("XYZ 2 Inc");
            customer.setCompanyName("XYZ 2, Inc.");
            customer.setEmail("bsanders@xyz.com");

            // Invoke upsert() operation
            WriteResponse response = _port.upsert(customer);

            // Process the response
            if (response.getStatus().isIsSuccess())
            {
                  _console.info("\nThe following customer was created/updated successfully:"
                        + "\nkey=" + ((RecordRef) response.getBaseRef()).getInternalId()
                        + "\nexternalId=" + ((RecordRef) response.getBaseRef()).getExternalId()

                        + "\nentityId=" + customer.getEntityId()
                        + "\ncompanyName=" + customer.getCompanyName()
                        + "\nemail=" + customer.getEmail());
            }
            else
            {
                  _console.error(getStatusDetails(response.getStatus()));
            }
      }
```

upsertList

The upsertList operation is used to add or update one or more instances of a record type in NetSuite.

If there are multiple records, they can either be of the same record type or different record types. For example, it's possible to add or update a customer and a contact within a single request using this operation.

The upsertList operation is similar to both the addList and updateList operations, but upsert can be run without first determining whether records exist in NetSuite. Records are identified by their external ID and their record type. If a record of the specified type with a matching external ID exists in the system, it is updated. If it does not exist, a new record is created.

Because external ID is mandatory for this operation, upsertList is supported only for records that support the external ID field. For a list of these records, see the help topic *NetSuite Record Types that Support External ID*. Also, this operation prohibits the passing of internal ID values.

Limitations on upsertList

- Although records of a particular type may be used in multiple integration scenarios, each record instance can only have a single external ID value. In order to maintain data integrity, only a single integrated application can set and update external ID values for each record type. External ID values for all records of a particular type must all come from the same external application.

- UpsertList cannot complete updates during an initialize / initializeList operation, when a record is transformed into a record of another type. In this case, upsertList only adds and does not update records.
- Updates through the upsertList operation are also subject to the same limitations as updates through the updateList operation. For details of these limitations, see updateList.

Request

The UpsertListRequest type is used for the request. It contains the following fields.

Element Name	XSD Type	Notes
record[]	Record	Contains an array of record objects. The record type is an abstract type so an instance of a type that extends record must be used—such as Customer or Event.

Note: An asynchronous equivalent is available for this operation, **asyncUpsertList**. For information about asynchronous request processing, see Synchronous Versus Asynchronous Request Processing.

Response

The UpsertListResponse type is used for the response. It contains the following fields.

Element Name	XSD Type	Notes
response[]	WriteResponse	Contains an array of WriteResponse objects, each of which contains details on the status of that upsert operation and a reference to the created or updated record.

Faults

This operation can throw one of the following faults. See SOAP Fault Status Codes for more information on faults.

- InvalidSessionFault
- InvalidCredentialsFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- ExceededRequestSizeFault

- UnexpectedErrorFault

This operation returns the following run-time error if the passed record type does not support the external ID field:

```
INVALID_RCRD_TYPE:  <record_type> does not support external ID and cannot be used with upsert
```

This operation returns the following run-time error if passed data includes internal ID:

```
USER_ERROR:  You cannot set internalId with upsert.
```

This operation returns the following run-time error if passed data does not include external ID:

```
USER_ERROR:  This operation requies a value for externalId.
```

For more information about error codes, see Error Status Codes.

Sample Code

SOAP Request

```
<soap:Body>
         <upsertList xmlns="urn:messages_2011_2.platform.webservices.netsuite.com">
            <record xmlns:q1="urn:relationships_2011_2.lists.webservices.netsuite.com" xsi:type
="q1:Customer" externalId="ext1">
               <q1:entityId>XYZ Inc 0</q1:entityId>
               <q1:companyName>XYZ, Inc. 0</q1:companyName>
            </record>
            <record xmlns:q2="urn:relationships_2011_2.lists.webservices.netsuite.com" xsi:type
="q2:Customer" externalId="ext2">
               <q2:entityId>XYZ Inc 1</q2:entityId>
               <q2:companyName>XYZ, Inc. 1</q2:companyName>
            </record>
         </upsertList>
      </soap:Body>
```

SOAP Response

```
<soapenv:Body>
         <upsertListResponse xmlns="urn:messages_2011_2.platform.webservices.netsuite.com">
            <writeResponseList>
               <writeResponse>
                  <platformCore:status isSuccess="true" xmlns:platformCore="urn:core_2011_2.pla
tform.webservices.netsuite.com"/>
                  <baseRef internalId="970" externalId="ext1" type="customer" xsi:type="platfor
mCore:RecordRef" xmlns:platformCore="urn:core_2011_2.platform.webservices.netsuite.com"/>
               </writeResponse>
               <writeResponse>
                  <platformCore:status isSuccess="true" xmlns:platformCore="urn:core_2011_2.pla
tform.webservices.netsuite.com"/>
```

```
                 <baseRef internalId="974" externalId="ext2" type="customer" xsi:type="platfor
mCore:RecordRef" xmlns:platformCore="urn:core_2011_2.platform.webservices.netsuite.com"/>
              </writeResponse>
           </writeResponseList>
        </upsertListResponse>
     </soapenv:Body>
```

C#

```
private void upsertCustomerList()
       {
             // This operation requires a valid session
             this.login( true );

             // Prompt for list of externalIds and put in an array
             _out.write( "\nEnter externalIds for customer records to be updated (separated by
commas): " );
             String reqKeys = _out.readLn();
             string [] nsKeys = reqKeys.Split( new Char[] {','} );

             // Create an array of Record objects to hold the customers
             Record[] records = new Record[nsKeys.Length];

             // For each submitted nsKey, populate a customer object
             for ( int i=0; i<nsKeys.Length; i++)
             {
                   Customer customer = new Customer();

                   // Update name
                   customer.entityId = "XYZ Inc " + i;
                   customer.companyName = "XYZ, Inc. " + i;

                   customer.externalId = nsKeys[i].Trim();
                   records[i] = customer;
             }

             // Invoke upsertList() operation to create or update customers
             WriteResponse[] responses = _service.upsertList( records );

             // Process responses for all successful updates
             _out.info( "\nThe following customers were updated or created successfully:" );

             bool hasFailures = false;
             for ( int i=0; i<responses.Length; i++ )
             {
                   if ( (responses[i] != null) && (responses[i].status.isSuccess) )
                   {
                         _out.info( "\nCustomer[" + i + "]:");
                         _out.info( "internalId=" + ((RecordRef) responses[i].baseRef).interna
lId + " externalId="+((RecordRef) responses[i].baseRef).externalId +
                         "\nentityId=" + ((Customer) records[i]).entityId +
                         "\ncompanyName=" + ((Customer) records[i]).companyName );
                   }
                   else
                   {
                         hasFailures = true;
                   }
             }
```

```
            // Process responses for all unsuccessful updates
            if ( hasFailures )
            {
                  _out.info( "\nThe following customers were not updated:\n" );
                  for ( int i=0; i<responses.Length; i++ )
                  {
                        if ( (responses[i] != null) && (!responses[i].status.isSuccess) )
                        {
                              _out.info( "Customer[" + i + "]:" );
                              _out.info( "key=" + ((RecordRef) responses[i].baseRef).internal
Id );
                             _out.errorForRecord( getStatusDetails(responses[i].status ) );
                        }
                  }
            }
      }
```

Java

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

                        // Prompt for list of externalIds and put in an array
                        _console.write("\nEnter externalIds for customer records to be updated
(separated by commas): ");
                        String reqKeys = _console.readLn();
                        String[] nsKeys = reqKeys.split(",");

                        // Create an array of Record objects to hold the customers
                        Record[] records = new Record[nsKeys.length];

                        // For each submitted nsKey, populate a customer object
                        for (int i = 0; i < nsKeys.length; i++)
                        {
                                    Customer customer = new Customer();

                                    // Update name
                                    customer.setEntityId("XYZ Inc " + i);
                                    customer.setCompanyName("XYZ, Inc. " + i);
                                    customer.setExternalId(nsKeys[i].trim());

                                    records[i] = customer;
                        }

                        // Invoke upsertList() operation to create or update customers
                        WriteResponseList responseList = _port.upsertList(records);

                        // Process responses for all successful upserts
                        WriteResponse[] responses = responseList.getWriteResponse();

                        boolean hasFailures = false;
                        _console.info("\nThe following customers were processed successfully:")
;

                        for (int i = 0; i < responses.length; i++)
                        {
```

```
                                    if ((responses[i] != null) && (responses[i].getStatus().isI
sSuccess()))
                                    {
                                                _console.info("\nCustomer[" + i + "]:");
                                               _console.info("key=" + ((RecordRef) responses[i]
.getBaseRef()).getInternalId()
                                                            + "\nexternalId=" + ((RecordRef) re
sponses[i].getBaseRef()).getExternalId()
                                                            + "\nentityId="          + ((Custo
mer) records[i]).getEntityId()
                                                            + "\ncompanyName=" + ((Customer) re
cords[i]).getCompanyName());
                                    }
                                    else
                                    {
                                                hasFailures = true;
                                    }
                        }

                        // Process responses for all unsuccessful updates
                        if (hasFailures)
                        {
                                    _console.info("\nThe following customers were not updated:\
n");
                                    for (int i = 0; i < responses.length; i++)
                                    {
                                                if ((responses[i] != null) && (!responses[i].ge
tStatus().isIsSuccess()))
                                                {
                                                            _console.info("Customer[" + i + "]:
");
                                                            _console.info("key=" + ((RecordRef)
 responses[i].getBaseRef()).getInternalId());
                                                            _console.errorForRecord(getStatusDe
tails(responses[i].getStatus()));
                                                }
                                    }
                        }

            }
```

Chapter 11 Using Web Services with PHP

Using the Web Services PHP Toolkit

If you are new to PHP and SuiteTalk, it is recommended that you review the following topics:

- PHP Toolkit Overview
- Downloading the PHP Toolkit
- Configuring an Environment for the PHP Toolkit
- Creating a Web Services PHP Project
- Creating and Submitting Records Using the PHP Toolkit
- Logging SOAP Requests and Responses Using the PHP Toolkit

Should you run into issues during your development, see Troubleshooting PHP and Web Services.

Important: Using PHP scripts with SuiteTalk requires PHP version 5.3 or later.

PHP Toolkit Overview

NetSuite provides a library and tools to assist in the development of PHP applications that interface to the SuiteTalk platform.

NetSuite's PHP toolkit for SuiteTalk is a core library containing all classes, objects and methods parsed from the WSDL of a SuiteTalk endpoint. The toolkit also comes with several helper functions that can be used to simplify working with arrays and objects.

Important: Each PHP Toolkit is specific to a SuiteTalk endpoint. You must download the correct version of the toolkit for the endpoint you are using. Also, updating to a new endpoint requires an update of the toolkit. **To use the 2014.2 PHP Toolkit, you must upgrade to the 2014.2 WSDL. The 2014.2 PHP Toolkit does not officially support different WSDL versions.**

Updating Existing PHP Code

NetSuite offered a completely revamped PHP Toolkit as part of Version 2012 Release 2. As of 2012.2, the PHP Toolkit has the ability to generate proxy classes for every new NetSuite WSDL, creating a development environment similar to Axis or .NET.

Additionally, the new toolkit uses a different programming model (the use of proxy classes over pure associative arrays, though the latter is still supported).

If you upgrade to any version of the PHP Toolkit later than 2012.2, existing PHP applications created using previous versions of the toolkit will need to be updated. To understand the nature of the coding changes, see the table below. Also see the sample applications that come with the new toolkit.

	PHP Toolkit 2012.2 and Later	PHP Toolkit Versions Older than 2012.2
Record object initialization	Each record type can now be initialized using the same record type name. Example - To initialize a Customer record object: $CustomerRec = new Customer();	Initialize the object using the object nsComplexObject. Example - To initialize a Customer record object: $CustomerRec = new nsComplexObject('Customer');
Object type Initialization All object types are listed and explained in Platform Enumerations.	Each complex, search and custom field type is initialized using the same object type name: Example - To initialize a basic search for customers: $selectTypeField = new RecordRef(); To initialize a recordRef object: $selectTypeField = new RecordRef();	These objects are initialized using the object nsComplexObject. Example - To initialize a basic search for customers: $search = new nsComplexObject('CustomerSearchBasic'); To initialize a recordRef object: $selectTypeField = new nsComplexObject ('RecordRef');
Calling the NetSuite SuiteTalk (Web Services) operations	The operations are called from the **service** object instantiated from the NetSuiteService class (NetSuiteService.php). Example - $service = new NetSuiteService(); … $addResponse = $service->add($request);	The operations are called from the **client** object instantiated from the nsClient class (declared on the phptoolkit.php). Example - $myNSclient = new nsClient(nsHost::live); … $addResponse = $myNSclient->add($contact);
Request object	You must instantiate a request object depending on the kind of operation. This request object will accept the Record object, which will then be set to the operation. Each operation has a corresponding request type. • AddRequest for add operation • AddListRequest for addList operation • SearchRequest for search operation. Example - To add a customer:	None. The record object is directly set to the operation. Example - To add a customer: $CustomerRec = new nsComplexObject('Customer'); … $addResponse = $myNSclient->add($CustomerRec);

	PHP Toolkit 2012.2 and Later	PHP Toolkit Versions Older than 2012.2
	$service = new NetSuiteService(); $CustomerRec = new Customer(); … $request = new AddRequest(); $request->record = $CustomerRec; … $addResponse = $service->add($request);	

The setFields method	A function to set an array of field objects to the record object.
	
	Example -
	
	$customerRec = new Customer();
	
	$customerFields = array (
	
	'isPerson' => true,
	'firstName' => $firstName,
	'lastName' => $lastName,
	'companyName' => $companyName
	
	);
	
	setFields($customerRec, $ customerFields);
		This is a method of the record object used to set the array of field values to the record.
		
		Example -
		
		$customerFields = array (
		
		'isPerson' => true,
		'firstName' => $firstName,
		'lastName' => $lastName,
		'companyName' => $companyName
		
		);
		
		// create Customer record
		$customer = new nsComplexObject('Customer');
		
		// set fields
		$customer->setFields($ customerFields);

Downloading the PHP Toolkit

To download the 2014.2 PHP Toolkit, click the link for the PHPToolkit_2014_2.zip file.

The toolkit is provided as an archive file and contains:

- A folder called PHPToolkit that contains:
  - NetSuiteService.php – Class library
  - NSconfig.php – Passport configuration file
  - NSPHPClient.php – Helper functions
- A folder called Samples that contains PHP code examples that are usable in your own projects.

Next see, Configuring an Environment for the PHP Toolkit.

Configuring an Environment for the PHP Toolkit

You can run PHP scripts from the command line by installing a PHP interpreter. Command line scripts are suitable for stand-alone applications and are useful for scripting and testing. See Installing PHP for the Command Line for more details.

The PHP Toolkit's real power, however, is in the ability to use a web server-based PHP interpreter module to integrate NetSuite into an existing web presence, whether it is an external e-commerce web site, or an intranet-only Human Resources portal, for example.

A web server executes a PHP script through the included PHP interpreter module. PHP scripts can be text files containing PHP code, or they can be HTML web pages containing PHP tags indicating sections of PHP code. See Installing a Web Server for PHP for more details.

Note: For information on downloading the PHP Toolkit, see Downloading the PHP Toolkit.

Installing PHP for the Command Line

Many distributions of Linux and Mac OS X® are pre-installed with PHP. Packages for Windows® can be obtained from http://windows.php.net/download/.

Once PHP is installed, you need to edit the php.ini file. The PHP Toolkit needs to use certain libraries that are not enabled in a PHP installation by default.

In the php.ini configuration file, in the section [Dynamic Extensions], the following lines need to be added if they do not exist:

```
extension=php_soap.dll
extension=php_openssl.dll
```

The modules indicated by the .dll extensions pertains to a Windows installation. If you are configuring a Unix-based platform, change **.dll** to **.so**:

```
extension=php_soap.so
extension=php_openssl.so
```

Note: For a Windows installation of PHP, change the caching directory to something similar to:

soap.wsdl_cache_dir="c:/Windows/Temp"

Installing a Web Server for PHP

Note: This section talks about using Apache Web Server, but any web server that supports PHP, SSL, and SOAP should be able to be used for SuiteTalk.

The simplest way to provision a web server for the PHP Toolkit is to install a LAMP package. LAMP is the acronym for Linux, Apache, MySQL, and PHP. Once a LAMP package is installed, a complete web server with database and PHP scripting functionality is available with very little configuration required. The PHP Toolkit does require, however, that you make a few modifications to the default LAMP installation.

Note: There are several LAMP distributions available free of charge for most computer architectures. A well known package for Windows® is WAMP, which is available at http://www.wampserver.com/en/. Follow the instructions that come with the package to install and initially configure it.

Once LAMP is installed, the php.ini and the httpd.conf configuration files need to be edited. Note that the LAMP installation will not use any existing PHP installation; it will install its own self-contained PHP package.

In the php.ini configuration file, in the section [Dynamic Extensions], the following lines need to be added if they do not exist:

```
extension=php_soap.dll
extension=php_openssl.dll
```

The modules indicated by the .dll extensions pertains to a Windows installation. If you are configuring a Unix-based platform, change **.dll** to **.so** :

```
extension=php_soap.so
extension=php_openssl.so
```

The Apache Web Server needs to be told to load the PHP module so that it can run PHP scripts. In the httpd.conf configuration file in the section Dynamic Shared Object Support, the following line must be added. (Note that the line may already exist; in that case check that the path to the PHP module is correct.)

```
LoadModule php5_module  C:\<PATH TO PHP>\modules\libphp5.dll
```

The module indicated by the .dll extension pertains to a windows installation. If you are configuring a Unix-based platform, the PHP module will have a **.so** extension. Note also that the path to the PHP module will differ:

```
LoadModule php5_module  /<PATH TO PHP>/bin/php/modules/libphp5.so
```

Important: As indicated in these examples, the name of the module is libphp5.dll or libphp5.so, but be aware that SuiteTalk requires PHP 5.**3** or later. In other words, the name of the module is not specific to the exact PHP version. (However, the exact version is likely indicated by the <PATH TO PHP> folder name.)

Important: After a change to any configuration files, LAMP's services must be restarted.

Important: If you are running PHP scripts from the command line **and** through a web server, there will be **two** php.ini files that need to be edited, one for the command line PHP installation, and one provided by the LAMP installation.

Note: For a Windows installation of PHP, change the caching directory to something similar to:

soap.wsdl_cache_dir="c:/Windows/Temp"

Creating a Web Services PHP Project

For each project you need to perform the following steps:

1. Copy the PHPToolkit directory to the directory in your IDE where your project resides.
2. Edit the NSconfig.php file, substituting your own account details.
3. If SOAP logging is required, create a folder called **nslog** inside the PHPToolkit folder. See Logging SOAP Requests and Responses Using the PHP Toolkit for more details.
4. Each solution needs to include the NetSuiteService.php file by adding the following line to the main source file:

   ```
   require_once '../PHPToolkit/NetSuiteService.php';
   ```

   This step will prompt most IDEs to parse the NetSuiteService.php file for auto completion information.
5. Instantiate a service:

```
$service = new NetSuiteService();
```

Important: Each time you upgrade your NetSuite endpoint, for example from WSDL version 2013.2 to 2014.2, you will also need upgrade the NetSuiteService.php class library to take advantage of any new features.

Creating and Submitting Records Using the PHP Toolkit

These sections provide basic examples for creating and submitting NetSuite records using the PHP toolkit.

Creating a Record with PHP

The following example shows how to create and submit a Purchase Order. In this case, the Purchase Order object is created then populated one field at a time.

Note that after the Purchase Order is instantiated, the IDE should automatically start suggesting members of the PurchaseOrder object using its code completion mechanism.

```
// Include the PHP Toolkit Library
require_once '../PHPToolkit/NetSuiteService.php';

// Instantiate a service
$service = new NetSuiteService();

// Create Purchase Order object
$po = new PurchaseOrder();

// Create entity record
$po->entity = new RecordRef();

// PO internal ID = 38
$po->entity->internalId = 38;

// Create PO Item sublist
$po->itemList = new PurchaseOrderItemList();

// Create PO Item
$poi = new PurchaseOrderItem();
$poi->item = new RecordRef();

// Set Item details
$poi->item->internalId = 15;
$poi->quantity = 5;

// Attach items array to PO
$po->itemList->item = array($poi);
```

Creating a Record with the setFields Method

An alternative method to create the object is to use array notation to set the object fields. The advantage is more compact code; the disadvantage is that code completion will not work.

The following example creates the same Purchase Order object using the setFields method:

```
// Create the Purchase Order object
$po = new PurchaseOrder();
```

```
// Create array of fields
$purchaseOrderFields = array('entity' => array('internalId' => 38), 'itemList' => array('item'
=> array( array('item' => array('internalId' => 15), 'quantity' => 5)) ) );

// Call setFields to recursively create objects and populate their fields
setFields($po, $purchaseOrderFields);
```

In addition to creating and populating objects, the setFields method performs basic type and field checking to make sure the array structure corresponds to the expected object hierarchy and types.

Note: The setFields method is defined in the NSPHPClient.php helper file, which is included automatically by NetSuiteService.php

Submitting a Record

Once the object is created and populated, the object can be submitted to NetSuite in one of the operations supported by the type of object you have created. The following code sample shows the **add** operation:

```
// Create Add request
$request = new AddRequest();

// Attach the Purchase Order to the request
$request->record = $po;

// Submit the record and save the response
$addResponse = $service->add($request);
```

The **add** operation returns a ReadResponse object that is used to access the result of the operation. The following code sample reads and displays a simple textual indication of the result:

```
if (!$addResponse->writeResponse->status->isSuccess)
{
 echo "ADD ERROR";
}
else
{
 echo "ADD SUCCESS, id " . $addResponse->writeResponse->baseRef->internalId;
}
```

Note: The PHP Toolkit package contains more code examples in the Samples folder.

Logging SOAP Requests and Responses Using the PHP Toolkit

To enable SOAP logging with the PHP Toolkit, create a folder called **nslog** in the **PHPToolkit** folder inside the project. The PHP Toolkit will automatically spool the SOAP requests generated by the PHP code, and the responses that have been returned by the server, into the nslog folder in standard XML format.

The following listing shows the typical project structure when logging is enabled:

```
Project_1 <Directory>
 PHPToolkit.php
 NSconfig.php
 NetSuiteService.php
 nslog <Directory>
  20120317.160138.6941-add-request.xml
  20120317.165167.8345-add-response.xml
  20120317.180653.3467-delete-request.xml
  20120317.187474.3356-delete-response.xml
  ...
```

Note: To disable logging, rename or delete the nslog folder.

Troubleshooting PHP and Web Services

The following table provides solutions for problems that may be encountered when using PHP with NetSuite Web Services.

Problem	Solution
The following error is returned when attempting to send requests over https: Fatal error: Uncaught SoapFault exception: [WSDL] SOAP-ERROR: Parsing WSDL: Couldn't load from 'https://webservices.netsuite.com/wsdl/v2013_1_0/netsuite.wsdl'.	Edit the php.ini file by uncommenting "extension=php_openssl.dll" under Dynamic Extensions. Windows only: Make sure that libeay32.dll and ssleay32.dll files are in your path. You can do this by copying the files to your System32 folder.
The following time out error is returned: Fatal error: Maximum execution time of 30 seconds exceeded.	Edit the php.ini file and change max_execution_time - 30 (or default value) to 200.
The following error message is returned: Debug Error: Uncaught SoapFault exception: [HTTP] Error Fetching http headers.	Edit the php.ini file and by setting default_socket_timeout = 200.
The following error message is returned: Warning: It is not yet possible to assign complex types to properties.	Open the php.ini file and edit error_reporting. Set it to E_ERROR instead of E_ALL.
The execution of script stops and error was not shown, or notices are being shown on the page.	Open the php.ini file and set error_reporting = E_ERROR & ~E_NOTICE.
The following error is returned: Fatal error: Allowed memory size of 8388608 bytes exhausted (tried to allocate 3024 bytes).	Open the php.ini file and set memory_limit = 80M.

Problem	Solution
Performance issue: Every time a web services request executes, PHP performs a GET on the wsdl.	For a non-Windows installation of PHP, look for property soap.wsdl_cache_dir in the php.ini file and make sure that the specified folder ("/tmp" -> c:\tmp) exists in the host. **Note:** For a **Windows** installation of PHP, change the caching directory to something similar to: soap.wsdl_cache_dir="c:/Windows/Temp"
You have made a change in your php.ini file, yet the value you set is not respected.	Make sure that the property exists only once in the php.ini file.

Chapter 12 Web Services Error Handling and Error Codes

This section provides information on the following:

- The three exception types that are supported in the SuiteTalk: warnings, errors, and faults. See Understanding Web Services Warnings, Errors, and Faults for an overview.
- SOAP Faults that can be thrown for each SuiteTalk operation. See SOAP Faults for Each Operation.
-
    - SOAP Fault Status Codes
    - Error Status Codes
    - Warning Status Codes

Understanding Web Services Warnings, Errors, and Faults

Based on the error code or fault that is received, the client can take the appropriate action.

- Warnings: informational notifications requiring an action within the UI but requiring no response from a Web service call. Data may or may not be processed depending on preference settings. For more information on preferences, see Setting Company-wide Preferences.
- Errors: exceptions returned on a record-by-record basis due to invalid or incomplete data. The service is processed as requested, however, only those records without errors are updated.
- Faults: a fundamental exception type that results in the entire request not being processed.

Warnings

A warning is a notification sent to a user in order to prevent a subsequent error or to ensure better data quality. In the Netsuite UI, a warning is presented to the user through a dialog box and generally requires an action from the user. Since there is no interaction of this nature in the Web services model, the request must specify what to do in the case of a warning. The options are:

- Ignore the warning and submit the record to the database
- Heed the warning and abort the submission — treat as an error

You can set a company wide preference on the Web Services Preferences page on how to handle warnings or you can specify how warnings should be handled in a specific request. Request level preferences override company-wide preferences. Refer to Setting Company-wide Preferences for information on how to set company-wide preferences.

Following is an example of a warning message in the response to a request to add a customer record without a zip code (with the Treat Warnings as Errors preference set to false):

Code

```
<addResponse xmlns="urn:messages.platform_2_5.webservices…">
 <writeResponse xmlns="urn:messages.platform_2_5…">
  <ns1:status isSuccess="true" xmlns:ns1="urn:core.platform_2_5…">
   <ns1:statusDetail type="WARN">
    <ns1:code>WARNING</ns1:code>
   <ns1:message>Without a zip/postal code it will not be possible to use
    this address with 3rd Party shippers. Click Cancel to edit the address.
   </ns1:message>
   </ns1:statusDetail>
  </ns1:status>
  <ns2:recordRef internalId="50" type="customer" xmlns:ns2="urn:core.platform_2_5…"/>
 </writeResponse>
</addResponse>
```

The above customer was added because no error occurred. However, future errors could be avoided by responding appropriately to the warning message.

For a detailed list of all warning messages and the associated codes generated by NetSuite, see Warning Status Codes.

Errors

Errors result if invalid or incomplete data is submitted when performing an operation. For example, if a client attempts to update a customer record with an invalid internal ID, the operation fails and the response contains an error code and message.

When numerous records are submitted within the same request, each is treated individually. For example, if a client attempts to update two events within the same request, where one record has invalid data and the other has valid data, only one of the records has an error and is not updated.

Code

```
<updateListResponse xmlns="urn:messages.platform_2_5…">
 <listWriteResponse>
```

```
  <writeResponse>
   <ns1:status isSuccess="true" xmlns:ns1="urn:core.platform_2_5··· />
   <ns2:recordRef internalId="100010" type="event"
    xmlns:ns2="urn:core.platform···" />
  </writeResponse>
  <writeResponse>
   <ns3:status isSuccess="false" xmlns:ns3="urn:core.platform_2_5···">
    <ns3:statusDetail type="ERROR">
     <ns3:code>USER_EXCEPTION</ns3:code>
     <ns3:message>Invalid record:
      type=event,id=100015,scompid=TSTDRV96
     </ns3:message>
    </ns3:statusDetail>
   </ns3:status>
   <ns4:recordRef internalId="100015" type="event"
    xmlns:ns4="urn:core.platform_2_5···" />
  </writeResponse>
 </listWriteResponse>
</updateListResponse>
```

For a detailed list of all error messages and the associated codes, see Error Status Codes.

Faults

Faults are exceptions that are of a more fundamental nature than errors. A key distinction between errors and faults is that a fault prevents any operation within a request from being processed whereas an error prevents only a single operation from succeeding on an individual record.

For example, continuing from the case above, if the user's session has timed out when making the request, neither update for either event record is processed and an invalidSessionFault is returned in the response.

Expand code

```
<soapenv:Fault>
 <faultcode>soapenv:Server.userException</faultcode>
 <faultstring>com.netledger.dto.faults.InvalidSessionFault: Your connection has
  timed out. Please log in again.
 </faultstring>
 <detail>
  <invalidSessionFault xmlns="urn:faults.platform_2_5.webservices.netsuite.com">
   <code>INVALID_SESSION</code>
   <message>Your connection has timed out. Please log in again.</message>
  </invalidSessionFault>
  <ns1:stackTrace xmlns:ns1="http://xml.apache.org/
  axis">com.netledger.dto.faults.InvalidSessionFault: Your connection has timed
   out. Please log in again.
  </ns1:stackTrace>
 </detail>
</soapenv:Fault>
```

SOAP uses the detail element to capture the error code through the code element and the error message through the message element. The faultcode and faultstring are automatically populated by the server.

Following is an example of a SOAP fault named InvalidCredentialsFault for an invalid e-mail address that is returned as part of the login operation.

Code

```
<soapenv:Fault>
  <faultcode>soapenv:Server.userExeption</faultcode>
  <faultstring>com.netledger.dto.faults.InvalidCredentialsFault: You have entered an
  invalid e-mail address or account number. Please try again.</faultstring>
  <detail>
   <InvalidCredentialsFault
   xmlns=”urn:faults.platform_2_5.webservices.netsuite.com”>
    <code>INVALID_USERNAME</code>
    <message>You have entered an invalid e-mail address or account number.
    Please try again.</message>
   </InvalidCredentialsFault>
  </detail>
</soapenv:Fault>
```

For a detailed list of all fault messages and the associated codes, see SOAP Fault Status Codes.

Important: Unexpected errors return an error ID that can be provided to NetSuite Support to help them isolate the error in your account. The following is an example of the error ID:

```
<ns2:code>UNEXPECTED_ERROR</ns2:code>
<ns2:message>An unexpected error occurred. Error ID: fevsjhv41tji2juy3le73</ns2:message>
```

Error IDs will be especially helpful for async operations, as the ID will indicate the error that occurred during the execution of the job, not the retrieval. The exception is if the async error occurred at the platform-level. In this case, the stack will be attached to the job itself.

SOAP Faults for Each Operation

The following table lists the SOAP faults that can be thrown for each SuiteTalk operation.

Login throws...
ACCT_TEMP_UNAVAILABLE
EMAIL_ADDRS_REQD
INVALID_ACCT
INVALID_LOGIN_CREDENTIALS

Login throws…
INVALID_VERSION
PSWD_REQD
UNEXPECTED_ERROR
WS_CONCUR_SESSION_DISALLWD
WS_FEATURE_REQD
WS_PERMISSION_REQD

MapSso throws…
ACCT_TEMP_UNAVAILABLE
EMAIL_ADDRS_REQD
INVALID_ACCT
INVALID_LOGIN_CREDENTIALS
INVALID_VERSION
PSWD_REQD
UNEXPECTED_ERROR
WS_CONCUR_SESSION_DISALLWD
WS_FEATURE_REQD
WS_PERMISSION_REQD

Logout throws...
INVALID_VERSION
UNEXPECTED_ERROR
WS_CONCUR_SESSION_DISALLWD
SESSION_TIMED_OUT

add / update / delete throw...
INVALID_VERSION
SESSION_TIMED_OUT
UNEXPECTED_ERROR
USER_ERROR
WS_CONCUR_SESSION_DISALLWD
WS_LOG_IN_REQD

addList / updateList / deleteList throw...
INVALID_VERSION
MAX_RCRDS_EXCEEDED
SESSION_TIMED_OUT
UNEXPECTED_ERROR

addList / updateList / deleteList throw...
USER_ERROR
WS_CONCUR_SESSION_DISALLWD
WS_LOG_IN_REQD

getList/getAll throw...
INVALID_VERSION
MAX_RCRDS_EXCEEDED
SESSION_TIMED_OUT
UNEXPECTED_ERROR
USER_ERROR
WS_CONCUR_SESSION_DISALLWD
WS_LOG_IN_REQD

get throws...
INVALID_VERSION
SESSION_TIMED_OUT
UNEXPECTED_ERROR
USER_ERROR
WS_CONCUR_SESSION_DISALLWD
WS_LOG_IN_REQD

search / searchNext / searchMore throw...
INVALID_VERSION
MAX_RCRDS_EXCEEDED
SESSION_TIMED_OUT
UNEXPECTED_ERROR
USER_ERROR
WS_CONCUR_SESSION_DISALLWD
WS_LOG_IN_REQD

SOAP Fault Status Codes

The following table defines SOAP fault types and their corresponding codes. For a complete description of faults and how they differ from errors and warnings, refer to Understanding Web Services Warnings, Errors, and Faults.

Fault Name	Description
InsufficientPermissionFault	This fault is thrown when the client does not have the appropriate permissions to perform an action based on the role under which they are currently logged in. If

Fault Name	Description
	the client (user) has more than one role, they may need to login again supplying a different role with more permissions.
InvalidAccountFault	This fault is thrown when the client attempts to login with an invalid account id.
InvalidPartnerCredentials	Partner ID or password submitted with this request is invalid.
InvalidRequestIP	Originating IP is not one of the registered IPs from which this Partner request may come.
InvalidSessionFault	This fault is thrown when the client's session has timed out or was terminated as the result of a second Web services client establishing another session. The fault message occurs on the first request attempted after the session is terminated/ timed out. For more details on how NetSuite handles sessions, see Session Management.
InvalidVersionFault	This fault is thrown in the event that the request message contains an unsupported version of the schema.
ExceededRecordCountFault	This fault is thrown in the event the maximum number of records allowed for an operation has been exceeded. For more information, see Understanding Web Services Governance.
ExceededRequestLimitFault	This fault is thrown if the allowed number of concurrent requests is exceeded. For more information, see Understanding Web Services Governance.
ExceededRequestSizeFault	
ExceededUsageLimitFault	
UnexpectedErrorFault	This fault is thrown in the event of an occurrence of an unexpected exception.
InvalidCredentialsFault	This fault is thrown in the event of an invalid username (e-mail), password and/or role supplied in a login attempt.
AsyncFault	

Error Status Codes

The following table lists error status code types that can be returned in a message Response. These are values that are used in the code field of the statusDetail type where the type attribute has a value of *error*.

Error Code Returned	Long Description or Message
ABORT_SEARCH_EXCEEDED_MAX_TIME	This search has timed out. You can choose to schedule it to run in the background and have the results emailed to you when complete. On the saved search form, click the Email tab, check Send According to Schedule, choose an email address on the Specific Recipients subtab and a recurrence pattern on the Schedule subtab.
ABORT_UPLOAD_VIRUS_DETECTED	The file {1} contains a virus {2}. Upload abort.
ACCESS_DENIED	Access to this configuration is denied. Please contact NetSuite to gain access to this configuration.
ACCTNG_PRD_REQD	Missing next accounting period
ACCT_DISABLED	account disabled

Error Code Returned	Long Description or Message
ACCT_DISABLED	This account has been disabled.
ACCT_DISABLED	Please contact Accounts Receivable at 650.627.1316 to re-enable this company.
ACCT_DISABLED	Your account has been inactivated by an administrator.
ACCT_MERGE_DUP	The account merge would result in one or more items using duplicate accounts.
ACCT_NAME_REQD	Accounts require a name.
ACCT_NEEDS_ CAMPAIGN_PROVISION	Please contact your account representative to provision campaign emailing for your account.
ACCT_NOT_CREATED	Account creation was unsuccessful. Technical Support has been alerted to this problem.
ACCT_NUM_REQD	Missing Account Number. Account number is a required field and it cannot be null or empty.
ACCT_NUMS_REQD_OR_ DONT_MATCH	Missing ACCT # or ACCT numbers don't match
ACCT_PERIOD_SETUP_ REQD	The accounting period range {1} has not been defined. Please visit '<A href='{2}'>Setup > Accounting > Manage Accounting Periods</A>' to define this period or set up your year.
ACCT_PRDS_BEING_ ADDED	Periods are currently being added to this account. Please try again later.
ACCT_REQD	Attempting to adjust provisioning for a customer without an existing account
ACCT_TEMP_DISABLED	You have entered an invalid password on {1} consecutive attempts. Access to your account has been suspended for {2} minutes. If you have forgotten your password, please contact Customer Support.
ACCT_TEMP_ UNAVAILABLE	Can't update information - this company's database is currently offline for maintenance. Please try again later.
ACCT_TEMP_ UNAVAILABLE	(Temporarily unavailable)
ACCT_TEMP_ UNAVAILABLE	The account you are trying to access is currently unavailable while we undergo our regularly scheduled maintenance.
ACCT_TEMP_ UNAVAILABLE	We are currently performing maintenance on our system. Please try again soon.
ACCT_TEMP_ UNAVAILABLE	The account you are trying to access is currently unavailable while we undergo our regularly scheduled maintenance.
ACCT_TEMP_ UNAVAILABLE	Your account is disabled for {1} more minutes due to {2} consecutive failed login attempts.
ACCT_TEMP_ UNAVAILABLE	Your account is not yet ready for you to log in. Please wait and try again.
ACCT_TEMP_ UNAVAILABLE	Your company database is offline.
ACCT_TEMP_ UNAVAILABLE	Your data is still being loaded. Please try again later. Contact Professional Services if you have questions.

Error Code Returned	Long Description or Message
ACH_NOT_AVAILBL	ACH Processing is not available in this environment.
ACH_SETUP_REQD	Account {1} is not setup for ACH transactions.
ACTIVE_AP_ACCT_REQD	This transaction requires an active Accounts Payable account. Please enable an existing Accounts Payable account, create a new Accounts Payable account, or contact your System Administrator.
ACTIVE_ROLE_REQD	You can only set an active login role as the Web Services default role.
ACTIVE_TRANS_EXIST	There are active direct deposit transactions for this paycheck
ADDRESS_LINE_1_REQD	Address Line 1 is a required field and it cannot be null or empty.
ADMIN_ACCESS_REQ	At least one active administrator for each account must have access.
ADMIN_ACCESS_REQ	At least one active administrator for this account must have access.
ADMIN_ACCESS_REQD	Only administrators may enter a memorized transaction in a closed period.
ADMIN_ONLY_ACCESS	{1} only the administrator may access this page.
ADMIN_ONLY_ACCESS	{1} only the adminstrator may currently access this page.
ADMIN_USER_REQD	User is not an Admin of the demo account.
ADMISSIBILITY_PACKG_TYP_REQD	An Admissibility Package Type is required for this international shipment.
ALL_DATA_DELETE_REQD	You must first delete all the data in your account before performing this action. Click here to delete your data.
ALL_MTRX_SUBITMES_OPTNS_REQD	The following matrix subitems exist but aren't included in the options you just specified. On the Matrix tab, please make sure the options you select include all existing subitems:<p> {1}
ALREADY_IN_INVT	The following {1} numbers are already in inventory: {2}
ALREADY_IN_INVT	The following {1} number is already in inventory: {2}
AMORTZN_INVALID_DATE_RANGE	Amortization end date can not be before amortization start date.
AMORTZN_TMPLT_DATA_MISSING	One or more line items on this transaction have Variable Amortization Templates, but do not have the required {1} also populated. Please either change the Template for these items or indicate which {1} will be used to schedule the amortization.
AMT_DISALLWD	Description items may not have an amount.
AMT_EXCEEDS_APPROVAL_LIMIT	No one in your chain of command has a sufficient spending limit to approve this transaction.
ANSWER_REQD	Please provide an answer.
APPROVAL_PERMS_REQD	{1} The restrictions on your role do not allow you to approve or reject this record.
AREA_CODE_REQD	Please include an area code with the phone number.
ASSIGNEE_REQD	{1} must be assigned to {2}
AT_LEAST_ONE_FILE_REQD	Download folder must have at least one file.

Error Code Returned	Long Description or Message
AT_LEAST_ONE_PACKAGE_REQD	1 or more packages are required.
AT_LEAST_ONE_RETURN_FLD_REQD	ou must specify at least one field to be returned.
AT_LEAST_ONE_SUB_REQD	You must choose at least one subsidiary.
ATTACH_SIZE_EXCEEDED	The data you are uploading exceeds the maximum allowable size of {1}. Please change your selection and try again
ATTACH_SIZE_EXCEEDED	You have exceeded the maximum attachments size of 5.0 MB. Please remove one or more attachments and try again.
ATTACHMNT_CONTAINS_VIRUS	The attachment with file name {1} contains a virus {2}. It is removed from the message.
ATTACHMNT_CONTAINS_VIRUS	The attachment file {0} contains virus {1}. Save message abort.
AUDIT_W2_1099	Use this report to audit the information that will be used to generate W2s and 1099s.
AUTO_NUM_UPDATE_DISALLWD	We currently do not support an automatic numbering update of more than {1} {2} records. Please contact <A href='/app/crm/support/nlcorpsupport.nl?type=support'>NetSuite support</A> to request a full numbering update of your {2}s.
BALANCE_EXCEEDS_CREDIT_LIMIT	Customer balance exceeds credit limit
BANK_ACCT_REQD	You must have a bank account to perform this operation. Click here to add one.
BASE_CRNCY_REQD	You may not delete you base currency.
BILL_PAY_STATUS_UNAVAILABLE	View Online Bill Pay Status information is currently not available. Please try again in a few minutes.
BILL_PAY_STATUS_UNAVAILABLE	View Online Bill Pay Status is not available until your bill pay registration is complete.
BILL_PMTS_MADE_FROM_ACCT_ONLY	Your payment has been recorded, but online bill payments can only be made from the account {1}, so no online bill pay payment will be made. You should return to the payment screen if you wish to print the check.
BILLABLES_DISALLWD	{1} does not allow billables.
BILLING_ISSUES	Your account has been locked due to billing issues. You must call your NetSuite Sales Representative for further assistance.
BILLING_ISSUES	Your account has not been fully paid for. Please log in to your account and follow the billing process or contact your Account Manager.
BILLING_SCHDUL_INVALID_RECURR	Billing schedules may not have a recurrence count greater than 500
BILLPAY_APPROVAL_UNAVAILBL	Approve Online Bill Payments is currently not available. Please try again in a few minutes.

Error Code Returned	Long Description or Message
BILLPAY_REGSTRTN_REQD	Online bill pay approve payments is not available until your billpay registration is complete.
BILLPAY_SRVC_UNAVAILBL	Online Bill Pay service is temporarily suspended. If you prefer to wait until Online Bill Pay service is restored, just leave payments to be approved in this list. We will notify you by email as soon as the service is available again. If you need to make an urgent payment, we suggest that you print a check. To do this: 1. Clear the Online Bill Pay check box for that payment. 2. Click the underlined date of the payment, and you will return to the Bill Payment page. 3. On the Bill Payment page, click To Be Printed instead of Bill Pay. 4. Click Submit. 5. Go to Transactions > Print Checks and Forms > Checks, and then mark the check to be printed. 6. Click Submit.
BILLPAY_SRVC_UNAVAILBL	The Online Bill Pay service is currently not available. Please try again in a few minutes.
BIN_DSNT_CONTAIN_ENOUGH_ITEM	The following bins do not contain enough of the requested item ({1}): {2}
BIN_DSNT_CONTAIN_ENOUGH_ITEM	The following bin does not contain enough of the requested item ({1}): {2}
BIN_ITEM_UNAVAILBL	The following bins are not available for the specified item: {1}
BIN_ITEM_UNAVAILBL	The following bins are not available for the specified item ({1}): {2}
BIN_ITEM_UNAVAILBL	The following bin in not available for the specified item ({1}): {2}
BIN_ITEM_UNAVAILBL	The following bin is not available for the specified item: {1}
BIN_SETUP_REQD	The following bins are not associated with the item '{1}': {2}. You can associate bins with an item on the inventory tab of the item record.
BIN_UNDEFND	The following bins specified for the item {1} are not defined in the transaction location ({2}): {3}
BUNDLE_IS_DEPRECATED	This bundle is no longer available. It has been deprecated by bundle {1} on account {2}.
BUNDLE_IS_DEPRECATED	You cannot update this bundle as it has been deprecated and you have not been granted access to the replacement bundle. Please contact the solution provider.
CALENDAR_PREFS_REQD	Set up {1} Calendar Preferences first.
CALENDAR_PREFS_REQD	Set up Calendar Preferences first
CAMPAGIN_ALREADY_EXECUTED	You cannot delete email campaigns that have already been executed
CAMPAIGN_IN_USE	You cannot delete a campaign event that already has activity.
CAMPAIGN_SET_UP_REQD	The following steps need to be performed before a campaign can be created:<p>{1}
CANNOT_RESET_PASSWORD	Sorry, we are currently unable to reset your password. Please contact support or try again later.
CANT_APPLY_PMT	The top level entity cannot accept payment because it has a status of {1}. It must have a status of {2} or {3} to accept payment.
CANT_APPLY_PMT	This entity cannot accept payment because it has a status of {1}. It must have a status of {2} or {3} to accept payment.

Error Code Returned	Long Description or Message
CANT_AUTO_CREATE_ADJSTMNT	The 'Intercompany Expenses' accounting preference does not currently support the automated creation of adjustments. If you would like to create adjustments automatically, please change that preference to 'Allow and Auto Adjust.'
CANT_CALC_FEDEX_RATES	FedEx rates cannot be calculated:
CANT_CANCEL_APPRVD_RETRN_AUTH	You cannot cancel this return authorization because it has already been approved.
CANT_CANCEL_BILL_PMT	The Online Bill Payment cannot be stopped because the payment may already have been made.
CANT_CHANGE_CONTACT_RESTRICTN	You cannot change the restriction on this contact.
CANT_CHANGE_CRMRECORDTYPELINKS	Cannot alter standard CrmRecordTypeLinks
CANT_CHANGE_EVENT_PRIMARY_TYP	You cannot change the primary type for this event
CANT_CHANGE_IP_ADDRESS	The domain {1} is currently associated with IP address {2}. You cannot change the IP address of a live domain.
CANT_CHANGE_LEAD_SOURCE_CAT	You cannot change the category for a leadsource that is defined as the default leadsource for another category
CANT_CHANGE_PSWD	Cannot change password as the company user does not exist.
CANT_CHANGE_PSWD	You changed your password less than 24 hours ago. NetSuite only allows one password change per 24-hour period.
CANT_CHANGE_REV_REC_TMPLT	The rev rec template on a billable expense can not be changed or removed once it is saved.
CANT_CHANGE_REV_REC_TMPLT	The rev rec template on billable time and items can not be changed or removed once it is saved.
CANT_CHANGE_SUB	You cannot change the subsidiary on this record because doing so will change the subsidiary selected on the associated employee record.
CANT_CHANGE_TASK_LINK	Cannot alter standard task links
CANT_CHANGE_UNITS_TYP	You may not change the units type of an item after it has been set.
CANT_CHANGE_VSOE_ALLOCTN	You are attempting to change the VSOE Allocation for a transaction in a closed period. You must either change the posting period for the related transaction or open the period.
CANT_CHG_POSTED_BILL_VRNC	The receipts for this bill can not be changed once the bill variance has been posted.
CANT_CHG_POSTED_BILL_VRNC	One or more lines have had their bill variance posted and can not be changed.
CANT_COMPLETE_FULFILL	The fulfillment cannot be completed.

Error Code Returned	Long Description or Message
CANT_CONNECT_TO_ STORE	Error - Unable to connect to store {1}
CANT_CREAT_SHIP_ LABEL	A shipping label could not be generated because the In Bond Code field is not set. Please enter a value in the In Bond Code field on the Item Fulfillment page.
CANT_CREATE_FILES	Could not create files for uploading your data
CANT_CREATE_NON_ UNIQUE_RCRD	A record with the same unique signatures already exists. You must enter unique signatures for each record you create.
CANT_CREATE_PO	Purchase Orders cannot be created for assembly items.
CANT_CREATE_SHIP_ LABEL	A shipping label could not be generated because a currency must be defined for the "Ship From" country "{1}" when using the Insured Value option. Go to Lists -> Accounting -> Currencies to create a currency for {1}.
CANT_CREATE_SHIP_ LABEL	A shipping label could not be generated because a currency must be defined for the "Ship To" country "{1}" when using the COD option. Go to Lists -> Accounting -> Currencies to create a currency for {1}.
CANT_CREATE_SHIP_ LABEL	A shipping label could not be generated because the Addressee field of the "Ship To" address is not set. Please enter a "Ship To" Addressee on the Item Fulfillment page.
CANT_CREATE_SHIP_ LABEL	A shipping label could not be generated because the Address 1 field of the "Ship From" address is not set. Please go to $(regex) to enter the "Ship From" Address 1.
CANT_CREATE_SHIP_ LABEL	A shipping label could not be generated because the Address 1 field of the "Ship To" address is not set. Please enter a "Ship To" Address 1 on the Item Fulfillment page.
CANT_CREATE_SHIP_ LABEL	A shipping label could not be generated because the Attention field of the "Ship From" address is not set. Please go to $(regex) to enter the "Ship From" Attention.
CANT_CREATE_SHIP_ LABEL	A shipping label could not be generated because the City field of the "Ship To" address is not set. Please enter a "Ship To" City on the Item Fulfillment page.
CANT_CREATE_SHIP_ LABEL	A shipping label could not be generated because the City of the "Ship From" address is not set. Please go to $(regex) to enter the "Ship From" City.
CANT_CREATE_SHIP_ LABEL	A shipping label could not be generated because the Company or Location Name of the "Ship From" address is not set. Please go to $(regex) to enter the "Ship From" Company/Location Name.
CANT_CREATE_SHIP_ LABEL	A shipping label could not be generated because the Country field of the "Ship To" address is not set. Please enter a "Ship To" Country on the Item Fulfillment page.
CANT_CREATE_SHIP_ LABEL	A shipping label could not be generated because the Country of the "Ship From" address is not set. Please go to $(regex) to set the "Ship From" Country.
CANT_CREATE_SHIP_ LABEL	A shipping label could not be generated because the Package Weight was not entered. Please enter a value in the Package Weight field on the Item Fulfillment page.
CANT_CREATE_SHIP_ LABEL	A shipping label could not be generated because the Phone Number of the "Ship From" address is not set. Please go to $(regex) to set the "Ship From" Phone Number.

Error Code Returned	Long Description or Message
CANT_CREATE_SHIP_LABEL	A shipping label could not be generated because the Phone Number of the "Ship To" address is not set. Please enter a "Ship To" Phone Number on the Item Fulfillment page.
CANT_CREATE_SHIP_LABEL	A shipping label could not be generated because the Pickup Type was not set. Please go to Setup > Set Up Shipping to select a shipping Pickup Type.
CANT_CREATE_SHIP_LABEL	A shipping label could not be generated because the shipping method was not set. Please go to Lists > Shipping Items to select a Shipping Label Integration shipping method for this shipping item.
CANT_CREATE_SHIP_LABEL	A shipping label could not be generated because the State field of the "Ship To" address is not set. Please enter a "Ship To" State on the Item Fulfillment page.
CANT_CREATE_SHIP_LABEL	A shipping label could not be generated because the State of the "Ship From" address is not set. Please go to $(regex) to enter the "Ship From" State.
CANT_CREATE_SHIP_LABEL	A shipping label could not be generated because the Zip Code of the "Ship From" address is not set. Please go to $(regex) to enter the "Ship From" Zip Code.
CANT_CREATE_SHIP_LABEL	A shipping label could not be generated because the Zip field of the "Ship To" address is not set. Please enter a "Ship To" Zip code on the Item Fulfillment page.
CANT_CREATE_SHIP_LABEL	A shipping label could not be generated because your {1} Account Number is not set. Go to Setup > Set Up Shipping > {2} Registration to enter your {3} Account Number.
CANT_CREATE_SHIP_LABEL	A shipping label could not be generated because your {1} Registration Address Line 1 is not set. Go to Setup > {2} Registration to complete the Address Line 1 field.
CANT_CREATE_SHIP_LABEL	A shipping label could not be generated because your {1} Registration City is not set. Go to Setup > {2} Registration to enter your City.
CANT_CREATE_SHIP_LABEL	A shipping label could not be generated because your {1} Registration Company field is not set. Go to Setup > {2} Registration to enter a name in the Company field.
CANT_CREATE_SHIP_LABEL	A shipping label could not be generated because your {1} Registration Country is not set. Go to Setup > {2} Registration to select your Country.
CANT_CREATE_SHIP_LABEL	A shipping label could not be generated because your {1} Registration Ship to Attention field is not set. Go to Setup > {2} Registration to enter a name in the Ship to Attention field.
CANT_CREATE_SHIP_LABEL	A shipping label could not be generated because your {1} Registration State is not set. Go to Setup > {2} Registration to select or enter your State.
CANT_CREATE_SHIP_LABEL	A shipping label could not be generated because your {1} Registration Zip Code is not set. Go to Setup > {2} Registration to enter your Zip Code.
CANT_CREATE_WORK_ORD	Work orders can only be created for assembly items.
CANT_DEL_DEFAULT_CALENDAR	
CANT_DEL_DEFAULT_SHIP_METHOD	This Shipping Item cannot be deleted because it is the Default Shipping Method. Please go to Setup > Accounting > Set Up Shipping and choose a new Default Shipping Method before deleting this Shipping Item.

Error Code Returned	Long Description or Message
CANT_DEL_REALIZED_GAINLOSS	A Realized Gain/Loss Transaction cannot be deleted.
CANT_DEL_TRANS_RVRSL	The reversal of the month-end Unrealized Gain/Loss transaction cannot be deleted.
CANT_DELETE_ACCT	This account cannot be deleted because it has associated transactions.
CANT_DELETE_ACCT	This account cannot be deleted because it is a special type of account needed by {1}
CANT_DELETE_ACCT	This account cannot be deleted because it is a special type of account needed by NetSuite
CANT_DELETE_ACCT	This account cannot be deleted because it is a special type of account needed by the system.
CANT_DELETE_ACCT	This account cannot be deleted because it is used by one or more transactions or it has child accounts or it is used by one or more items.
CANT_DELETE_ACCT_PRD	You may not delete an accounting period with transactions posted to it. You must first edit the transactions, change the posting period and then delete the period.
CANT_DELETE_ALLOCTN	This allocation detail can not be deleted because it has a journal entry.
CANT_DELETE_BIN	You may not delete this bin record because it is already in use. You must either remove all references to it in item records and transactions or make it inactive.
CANT_DELETE_CATEGORY	This category cannot be deleted because it has child items
CANT_DELETE_CATEGORY	This category cannot be deleted because it has subcategories
CANT_DELETE_CC_PROCESSOR	This credit card processor is used in transaction and cannot be deleted.
CANT_DELETE_CELL	This cell cannot be deleted because it has child items
CANT_DELETE_CHILD_RCRD_FOUND	This {1} record cannot be deleted because it is referenced by other records.
CANT_DELETE_CHILD_RCRDS_EXIST	This record can not be deleted because it has child records.
CANT_DELETE_CLASS	This class cannot be deleted because it has child items
CANT_DELETE_COLOR_THEME	This color theme cannot be deleted because it is being used
CANT_DELETE_COMMSSN_SCHDUL	This schedule has already been used to generate commission calculations and can't be deleted. If no authorizations have been made, schedule can be deleted after being removed from all active plans.
CANT_DELETE_COMPANY	This company cannot be deleted because it has child entities
CANT_DELETE_COMPANY_TYP	This company type cannot be deleted because the company has associated transactions.
CANT_DELETE_CONTACT_HAS_CHILD	The contact record cannot be deleted because it has child records.

Error Code Returned	Long Description or Message
CANT_DELETE_CONTACT_HAS_CHILD	This contact cannot be deleted because it has child entities
CANT_DELETE_CSTM_FIELD	This custom field cannot be deleted because it is referred to by other custom fields
CANT_DELETE_CSTM_FORM	This custom form cannot be deleted because it is referred to by other custom forms
CANT_DELETE_CSTM_ITEM_FIELD	This custom item field has dependent matrix items. It can not be deleted.
CANT_DELETE_CSTM_LAYOUT	This custom layout cannot be deleted because it is used by custom forms
CANT_DELETE_CSTM_LIST	This custom list cannot be deleted because it is referred to by custom fields
CANT_DELETE_CSTM_RCRD	This custom record cannot be deleted because it is referred to by custom fields
CANT_DELETE_CSTM_RCRD_ENTRY	This custom record entry cannot be deleted because it is referred to by other records
CANT_DELETE_CUST	You can't delete this customer because it's set up as default Anonymous Customer
CANT_DELETE_CUSTOMER	This customer or job cannot be deleted because it has child entities.
CANT_DELETE_DEFAULT_FLDR	You cannot delete the default folders.
CANT_DELETE_DEFAULT_PRIORITY	You cannot delete the default case priority. Please select a new default first.
CANT_DELETE_DEFAULT_SALES_REP	Default Sales Rep Role cannot be deleted.
CANT_DELETE_DEFAULT_STATUS	You cannot delete a default case status. Please select a new default first.
CANT_DELETE_DEFAULT_STATUS	You can't delete or inactivate that status because it is a set up as a default status. Please navigate to Sales Preferences and change that status
CANT_DELETE_DEFAULT_VALUE	You may not delete or inactivate that value because it is a default. Please select a new default first.
CANT_DELETE_DEFAULT_WEBSITE	The default Web site cannot be deleted.
CANT_DELETE_EMPL	This employee cannot be deleted because it has child entities
CANT_DELETE_ENTITY	This entity cannot be deleted because it has child items
CANT_DELETE_FIN_STATMNT_LAYOUT	This financial statement layout cannot be deleted because it is referred to by other layouts.
CANT_DELETE_FLDR	These predefined folders cannot be deleted
CANT_DELETE_HAS_CHILD_ITEM	This {1} cannot be deleted because it has child items

Error Code Returned	Long Description or Message
CANT_DELETE_INFO_ITEM	This information item cannot be deleted because it has child items
CANT_DELETE_ITEM	This item cannot be deleted because it has child items
CANT_DELETE_ITEM_LAYOUT	This item/category layout cannot be deleted because it is used by store tabs
CANT_DELETE_ITEM_TMPLT	This item/category template cannot be deleted because it is referred to by a theme or an item
CANT_DELETE_JOB_RESOURCE_ROLE	Default Job Resource Role cannot be deleted.
CANT_DELETE_LEGACY_CATEGORY	Legacy category cannot be removed
CANT_DELETE_LINE	This line cannot be deleted, because it is referred to by other records. Before removing this line, remove any discount or markup lines applied to it.
CANT_DELETE_MEDIA_ITEM	This media item cannot be deleted because it is being referenced by another item.
CANT_DELETE_MEMRZD_TRANS	This memorized transaction cannot be deleted because it referenced in transactions
CANT_DELETE_OR_CHANGE_ACCT	Special accounts cannot be deleted and their type cannot be changed
CANT_DELETE_PLAN_ASSGNMNT	Trying to delete plan assignment referenced by precalcs.
CANT_DELETE_PRESNTN_CAT	This presentation category cannot be deleted because it has subcategories
CANT_DELETE_RCRD	This {1} record cannot be deleted because it referenced by other records
CANT_DELETE_RCRD	This record cannot be deleted because it has {1}child records{2}
CANT_DELETE_RCRD	This record cannot be deleted because it is referenced by other records or it is used by one or more transactions.
CANT_DELETE_RCRD	This record cannot be deleted, because it is referred to by other records.
CANT_DELETE_RCRDS	Selected records could not be deleted because one or more of them are of a special type of account needed by {1}
CANT_DELETE_RCRDS	Selected records could not be deleted because one or more of them are referenced by other records.
CANT_DELETE_SITE_TAG	This site tag cannot be deleted because it is being used.
CANT_DELETE_SITE_THEME	This site theme cannot be deleted because it is being used
CANT_DELETE_SOLUTN	This solution cannot be deleted because it has been applied to support cases.
CANT_DELETE_STATUS_TYPE	You cannot delete the only status of type {1}
CANT_DELETE_SUBTAB	This subtab cannot be deleted because it is referred to by custom fields
CANT_DELETE_SYSTEM_NOTE	You cannot alter or delete a system logged note.

Error Code Returned	Long Description or Message
CANT_DELETE_TAX_VENDOR	This is a special tax vendor and cannot be deleted.
CANT_DELETE_TMPLT_RCRD	This template record cannot be deleted.
CANT_DELETE_TRANS	This transaction cannot be deleted because it is linked to one or more commission transactions. The commission authorizations due to this transaction need to be removed to be able to delete this transaction.
CANT_DELETE_TRANS	This transaction cannot be deleted because it is referred to by other transactions. It may be a bill or an invoice that has been paid or an expense that has been reimbursed.
CANT_DELETE_TRANS	This transaction cannot be deleted because it is referenced by an intercompany adjustment. The adjustment must be deleted first.
CANT_DELETE_TRAN_LINE	Failed to delete line {1}. This line is linked to another transaction.
CANT_DELETE_TRAN_LINES	Lines with partially recognized rev rec or amortization schedules can not be deleted.
CANT_DELETE_UPDATE_ACCT	This account cannot be deleted or changed because it is a special type of account needed by {1}
CANT_DELETE_URL	This third party conversion tracking URL cannot be deleted because it is reference by other records.
CANT_DELETE_VENDOR	This {1:Vendor} is related to a {2:Partner} which is eligible for commission and cannot be deleted.
CANT_DELETE_VENDOR	This vendor cannot be deleted because there are dependent items, such as a pending payment. If you wish to remove the payee, you must first delete all such dependent items.
CANT_DIVIDE_BY_ZERO	There is a divide by zero error in this search. It may be an error with a formula you have used. Please retry without the formula(s). If an error still occurs, please file with Customer Support. If it does not, please correct your formula. Typically that consists of taking the denominator and wrapping it in NULLIF(<denominator>,0).
CANT_DOWNLOAD_EXPIRED_FILE	This file has expired and can no longer be downloaded
CANT_EDIT_CHARGED_ORDER	You cannot charge an order that is already completely charged.
CANT_EDIT_CUST_LIST	Can not modify this Custom List because its entries are in use.
CANT_EDIT_CUST_PMT	This customer payment cannot be edited while it has an Automated Clearing House transmission in process.</TD></TR><TR><TD class=text> </TD></TR><TR><TD class=text> To view the status of customer payments with ACH transmissions, go to Transactions > View Electronic Funds Transfer Status.
CANT_EDIT_DPLYMNT_IN_PROGRESS	You cannot change or delete a deployment that is in progress or in the queue.
CANT_EDIT_DPLYMNT_IN_PROGRESS	You cannot edit a script deployment when it is being executed.
CANT_EDIT_FOLDER	Predefined folders cannot be updated.

Error Code Returned	Long Description or Message
CANT_EDIT_OLD_CASE	This case cannot be edited because it was closed {1} or more days ago.
CANT_EDIT_STANDARD_OBJ	Cannot alter standard dashboards
CANT_EDIT_STANDARD_OBJ	Cannot alter standard dashboard role maps
CANT_EDIT_STANDARD_OBJ	Cannot alter standard dashboard section maps
CANT_EDIT_STANDARD_OBJ	Cannot alter standard fields
CANT_EDIT_STANDARD_OBJ	Cannot alter standard forms or layouts
CANT_EDIT_STANDARD_OBJ	Cannot alter standard portlets
CANT_EDIT_STANDARD_OBJ	Cannot alter standard report snapshot layouts
CANT_EDIT_STANDARD_OBJ	Cannot alter standard roles
CANT_EDIT_STANDARD_OBJ	Cannot alter standard searches
CANT_EDIT_STANDARD_OBJ	Cannot alter standard sections
CANT_EDIT_STANDARD_OBJ	Cannot alter standard tabs
CANT_EDIT_STANDARD_OBJ	Cannot alter standard tasks
CANT_EDIT_STANDARD_OBJ	Cannot alter standard task categories
CANT_EDIT_STANDARD_OBJ	Cannot alter standard templates
CANT_EDIT_STANDARD_OBJ	Cannot Alter Standard Types
CANT_EDIT_STANDARD_OBJ	Cannot alter standard words
CANT_EDIT_TAGATA	The Receivable Tegata is linked to Invoices and is no longer editable
CANT_EDIT_TRAN	You cannot edit intercompany adjustments.
CANT_EDIT_TRAN	You cannot edit the account of a transaction line that is linked to others
CANT_EDIT_TRAN	You cannot edit this expense report
CANT_EDIT_TRAN	You cannot edit this transaction because it was automatically created by NetSuite's Payroll Service
CANT_ESTABLISH_LINK	Unable to establish link with {1}
CANT_FIND_BUG	Cannot locate the bug that was just entered (1)!

Error Code Returned	Long Description or Message
CANT_FIND_MAIL_ MERGE_ID	Mail Merge Id not found
CANT_FIND_RCRD	Could not find record with {1} = {2}
CANT_FIND_SAVED_ IMPORT	No saved import with internalId {1}
CANT_FIND_SOURCE_ AMORTZN_ACCT	The source account for the amortization schedule could not be determined.
CANT_FIND_UPS_REG_ FOR_LOC	No UPS registration was found for the location selected. Please select a different shipping item, or go to Setup > Set Up Shipping to register a UPS account for this location.
CANT_FULFILL_ITEM	An item receipt has been posted
CANT_INACTIVATE_ COMMSSN_PLAN	You cannot inactivate a plan that has commission payments that are pending authorization. Please clear the commission payments at Transactions > Authorize Commissions before inactivating this plan.
CANT_INACTIVE_ DEFAULT_SYNC_CAT	You cannot inactivate the default synchronization category.
CANT_INACTIVE_ DEFAULT_TMPLT	You cannot inactivate this template record because it is set up as a default template
CANT_LOAD_SAVED_ SEARCH_PARAM	Error loading saved search params
CANT_LOGIN_WITH_ OAUTH	A login operation or Request Level Credentials must not be used in conjunction with OAuth authorization{:do not translate 'login' or captilized words}
CANT_LOOKUP_FLD	Can not lookup field {1} by {2}
CANT_MAKE_CONTACT_ PRIVATE	Employee contacts cannot be made private
CANT_MAKE_CONTACT_ PRIVATE	Individual relationship contacts cannot be made private
CANT_MARK_SHIPPED	Item {1:item name} cannot be marked shipped because the remaining quantity on the linked purchase order does not match the remaining quantity on the sales order.
CANT_MERGE_EMPLS	employees can not be merged
CANT_MODIFY_APPRVD_ TIME	Time records can not be modified once they have been approved.
CANT_MODIFY_FULFILL_ STATUS	You may not change the fulfillable status of an item which has transactions associated with it.
CANT_MODIFY_ISSUE_ STATUS	The issue status '{1:issue status name}' cannot be changed from {2:base status} to {3:base status} because it is in use.
CANT_MODIFY_LOCKED_ FLD	You may not update or delete a locked custom field.
CANT_MODIFY_PARENT	Payments have been accepted from the top level parent. The top level parent can not be changed.

Error Code Returned	Long Description or Message
CANT_MODIFY_RCRD	You cannot add, update or delete accounting periods through Web Services APIs. Any changes must be made through NetSuite UI.
CANT_MODIFY_REV_REC	The value of Rev Rec on Rev Commit may not be modified for this transaction.
CANT_MODIFY_SUB	You cannot change the subsidiary of this entity because one or more transactions exist for this entity.
CANT_MODIFY_TAGATA	The Payable Tegata is no longer in Issued state and cannot be modified.'
CANT_MODIFY_TAGATA	The Receivable Tegata is no longer in Holding state and cannot be modified.
CANT_MODIFY_TEGATA	The Payable Tegata is linked to bills and cannot be modified.'
CANT_MODIFY_VOID_ TRANS	The G/L impact of a voided transaction cannot be changed.
CANT_MODIFY_VOID_ TRANS	You may not change the GL impact on a voided {1: transaction type}.
CANT_MOVE_REALIZED_ GAINLOSS	You cannot move a Realized Gain/Loss transaction to a date before either the source or the payment transaction.
CANT_PAY_TAGATA	Endorsed Tegata can only be paid on or after its maturity date.
CANT_PAY_TAGATA	Payable Tegata can only be paid on or after its maturity date.
CANT_PROCESS_IMG	Faceless PDF Library unable to process image. Image DPI:{1}.
CANT_RCEIV_BEFORE_ FULFILL	The item receipt for a transfer order line can not occur before the item fulfillment.
CANT_RCEIV_ITEM	A mark shipped fulfillment line has been processed against item {1:item name}. This item cannot be received.
CANT_RECEIVE_TAGATA	Receivable Tegata can only be collected on or after its maturity date.
CANT_REJECT_ORDER	You cannot reject this order because it has already been approved.
CANT_REMOV_ALL_ FULFILMNT_LINKS	You may not modify this sales order in such a way that it removes all links to any fulfillment. The modifications you made would leave the fulfillment {2} unlinked.
CANT_REMOV_ITEM_SUB	You may not remove a subsidiary from an item that is a member of an assembly, group, or kit item if the parent item is available in that subsidiary.
CANT_REMOVE_ACH_ PAY_METHOD	ACH payment methods cannot be removed
CANT_REMOVE_ APPROVAL	The accounting approval cannot be removed from this expense report because some of its lines have already been invoiced to the customer.
CANT_REMOVE_DOMAIn	You are trying to remove a domain that is referenced by {1} CRM template(s). Please first clear the domain from CRM templates before trying to remove it.
CANT_REMOVE_NEXUS	A nexus cannot be removed from a subsidiary if the nexus is associated with a transaction.
CANT_REMOVE_SCHDUL	You have attempted to remove an active schedule from a plan. Removing this participant is not permitted once commissions against the plan have been generated.
CANT_REMOVE_SUB	You cannot remove subsidiary: {1} because this record is used on a transaction for subsidiary: {1}.

Error Code Returned	Long Description or Message
CANT_REMOVE_SUB	You attempted to remove one or more subsidiaries from this item, but the item appears in at least one transaction in those subsidiaries. In order to remove a subsidiary from the item, make sure the item does not appear in any transactions for that subsidiary.
CANT_RESUBMIT_ FAILED_DPLYMNT	You cannot submit a deployment for execution whose status is set to Failed or Scheduled.
CANT_RETURN_FLD	Can not return field {1}. Reason: {2}
CANT_RETURN_USED_ GIFT_CERT	Used gift certificates can not be returned.
CANT_REV_REC_BODY_ AND_LINE	The Revenue Recognition fields must be specified at EITHER the transaction body or the item line level, and may NOT be specified at both levels.
CANT_REVERSE_AUTH	Card type doesn't allow reversals. No resolution. Authorization cannot be reversed.
CANT_SCHDUL_RECUR_ EVENT	Because the number of days in each month differs, recurring monthly events cannot be scheduled after the 28th.
CANT_SEND_EMAIL	Unable to send notification email
CANT_SEND_EMAIL	Unable to send notification email to support rep
CANT_SET_CLOSE_DATE	Unable to set expected close date of prospect/lead based on current estimates/ opportunities.
CANT_SET_INTERNAL_ID	You cannot set internalId with upsert.
CANT_SET_STATUS	Unable to set status of prospect/lead based on current estimates.
CANT_SWITCH_ROLES_ FROM_LOGIN	Role switching is not allowed from this login.
CANT_SWITCH_SHIP_ METHOD	Switching the shipping method to another carrier is an unsupported operation, because it requires reloading the item fulfillment form for that carrier.
CANT_UPDATE_ACCTNG_ PRDS	You cannot update accounting periods using SuiteScript or Web Services. Go to Setup > Accounting > Manage G/L > Manage Accounting Periods.
CANT_UPDATE_AMT	The amount on lines containing partially/fully recognized schedules can not be changed
CANT_UPDATE_ DYNAMIC_GROUP	You cannot update dynamic groups. Instead you must modify the saved search associated with the group
CANT_UPDATE_FLDR	These predefined folders cannot be updated
CANT_UPDATE_LINKED_ TRANS_LINES	You cannot update linked transaction lines
CANT_UPDATE_ PRODUCT_FEED	This item has multiple product feeds. Web Services schema version 2_6 or greater is required to modify product feeds for this item
CANT_UPDATE_RECRD_ HAS_CHANGED	Cannot update bug. Record has changed since you last retrieved it.
CANT_UPDATE_RECUR_ EVENT	Event <id {1}> contains recurrence patterns that are not supported in your client application. You are not allowed to update recurrence pattern on this event. Contact your software vendor for the latest Web Services upgrade.

Error Code Returned	Long Description or Message
CANT_UPDATE_ROOT_CATEGORY	Can not update root level website categories through Web Services.
CANT_UPDATE_STATUS_TYPE	You cannot update the only status of type {1}
CANT_VERIFY_CARD	Card Verify not supported. Retry request.
CANT_VOID_TRANS	You cannot void this transaction because it is linked to by one or more transactions such as payments. You must delete or void those transactions first
CARD_EXPIRED	Expired Card. Re-submit with valid expiration date.
CARD_ID_REQD	Card ID required. Provide a valid card ID.
CASE_ALREADY_ASSIGNED	This case cannot be grabbed because it is already assigned to another rep. To view the case, go back and click on the case number.
CASE_DSNT_EXIST	Case doesn't exist or no customer is associated with case.
CASE_NOT_GROUP_MEMBER	{1} this case record does not belong to your group.
CASH_SALE_EDIT_DISALLWD	This cash sale cannot be edited while it has an Automated Clearing House transmission in process.</TD></TR><TR><TD class=text> </TD></TR><TR><TD class=text> To view the status of cash sales with ACH transmissions, go to Transactions > View Electronic Funds Transfer Status.
CC_ACCT_REQD	You must have a credit card account to perform this operation.
CC_ACCT_REQD	You must have a credit card account to perform this operation. Click here to add one.
CC_ALREADY_SAVED	That credit card is already saved. Please use the saved credit card.
CC_EMAIL_ADDRESS_REQD	Please go back and provide an email address to CC store orders to.
CC_NUM_REQD	Please provide a credit card number.
CC_PROCESSOR_ERROR	An error occurred while processing the credit card. Please contact the merchant for assistance.
CC_PROCESSOR_ERROR	An unexpected error occurred while processing the credit card through Merchant e-Solutions (reason code = {1}). Please contact NetSuite support.
CC_PROCESSOR_NOT_FOUND	A suitable credit card processor was not found for this transaction.
CC_SECURITY_CODE_REQD	This transaction requires the Credit Card Security Code. Please enter the required value in the {1} field and re-submit.
CERT_AUTH_EXPD	CA expired on {1}
CERT_EXPD	Certificate expired on {1}
CERT_UNAVAILABLE	Certificate unavailable (most likely has not been presented by client)
CHANGE_PMT_DATE_AND_REAPPROVE	The payment is more than 30 days past due and has NOT been sent. Edit the payment to change the date and reapprove.
CHAR_ERROR	Character error on Line# {1} Column# {2} (Byte # {3}). {4}
CHECKOUT_EMAIL_REQD	Please go back and provide an email address to email checkout errors to.

Error Code Returned	Long Description or Message
CITY_REQD	City is a required field and it cannot be null or empty.
CLASS_ALREADY_EXISTS	A class already exists with that name. Go back, change the name and resubmit.
CLASS_OR_DEPT_OR_CUST_REQD	only one of class, cust, and dept can be non-null
CLEAR_AUTOCALC	For items that use the time phased replenishment method, you must clear the Auto-Calculate checkbox next to the Reorder Point and Preferred Stock Level fields. The mass update cannot be performed unless these settings are changed.
CLOSE_PREVIOUSE_PERIOD	Please close previous period before working on this one.
CLOSED_TRAN_PRD	The G/L impact of a transaction in a closed period cannot be changed.
CLOSED_TRAN_PRD	You cannot move a transaction to or from a closed period.
COGS_ERROR	COGS lines not cleaned up
COGS_ERROR	Cost of Goods Sold lines not in balance
COGS_ERROR	LIFO/FIFO COGS count does not equal the number of items requested COGS ERROR 9765 itemsLinked={1}, itemsTotal={2} kdoc={3}, nid={4}
COMMSSN_ALREADY_CALCLTD	You have attempted to remove an active sales participant from a plan. Removing this participant is not permitted once commissions against the plan have been generated.
COMMSSN_FEATURE_DISABLED	You have not enabled the Commissions feature.
COMMSSN_FEATURE_DISABLED	You have not enabled the Partner Commissions/Royalties feature.
COMMSSN_PAYROLL_ITEM_REQD	A commission payroll item must be added for each employee to be processed through payroll
COMP_DELETED_OR_MERGED	The company you try to attach the context to has been deleted or merged.
COMPANION_PROP_REQD	Error - Items do not have companion property (column) {1}
COMPANY_DISABLED	Please contact Accounts Receivable at {1} or 650.627.1316 to re-enable this company.
COMPANY_FLD_REQD	The Company field is required for COD shipments. Go to Setup > Accounting > Shipping, set the Company field on your FedEx Registration record and re-submit.
CONCUR_BILLPAY_JOB_DISALLWD	Your account currently has a bill pay approval job in progress. Only one bill pay approval job per account is allowed at a time. Please wait until this process completes before submitting another group of payments for approval. Visit the status page to track the progress of the current job.
CONCUR_BULK_JOB_DISALLWD	This Account is already running a bulk processing job. Please visit the status page to track the progress of the current job.

Error Code Returned	Long Description or Message
CONCUR_MASS_ UPDATE_DISALLWD	A mass update is currently running in this account. Please try again in a few minutes.
CONCUR_SEARCH_ DISALLWD	Search aborted by concurrent {1} search. Only one search may run at a time.
CONSLD_PRNT_AND_ CHILD_DISALLWD	A company can be a consolidated child or a consolidated parent but not both
CONTACT_ALREADY_ EXISTS	A contact record with this name already exists. Every contact record must have a unique name.
CONTACT_ALREADY_ EXISTS	A contact with the name [{1}] already exists
CONTACT_NOT_GROUP_ MEMBR	{1} this contact does not belong to your group.
COOKIES_DISABLED	You have disabled cookies from being stored on your computer or turned off per-session cookies. Please enable this feature and try again
COUNTRY_STATE_ MISMATCH	The country and state/province are mismatched, the country is {1} and the state/ province is {2}. Please enter a state/province short name that matches the country (see the "state" record for legal short names).
CREATEDFROM_REQD	Please enter a value for createdFrom.
CREDITS_DISALLWD	Credits Not Allowed. Contact Merchant e-Solutions to have credits enabled.
CRNCY_MISMATCH_ BASE_CRNCY	The currency you are registered to use is different from the base currency of this company.
CRNCY_NOT_UPDATED	The following currencies were not updated: {1}
CRNCY_RCRD_DELETED	This currency record has been deleted. You can create a new currency record at Lists > Currencies.
CRNCY_REQD	currency expected for pricing element
CSC_SETUP_REQD	To display the CSC field on the form, you must enable the "Use Card Security Code for Credit Card Transactions" preference, located on the Setup > Accounting Preferences task.
CSTM_FIELD_KEY_REQD	The specified custom field key is missing.
CSTM_FIELD_VALUE_ REQD	The specified custom field value is missing.
CSV_DELIMITER_ERROR	Values in the following CSV file(s) are not comma separated, and cannot be imported. Please reformat the file(s) and try again. For instructions, visit this help topic.
CUST_ARLEADY_HAS_ ACCT	Attempting to provision a new account to a customer with an existing account
CUST_CNTR_USER_ ACCESS_ONLY	This form is only accesible to customer center users.
CUST_LEAD_NOT_ GROUP_MEMBR	{1} this customer or lead does not belong to your group.

Error Code Returned	Long Description or Message
CYBERSOURCE_ERROR	The credit card transaction was denied by the issuing bank. Please try another card or contact the card issuer for more information.
CYBERSOURCE_ERROR	The credit card has expired or the expiration date does not match the date on file with the card issuer. Please correct the expiration date or try another card.
CYBERSOURCE_ERROR	The credit card transaction was denied due to insufficient funds. Please try another card or contact the card issuer for more information.
CYBERSOURCE_ERROR	The credit card transaction could not be completed because the issuing bank was not available. Please try another card or wait a few minutes and try again.
CYBERSOURCE_ERROR	Inactive card or card not authorized for card-not-present transactions. Please try another card or contact the card issuer for more information.
CYBERSOURCE_ERROR	The card has reached the credit limit. Please try another card or contact the card issuer for more information.
CYBERSOURCE_ERROR	Invalid card verification number. Please check to make sure you have provided the correct card verification number.
CYBERSOURCE_ERROR	Invalid credit card account number. Please check to make sure you have provided the correct credit card account number.
CYBERSOURCE_ERROR	The type of credit card provided is not accepted by this merchant. Please try another card or contact the merchant for more information.
CYBERSOURCE_ERROR	The type of credit card provided is not accepted by this merchant. Please try another card or contact the merchant for more information.
CYBERSOURCE_ERROR	Successful transaction.
CYBERSOURCE_ERROR	The request is missing one or more required fields. Possible action: See the reply fields missingField_0...N for which fields are missing. Resend the request with the complete information.
CYBERSOURCE_ERROR	One or more fields in the request contains invalid data. Possible action: See the reply fields invalidField_0...N for which fields are invalid. Resend the request with the correct information.
CYBERSOURCE_ERROR	The merchantReferenceCode sent with this authorization request matches the merchantReferenceCode of another authorization request that you sent in the last 15 minutes. Possible action: Resend the request with a unique merchantReferenceCode value.
CYBERSOURCE_ERROR	Error: General system failure. See the documentation for your CyberSource client (SDK) for information about how to handle retries in the case of system errors.
CYBERSOURCE_ERROR	Error: The request was received but there was a server timeout. This error does not include timeouts between the client and the server. Possible action: To avoid duplicating the order, do not resend the request until you have reviewed the order status in the Business Center. See the documentation for your CyberSource client (SDK) for information about how to handle retries in the case of system errors.
CYBERSOURCE_ERROR	Error: The request was received, but a service did not finish running in time. Possible action: To avoid duplicating the order, do not resend the request until you have reviewed the order status in the Business Center. See the documentation for your CyberSource client (SDK) for information about how to handle retries in the case of system errors.

Error Code Returned	Long Description or Message
CYBERSOURCE_ERROR	The issuing bank has questions about the request. You do not receive an authorization code programmatically, but you might receive one verbally by calling the processor. Possible action: Call your processor or the issuing bank to possibly receive a verbal authorization. For contact phone numbers, refer to your merchant bank information.
CYBERSOURCE_ERROR	Expired card. You might also receive this if the expiration date you provided does not match the date the issuing bank has on file. Possible action: Request a different card or other form of payment.
CYBERSOURCE_ERROR	General decline of the card. No other information provided by the issuing bank. Possible action: Request a different card or other form of payment.
CYBERSOURCE_ERROR	Insufficient funds in the account. Possible action: Request a different card or other form of payment.
CYBERSOURCE_ERROR	Stolen or lost card. Possible action: Review the customers information and determine if you want to request a different card from the customer.
CYBERSOURCE_ERROR	Issuing bank unavailable. Possible action: Wait a few minutes and resend the request.
CYBERSOURCE_ERROR	Inactive card or card not authorized for card-not-present transactions. Possible action: Request a different card or other form of payment.
CYBERSOURCE_ERROR	The card has reached the credit limit. Possible action: Request a different card or other form of payment.
CYBERSOURCE_ERROR	Invalid card verification number. Possible action: Request a different card or other form of payment.
CYBERSOURCE_ERROR	The customer matched an entry on the processors negative file. Possible action: Review the order and contact the payment processor.
CYBERSOURCE_ERROR	Invalid account number. Possible action: Request a different card or other form of payment.
CYBERSOURCE_ERROR	The card type is not accepted by the payment processor. Possible action: Request a different card or other form of payment. Also, check with CyberSource Customer Support to make sure your account is configured correctly.
CYBERSOURCE_ERROR	General decline by the processor. Possible action: Request a different card or other form of payment.
CYBERSOURCE_ERROR	There is a problem with your CyberSource merchant configuration. Possible action: Do not resend the request. Contact Customer Support to correct the configuration problem.
CYBERSOURCE_ERROR	The requested amount exceeds the originally authorized amount. Occurs, for example, if you try to capture an amount larger than the original authorization amount. This reason code only applies if you are processing a capture through the API. See Using the API for Captures and Credits. Possible action: Issue a new authorization and capture request for the new amount.
CYBERSOURCE_ERROR	Processor failure. Possible action: Tell the customer the payment processing system is unavailable temporarily, and to try their order again in a few minutes.
CYBERSOURCE_ERROR	The authorization has already been captured. This reason code only applies if you are processing a capture through the API. See Using the API for Captures and Credits. Possible action: No action required.

Error Code Returned	Long Description or Message
CYBERSOURCE_ERROR	The requested transaction amount must match the previous transaction amount. This reason code only applies if you are processing a capture or credit through the API. See Using the API for Captures and Credits. Possible action: Correct the amount and resend the request.
CYBERSOURCE_ERROR	The card type sent is invalid or does not correlate with the credit card number. Possible action: Ask your customer to verify that the card is really the type that they indicated in your Web store, then resend the request.
CYBERSOURCE_ERROR	The request ID is invalid. This reason code only applies when you are processing a capture or credit through the API. See Using the API for Captures and Credits. Possible action: Request a new authorization, and if successful, proceed with the capture.
CYBERSOURCE_ERROR	You requested a capture through the API, but there is no corresponding, unused authorization record. Occurs if there was not a previously successful authorization request or if the previously successful authorization has already been used by another capture request. This reason code only applies when you are processing a capture through the API. See Using the API for Captures and Credits. Possible action: Request a new authorization, and if successful, proceed with the capture.
CYBERSOURCE_ERROR	The capture or credit is not voidable because the capture or credit information has already been submitted to your processor. Or, you requested a void for a type of transaction that cannot be voided. This reason code applies only if you are processing a void through the API. See Using the API for Voids for information about voids. Possible action: No action required.
CYBERSOURCE_ERROR	You requested a credit for a capture that was previously voided. This reason code applies only if you are processing a void through the API. See Using the API for Voids for information about voids. Possible action: No action required.
CYBERSOURCE_ERROR	Error: The request was received, but there was a timeout at the payment processor. Possible action: To avoid duplicating the transaction, do not resend the request until you have reviewed the transaction status in the Business Center.
CYBERSOURCE_ERROR	The authorization request was approved by the issuing bank but declined by CyberSource based on your Smart Authorization settings. Possible action: Do not capture the authorization without further review. Review the ccAuthReply_avsCode, ccAuthReply_cvCode, and ccAuthReply_authFactorCode fields to determine why CyberSource rejected the request.
CYBERSOURCE_ERROR	Unable to process credit card transaction. The code returned from CyberSource {1} is not a recognized reason code. Please contact NetSuite support.
CYCLE_IN_PROJECT_PLAN	The changes made to this entity have cause a cycle in the project plan. Select a different parent and/or predecessors to avoid the cycle.
DASHBOARD_LOCKED	Your dashboard has been set up and locked by an administrator. Please contact them for details.
DATA_MUST_BE_UNIQUE	The update failed because every entry in this column must be unique.
DATA_REQD	You need to provide a proper value for the required field: {1}.
DATA_REQD	You are missing the following required field(s):{1}
DATE_EXPECTED	You entered '{1}' into a field where a calendar date was expected.\nPlease go back and change this value to the correct date.
DATE_PARAM_REQD	missing date parameter

Error Code Returned	Long Description or Message
DATE_PRD_MISMATCH	Your transaction date does not fall between the start and end dates of your accounting period.
DEFAULT_CUR_REQD	Default currency cannot be null
DEFAULT_EXPENSE_ACCT_REQD	A default expense account must be specified in order to activate items on the list.nGo to Setup > Set Up Payroll and click the Default Accounts subtab.nIn the Payroll Expenses Account field, choose a default general ledger account for your payroll expenses. Then, click Save.
DEFAULT_ISSUE_OWNER_REQD	There is no default owner for the issue role {1}. This operation cannot be completed until this is corrected.
DEFAULT_LIAB_ACCT_REQD	A default liability account must be specified in order to activate items on the list.nGo to Setup > Set Up Payroll and click the Default Accounts subtab.nIn the Payroll Liabilities Account field, choose a default general ledger account for your payroll liabilities. Then, click Save.
DEFAULT_ROLE_REQD	Login Failed because you do not have a default role for the company and email entered. Please Try Again.
DEFAULT_TYPE_DELETE_DISALLWD	You cannot delete default types
DEFERRAL_ACCT_REQD	Lines with amortization templates must have a deferral account.
DEFERRAL_ACCT_REQD	Lines with revenue recognition templates must have a deferral account.
DEFERRED_REV_REC_ACCT_REQD	The {1} item does not have a Deferred Revenue Account specified. Please assign the item a Deferred Revenue Account using the standard User Interface, and then re-import the transaction.
DEPT_IN_USE	Your classes cannot be converted to departments because your existing department records are referred to by transactions or other records. These department records cannot be overwritten.
DFRNT_SWAP_PRICE_LEVELS_REQD	Please select different price levels to swap prices.
DISALLWD_IP_ADDRESS	The specified IP address rules must allow the login of your current IP Address. Your current IP address is {1}. For information on entering IP address rules, click Help at the top of the page.
DISCOUNT_ACCT_SETUP_REQD	Please Set Up Discount Accounts first.
DISCOUNT_DISALLWD	You have attempted to save this transaction with one or more discounts and where all items have Permit Discount = Never. You must change one of the items to permit a discount, add a new item without the restriction or remove the discount from the transaction.
DISCOUNT_DISALLWD_VSOE	Posting discounts are not allowed on items in VSOE bundles.
DISCOUNT_EXCEED_TOTAL	Discount can not exceed item total.
DISTRIB_REQD_ONE_DAY_BFORE	All items must be distributed at least one day before they may be transferred.
DOMAIN_IN_USE	The domain {1} is already in use

Error Code Returned	Long Description or Message
DOMAIN_WEBSITE_REQD	Please select a Web Site for domain {1}
DROP_SHIP_ERROR	The following error occurred when updating the quantity on the drop ship {1:transaction type}: {2:error message}
DROP_SHIP_ERROR	The transaction was successfully saved, but an error occurred while running user events and email alerts after updating the drop ship {1:transaction type}(s)
DROP_SHIP_ERROR	The transaction was successfully saved, but an error occurred while running user events after updating the drop ship {1:transaction type}(s)
DROP_SHIP_OR_ SPECIAL_ORD_ALLWD	Items can be Drop Ship or Special Order but not both
DUE_DATE_BFORE_ START_DATE	Due date occurs before start date
DUE_DATE_REQD	Please enter a value for {1} Due Date
DUP_ACCT_NAME	The account name you have chosen is already used. Go back, change the name and resubmit.
DUP_ACCT_NOT_ALLWD	You may not use duplicate accounts on an item.
DUP_ACCT_NUM	The account number you have chosen is already used.
DUP_ACCT_NUM	The account number you have chosen is already used. Go back, change the number and resubmit.
DUP_ACCT_ON_TRANS	This transaction has duplicate accounts. The main line of the transaction and the line labeled '{1}' both use the account named '{2}'.
DUP_BIN	A bin already exists with that name. Go back, change the name, and resubmit.
DUP_BIN	There is already another bin with that number. Please choose a bin number that is not used by another bin.
DUP_BUNDLE_IN_ACCT	That bundle has already been copied or installed in this account.
DUP_BUNDLE_IN_ACCT	You cannot install this bundle as it is a copy of bundle {1} that you previously installed.
DUP_CATEGORY	This category already exists
DUP_CATEGORY_NAME	A category already exists with that name. Go back, change the name and resubmit.
DUP_COLOR_THEME	This color theme already exists
DUP_CSTM_FIELD	This custom field already exists
DUP_CSTM_LAYOUT	This custom layout already exists
DUP_CSTM_LIST	There is already a Custom List or Custom List element with that name
DUP_CSTM_RCRD	There is already a Custom Record with that name
DUP_CSTM_RCRD_ENTRY	There is already a Custom Record Entry with that name
DUP_CSTM_TAB	This custom tab already exists
DUP_EMPL_EMAIL	There is already an employee with external access to this account using that email address. All employees with external access must have a unique email address for login purposes. Go back, change the email address and resubmit.

Error Code Returned	Long Description or Message
DUP_EMPL_ENTITY_NAME	There is already an employee with external access to this account using that entity name. All employees with external access must have a unique entity name for login purposes. Go back, change the entity name and resubmit.
DUP_EMPL_TMPLT	There is already an employee template with that name. Go back, change the template name and resubmit.
DUP_ENTITY	This entity already exists.
DUP_ENTITY_EMAIL	There is already an external entity (eg. customer, vendor, or employee) with access to this account using that email address. All external entities with access must have a unique email address for login purposes.
DUP_ENTITY_NAME	There is already an external entity (eg. customer, vendor, or employee) with access to this account using that entity name. All external entities with access must have a unique entity name for login purposes.
DUP_FEDEX_ACCT_NUM	There is an existing NetSuite registration for FedEx account number {1}.
DUP_FINANCL_STATMNT_LAYOUT	This financial statement layout already exists.
DUP_INFO_ITEM	This information item already exists
DUP_ISSUE_NAME_OR_NUM	You cannot set {1:issue record name} {2:issue number} to be a duplicate of itself or one of its duplicates.
DUP_ITEM	Uniqueness error - there is already an item with that name or name/parent combination.
DUP_ITEM_LAYOUT	This item/category layout already exists
DUP_ITEM_NAME	There is already an item with that name. . Go back, change the name and resubmit.
DUP_ITEM_OPTION	A child item child with that combination of options already exists
DUP_ITEM_TMPLT	This item/category template already exists
DUP_MATRIX_OPTN_ABBRV	Matrix option '{1}' already uses that abbreviation. Please choose another.
DUP_MEMRZD_TRANS	There is already a Memorized Transaction with that name. Go back, change the name and resubmit.
DUP_NAME	That name is already in use. Go back, change the name and resubmit.
DUP_PAYROLL_ITEM	There is already a payroll item named {1}
DUP_PRESNTN_CAT	This presentation category already exists
DUP_RCRD	A {1} already exists with that name. Go back, change the name and resubmit.
DUP_RCRD	Matched more than one record (internalIds {1} and {2})
DUP_RCRD	This record already exists
DUP_RCRD_LINK	Link to that record already exists

Error Code Returned	Long Description or Message
DUP_SALES_TAX_ITEM	You have entered a duplicate Sales Tax Item. Go back, change the name, city, state or zip code and resubmit.
DUP_SHIPPING_ITEM	You have entered a duplicate Shipping Item. Go back, change the name and resubmit.
DUP_SHORT_NAME	Duplicate short name
DUP_SITE_THEME	This site theme already exists
DUP_SOURCE_ACCT	Duplicate source accounts are not allowed.
DUP_TAX_CODE	You have entered a duplicate Tax Code. Go back, change the name and resubmit.
DUP_TAX_CODE	You have entered a duplicate Tax Code. Go back, change the name, city, state or zip code and resubmit.
DUP_TRACKING_NUM	You entered the following tracking number twice: {1}. Note that a single tracking number may not contain spaces or commas. A space or comma will be interpreted as the separator between different tracking numbers. For example, '1029 3847 465' will be interpreted as 3 different tracking numbers. It should be entered without spaces: '10293847465'.
DUP_TRANS	Duplicate Trans. Unable to locate, no match.
DUP_UPS_ACCT_NUM	There is an existing NetSuite registration for UPS account number {1}.
DUP_VENDOR_EMAIL	There is already a vendor with external access to this account using that email address. All vendors with external access must have a unique email address for login purposes. Go back, change the email address and resubmit.
DUP_VENDOR_NAME	There is already a vendor using that entity name. All vendors must have a unique entity name. Go back, change the entity name and resubmit.
DUPLICATE_INVENTORY_NUM	Duplicate inventory number found in entry: {1}
DUPLICATE_INVENTORY_NUM	Duplicate inventory number found on different lines of transaction
DUPLICATE_KEYS	This record contains duplicated key or keys. Please correct it before next update.
DUPLICATE_NAME_FOR_PRD	Please choose a different period name. "{1}" is already taken.
DUPLICATE_NAME_FOR_ROLE	Please choose a different role name. "{1}" is already taken.
EARNING_ITEM_REQD	At least one employee in this payroll has no earning items. Please make sure that every employee has at least one earning item.
EBAY_FEE_ERROR	An error has occurred while getting the fees for this eBay listing. Please try your request again in a few minutes.
EBAY_TMPLT_ERROR	You must select a Theme and Layout to preview the eBay Description template.
EDITION_DSNT_SUPRT_WORLDPAY	WorldPay is not supported in this edition.

Error Code Returned	Long Description or Message
EIN_OR_TIN_REQD	You must set either the Employer Identification Number (EIN) or SSN/TIN (Social Security Number, Tax ID Number) to complete this fulfillment. Please go to Setup > Company Information to set either of these fields and re-submit.
EMAIL_ADDRS_REQD	Please enter your email address
EMAIL_ADDRS_REQD_TO_NOTIFY	Please enter an email address for this company. A notification email will be sent when this case record is saved.
EMAIL_ADDRS_REQD_TO_NOTIFY	The recipient you are sending this email to does not have an email address. Please enter one and try again
EMAIL_REQ_HANDLER_ERROR	an error occurred while instantiating an email requesthandler
EMAIL_REQ_HANDLER_ERROR	An Error occurred while performing system-level validation of email request for <{1}> from <{2}>
EMAIL_REQ_HANDLER_ERROR	An Error occurred while POSTing data into requestHandlder: {1}
EMAIL_REQ_HANDLER_ERROR	an error occurred while servicing an email requesthandler in state: {1}
EMAIL_REQD	You must enter a valid email address in order to email the transaction.
EMAIL_REQD_ACCT_PROVISION	Cannot provision an account without an Email address for this customer: Was external access granted?
EMPL_IN_USE	You can't delete this employee, as commissions have been calculated for this employee.
EMPL_IN_USE	You can't delete this employee, as it is or has been referenced by other employees as a supervisor.
ERROR_DELETE_CARD_DATA	Failed to delete card data. Retry request.
ERROR_IN_TERRITORY_ASSGNMNT	Error Performing Initial Round_Robin Assignment for Territory: {1}
ERROR_IN_TERRITORY_ASSGNMNT	Error Performing Round_Robin Assignment for Territory: {1}
ERROR_PRCSSNG_TRANS	There were errors processing the selected transactions. Please process them individually for more information.
ERROR_REFUND_TRANS	Failed to refund International transaction. Retry request.
ERROR_REVERSE_AUTH	Failed to reverse International authorization. Retry request.
ERROR_SENDING_TRAN_EMAIL	The transaction was entered successfully, but an unexpected error occurred while sending the transaction email {1}
ERROR_VOID_TRANS	Failed to void International transaction. Retry request.
EVENT_ID_NOT_FOUND	Event ID not found
EXCEEDED_MAX_ALLWD_LOC	You have reached the maximum allowance of {1} location records. If you need to create additional location records, please contact our NetSuite Customer Support team for assistance

Error Code Returned	Long Description or Message
EXCEEDED_MAX_CONCUR_RQST	The maximum number of concurrent requests has been exceeded. Please try your request again when an existing session has completed.
EXCEEDED_MAX_EMAILS	This account has {1} more bulk emails that can be sent. If you would like to purchase an additional block of emails, please contact your account manager.
EXCEEDED_MAX_EMAILS	The merge exceeds the number of bulk merge emails allotted to your account this year. This account has {1} more bulk emails that can be sent this year. Please contact your NetSuite account manager to purchase additional block of emails.
EXCEEDED_MAX_EMAILS	This campaign email event exceeds the number of emails ({1}) that can be sent per event without setting up a default campaign domain or specifying one on the campaign email template.
EXCEEDED_MAX_EMAILS	This merge operation exceeds the number of emails ({1}) that can be sent per execution without setting up a bulk merge domain or specifying one on the email template.
EXCEEDED_MAX_EMAILS	You cannot schedule more than {1} emails per year. If you'd like to purchase an additional block of emails, please contact your account manager.
EXCEEDED_MAX_FEATURED_ITEMS	There is a limit of {1} featured items on this page.
EXCEEDED_MAX_FIELD_LENGTH	Address line 1 cannot exceed 35 characters. Please check the shipper and recipient address to ensure the "Address 1" field is a maximum of 35 characters.
EXCEEDED_MAX_FIELD_LENGTH	Address line 2 cannot exceed 35 characters. Please check the shipper and recipient address to ensure the "Address 2" field is a maximum of 35 characters.
EXCEEDED_MAX_FIELD_LENGTH	The field {1} contained more than the maximum number ({2}) of characters allowed.
EXCEEDED_MAX_FIELD_LENGTH	The string "{1}" contained more than the maximum number of characters allowed.
EXCEEDED_MAX_FIELD_LENGTH	Too many characters for a field
EXCEEDED_MAX_MATRIX_OPTNS	The total combination of subitems you have selected exceeds the maximum allowed of 2000. Please choose fewer options on the matrix tab.
EXCEEDED_MAX_MATRIX_OPTNS	The total combination of subitems you have selected exceeds the maximum allowed of 2000. Please choose fewer options on the matrix tab.
EXCEEDED_MAX_PDF_ELEMENTS	There is a maximum of 100 custom elements allowed on a PDF layout.
EXCEEDED_MAX_PDF_EXPORT_COL	PDF Export is limited to 30 columns.
EXCEEDED_MAX_PIN_RETRIES	PIN entered incorrectly too often.
EXCEEDED_MAX_RCRD	You have reached the maximum allowance of {1} {2} records. If you need to create additional {2} records, please contact our NetSuite Customer Support team for assistance
EXCEEDED_MAX_REPORT_COL	The option you selected in the Column field results in a report that exceeds the maximum number of columns allowed. Please select additional filters or select a shorter date/period range.

Error Code Returned	Long Description or Message
EXCEEDED_MAX_REPORT_ROWS	Reports are limited to {1} rows. Please narrow your results.
EXCEEDED_MAX_REPORT_SIZE	The results of this report are too large. Please narrow your results.
EXCEEDED_MAX_SESSIONS	Maximum active sessions exceeded. Please wait 5 minutes and login again.
EXCEEDED_MAX_SHIP_PACKAGE	The maximum number of custom shipping packages has been exceeded: {1}. Please reduce item quantities to generate fewer packages, or enter the packages manually.
EXCEEDED_MAX_TIME	The operation has exceeded maximum allowed time for completion. Operation aborted.
EXCEEDED_MAX_TRANS_LINES	Transactions may not contain more than {1} lines.
EXCEEDED_MAX_USERS_ALLWD	The changes you have made to this employee's access have caused you to exceed either your Full Access or Employee Center allowance. To make these changes you must either adjust the number of employees assigned the Employee Center role or contact your account representative to purchase additional licenses.
EXCEEDED_MAX_USERS_ALLWD	Assigning this role would exceed your Full Access licenses ({1}). To assign this role, you must remove another employee's full access roles or contact your account representative to purchase additional licenses.
EXCEEDED_MAX_USERS_ALLWD	The changes you have made to this employee's access have caused you to exceed either your Full Access or Retail User allowance. To make these changes you must either adjust the number of employees assigned the Retail Clerk role or contact your account representative to purchase additional licenses.
EXCEEDED_PER_TRANS_MAX	Exceeded per transaction maximum on account {1}
EXCEEDED_RQST_SIZE_LIMIT	You have exceeded the permitted request size limit ({1})
EXCEEDS_ALLWD_LICENSES	Adding access for this user exceeds the number of licenses you have purchased. To add another user, you must first remove access from an existing user or contact NetSuite to purchase additional licenses.
EXCEEDS_ALLWD_LICENSES	Adding a {1} would exceed the number of licenses you have purchased. Please contact NetSuite for additional licenses.
EXPENSE_ENTRY_DISALLWD	{1} does not allow expense entry.
EXPIRED_SEARCH_CRITERIA	Your search criteria expired. The criteria for a given search generally expire after 15 minutes of inactivity. Please return to the search definition page and re-submit your search.
EXT_CAT_LINK_SETUP_REQD	Error - you have not properly set up links from your External Catalog Site back into {1}!
EXTERNALID_NOT_SUPPORTED	Field {1} does not support externalId
EXTERNALID_REQD	This operation requires a value for externalId.

Error Code Returned	Long Description or Message
FAILED_FEDEX_LABEL_VOID	Failed FedEx Label Void
FAILED_FORM_VALIDATION	Form validation failed. You cannot submit this record.
FAILED_UPS_LABEL_VOID	Failed UPS Label Void
FAX_NUM_REQD	You must enter a fax number.
FAX_NUM_REQD	You must enter a fax number for this recipient before performing a fax merge operation.
FAX_NUM_REQD	You must enter a valid fax number in order to fax the transaction.
FAX_SETUP_REQD	Before you can send faxes, you need to go to the Set Up Printing, Fax & Email page and set up the fax service.
FEATURE_DISABLED	You do not have the correct features enabled to search on {1}.
FEATURE_DISABLED	The "{1}" feature is not enabled in your {2} account.
FEATURE_DISABLED	The feature '{1}' required to access this page is not enabled in this account.
FEATURE_UNAVAILABLE	{1} Trial does not allow access to this feature. If you would like more information about this feature, please contact your account manager.
FEATURE_UNAVAILABLE	Error - This business does not have the External Catalog Site feature enabled.
FEATURE_UNAVAILABLE	Test Drive does not allow access to this feature. If you would like more information about this feature, please contact your account manager.
FEATURE_UNAVAILABLE	That feature is only available to Plus users
FEATURE_UNAVAILABLE	The {1} feature is not available to your company.
FEATURE_UNAVAILABLE	This feature is not available to your company.
FED_ID_REQD	Must have Federal Identification Number to process 1099-MISC forms.
FED_WITHHOLDING_REQD	Your employee record does not have current Federal Withholding information. Please contact your supervisor to set up your record with the appropriate information.
FEDEX_ACCT_REQD	The FedEx Account Number has not been set.
FEDEX_CANT_INTEGRATE_FULFILL	The fulfillment cannot be integrated with {1} because the Shipping Integration Carrier is set to UPS.
FEDEX_DROPOFF_TYP_REQD	The FedEx Dropoff Type has not been set.
FEDEX_INVALID_ACCT_NUM	This account number was not recognized by FedEx. Please re-enter your account number, or contact FedEx to open a new account.
FEDEX_ITEM_CONTENTS_REQD	For international shipments, {1} requires specific information about the item contents.
FEDEX_METER_NOT_RETRIEVED	A FedEx Meter Number was not retrieved for account number {1}. Please try your request again in a few minutes.
FEDEX_METER_REQD	The FedEx Meter Number has not been set.

Error Code Returned	Long Description or Message
FEDEX_ONE_PACKG_ ALLWD	The selected FedEx service allows only one package per fulfillment. If more than one package is required, please break up the shipment into multiple fulfillments of one package each.
FEDEX_ORIGIN_ COUNTRY_US_REQD	The origin country must be United States (US) for all Item Fulfillments when using a FedEx shipping method.
FEDEX_RATING_SRVC_ UNAVAILBL	The FedEx rating services application is currently unavailable. Please try your request again in a few minutes.
FEDEX_REG_NOT_ FOUND	A valid FedEx Registration was not found for the specified location:
FEDEX_SHIP_SRVC_REQD	The FedEx Shipping Service has not been set.
FEDEX_SHIP_SRVC_ UNAVAILBL	The FedEx shipping services application is currently unavailable. Please try your request again in a few minutes.
FEDEX_UNSUPRTD_ ORIGIN_COUNTRY	The origin country {1} is currently not supported for Item Fulfillments when using a FedEx shipping method.
FEDEX_USD_EXCHANGE_ RATE_REQD	Cannot retrieve FedEx realtime rates: USD Exchange Rate is required when requesting FedEx realtime rates.
FEDEX_VOID_ERROR	The FedEx Void failed due to a system error.
FIELD_CALL_DATE_REQD	Missing Required Field: Call Date
FIELD_DEFN_REQD	Field definition not found
FIELD_NOT_SETTABLE_ ON_ADD	You are not allowed to set the nsKey for a record
FIELD_PARAM_REQD	Please enter a value for {1}
FIELD_PARAM_REQD	Please enter values for {1}.
FIELD_REQD	Mandatory Field Missing
FIELD_REQD	You must first select a field
FILE_ALREADY_EXISTS	A file with the same name already exists in the selected folder.
FILE_ALREADY_EXISTS	Note: You are attempting to upload a file with a name matching an existing file in the selected folder. Please rename this file or select another folder, and then upload your file.
FILE_DISALLWD_IN_ ROOT_FLDR	You attempted to copy a file to the root directory. Only folders can exist in the root directory.
FILE_DISALLWD_IN_ ROOT_FLDR	You attempted to move a file to the root directory. Only folders can exist in the root directory.
FILE_MISSING	File Missing
FILE_NOT_DOWNLOADA BLE	Illegal request for a file that isn't downloadable
FILE_NOT_FOUND	File/Media Item {1} not found.
FILE_NOT_FOUND	File not found. Please try your download again.
FILE_REQD	You must enter a file before submitting this form.
FILE_REQD	You must upload a file before creating this media item

Error Code Returned	Long Description or Message
FILE_UPLOAD_IN_PROGRESS	Files are currently being uploaded to this account.
FILTER_BY_AMT_REQD	Please enter an amount to filter by.
FINANCE_CHARGE_SETUP_REQD	Please set Finance Charge Preferences first.
FIRST_LAST_NAMES_REQD	Please enter both your first and last name.
FIRST_QTY_BUCKET_MUST_BE_ZERO	Quantity defined for first quantity bucket must be zero
FLD_VALUE_REQD	Results are incomplete. You must provide a value for field {1}.
FLD_VALUE_TOO_LARGE	Value for field {1} is too large to be processed.
FOLDER_ALREADY_EXISTS	A folder with the same name already exists in the selected folder.
FORM_RESUBMISSION_REQD	You have logged in to a different user since you navigated to this form. You must re-submit this form as the new user.
FORM_SETUP_REQD	No appropriate forms are enabled for this role. Please contact your Administrator.
FORM_UNAVAILBL_ONLINE	This form is not available online
FORMULA_ERROR	Your formula has an error in it. It could resolve to the wrong datatype, use an unknown function, or have a syntax error. Please go back, correct the formula, and re-submit.
FRIENDLY_NAME_REQD	Missing Friendly Name. Friendly Name is a required field and it cannot be null or empty.
FULFILL_REQD_FIELDS_MISSING	For the listed items, please edit the item record and provide values for the specified fields, and retry the fulfillment.
FULFILL_REQD_FIELDS_MISSING	The {1} field is required to complete this fulfillment. Please return to the International tab on the item fulfillment and provide a value for the specified field and retry the fulfillment.
FULFILL_REQD_PARAMS_MISSING	Could not perform operation '{1}' since {2} parameter was not set.
FULL_DISTRIB_REQD	You must fully distribute all {1} numbers for {1} numbered items.
FULL_USERS_REQD_TO_INTEGRATE	Only full {1} users can integrate with partners.
FX_MALFORMED_RESPONSE	Received malformed response from Foreign Exchange source.
FX_RATE_REQD_FEDEX_RATE	Cannot retrieve {1} realtime rates: {2} Exchange Rate is required when requesting {3} realtime rates.
FX_TRANS_DISALLWD	FX transactions not accepted for this account. Contact Merchant e-Solutions.
GETALL_RCRD_TYPE_REQD	The getAll record type is required.

Error Code Returned	Long Description or Message
GIFT_CERT_AMT_EXCEED_AVAILBL	Gift certificate redemption amount exceeds available amount on the gift certificate
GIFT_CERT_AUTH_ALREADY_EXISTS	Gift certificate authorization code {1} already exists
GIFT_CERT_CAN_BE_USED_ONCE	A gift certificate may only be used once on a transaction
GIFT_CERT_CODE_REQD	Gift certificate codes are missing
GIFT_CERT_CODE_REQD	Missing gift certificate authorization code(s). Please go back and enter authorization codes on the {1}.
GIFT_CERT_CODE_REQD	You must specify a gift certificate code.
GIFT_CERT_IN_USE	Another user is using gift certificate {1}
GIFT_CERT_IN_USE	Gift certificate code {1} is already in use
GIFT_CERT_INVALID_NUM	Gift certificate numbers may not contain the '{1}' character.
GROUP_DSNT_EXIST	That group does not exist
GROUP_REQD	You cannot perform a bulk merge operation with an empty group
GROUP_TYPE_REQD	The group type is required.
GRTR_QTY_PRICE_LEVEL_REQD	Each quantity pricing level must be greater than the previous quantity pricing level.
ILLEGAL_ID	Illegal ID. Please enter a name.
ILLEGAL_PERIOD_STRUCTURE	Illegal period structure. Date {1} is in multiple periods.
INACTIVE_CC_PROFILE	The credit card processing profile provided is inactive.
INACTIVE_RCRD_FOR_ROLE	The record for this role has been made inactive.
INAVLID_FILE_TYP	A change has been made to this file's format. You cannot upload this type of file.
INAVLID_FILE_TYP	You attempted to upload a restricted file type. Please try again with a selection from the list below:
INAVLID_ITEM_TYP	Invalid item type [{1}] for item [{2}].
INAVLID_PRICING_MTRX	Invalid Quantity Pricing Matrix for quantity level {1} : Quantity {2}, Base Price {3}
INCOMPATIBLE_ACCT_CHANGE	The account change you have made is incompatible with old transactions. If you need to swap two accounts, you need to do it in 3 steps. For example, to change the income and asset accounts for an item: (1) Change the income account to a temporary account and save (2) Change asset account to the old income account and save (3) Change the income to the old asset account and save Please contact customer support if you need assistance with this.
INCOMPATIBLE_ACCT_CHANGE	The account change you have made is incompatible with old transactions. Please either change the account selection appropriately or do not request to update past transactions.

Error Code Returned	Long Description or Message
INCOMPLETE_BILLING_ADDR	Billing address is incomplete.
INCOMPLETE_FILE_UPLOAD	The upload did not complete correctly. Please try uploading the file again. If you have repeatedly received this error message, please send mail to {2} Technical Support.
INCRCT_ORD_INFO	The order contains incorrect information and was not placed.
INITIALIZE_ARG_REQD	The initialize reference id is required.
INITIALIZE_ARG_REQD	The initialize reference type is required.
INITIALIZE_ARG_REQD	The initialize type is required.
INITIALIZE_AUXREF_REQD	The initialize auxReference type is required.
INSTALL_SCRIPT_ERROR	Installation Script Error
INSUFCNT_NUM_PRDS_FOR_REV_REC	Not enough accounting periods in range specified for revenue recognition.
INSUFCNT_OPEN_PRDS_FOR_REV_REC	Not enough open accounting periods available for revenue recognition.
INSUFFICIENT_CHARS_IN_SEARCH	Global searches must contain at least three characters to prevent excessive matches.
INSUFFICIENT_FLD_PERMISSION	You are attempting to read an unauthorized field: {1}
INSUFFICIENT_FLD_PERMISSION	You cannot access this search because it includes restricted fields. Please contact your administrator.
INSUFFICIENT_FUND	Decline. Insufficient funds.
INSUFFICIENT_PERMISSION	For security reasons, only an administrator is allowed to edit an administrator record.
INSUFFICIENT_PERMISSION	Global search is not permitted from this role.
INSUFFICIENT_PERMISSION	Insufficient privileges
INSUFFICIENT_PERMISSION	Your issue DB access has been inactivated. Please contact your issue DB administrator.
INSUFFICIENT_PERMISSION	Your current login role does not have an associated Issue Role. Please change to a different role or contact your Issue administrator.
INSUFFICIENT_PERMISSION	Only the owner can make a contact private
INSUFFICIENT_PERMISSION	Only the super user can update or delete bug entries
INSUFFICIENT_PERMISSION	Permission error: you may not edit this role.
INSUFFICIENT_PERMISSION	Permission Violation: partners do not have access to this report.

Error Code Returned	Long Description or Message
INSUFFICIENT_ PERMISSION	Permission Violation: partners may not delete saved reports.
INSUFFICIENT_ PERMISSION	Permission Violation: You cannot delete saved reports not created by yourself.
INSUFFICIENT_ PERMISSION	Test Drive does not allow access to this feature. If you would like more information about this feature, please contact your account manager.
INSUFFICIENT_ PERMISSION	The restriction settings on your role deny you access to this item.
INSUFFICIENT_ PERMISSION	This folder does not exist or you do not have permission to access this folder.
INSUFFICIENT_ PERMISSION	This folder does not permit the direct addition of files
INSUFFICIENT_ PERMISSION	This order has been partially or fully processed and may not be edited by a user without permission to approve sales orders.
INSUFFICIENT_ PERMISSION	User permission level could not be established
INSUFFICIENT_ PERMISSION	You do not have permissions to set a value for element {1} due to one of the following reasons: 1) The field is read-only; 2) An associated feature is disabled; 3) The field is available either when a record is created or updated, but not in both cases.
INSUFFICIENT_ PERMISSION	Your role does not have permission to provision accounts.
INSUFFICIENT_ PERMISSION	You are not allowed to approve your own transactions.
INSUFFICIENT_ PERMISSION	You are not authorized to change this event's organizer. Public events may only have their organizer changed by administrators, the event's organizer, or delegates with edit permission to the event's calendar. Private or busy events may only have their organizer changed by the owner.
INSUFFICIENT_ PERMISSION	You cannot update a system defined template.
INSUFFICIENT_ PERMISSION	You cannot update cases using this form.
INSUFFICIENT_ PERMISSION	You can not access this page unless you are logged in as the consolidated parent company.
INSUFFICIENT_ PERMISSION	You can only delete notes that you created.
INSUFFICIENT_ PERMISSION	You do not have access to the activity history for that record
INSUFFICIENT_ PERMISSION	You do not have access to the media item you selected.
INSUFFICIENT_ PERMISSION	You do not have access to this page

Error Code Returned	Long Description or Message
INSUFFICIENT_ PERMISSION	You do not have access to this template
INSUFFICIENT_ PERMISSION	You do not have permission to access this list.
INSUFFICIENT_ PERMISSION	You do not have permission to access this register.
INSUFFICIENT_ PERMISSION	You do not have permission to access this type of transaction.
INSUFFICIENT_ PERMISSION	You do not have permission to create this type of record. Please choose a different record type.
INSUFFICIENT_ PERMISSION	You do not have permission to email transactions.
INSUFFICIENT_ PERMISSION	You do not have permission to perform this operation.
INSUFFICIENT_ PERMISSION	You do not have permission to print {1}
INSUFFICIENT_ PERMISSION	You do not have permission to view this page.
INSUFFICIENT_ PERMISSION	You do not have privileges to approve commissions.
INSUFFICIENT_ PERMISSION	You do not have privileges to create commissions.
INSUFFICIENT_ PERMISSION	You do not have privileges to create this transaction.
INSUFFICIENT_ PERMISSION	You do not have privileges to perform that operation.
INSUFFICIENT_ PERMISSION	You do not have privileges to perform this action.
INSUFFICIENT_ PERMISSION	You do not have privileges to perform this operation
INSUFFICIENT_ PERMISSION	You do not have privileges to use this page.
INSUFFICIENT_ PERMISSION	You do not have privileges to view this account
INSUFFICIENT_ PERMISSION	You do not have privileges to view this page
INSUFFICIENT_ PERMISSION	You may not create a new Liability Adjustment or edit existing Liability Adjustments.
INSUFFICIENT_ PERMISSION	You may not delete built-in audiences.
INSUFFICIENT_ PERMISSION	You may not delete built-in categories.

Error Code Returned	Long Description or Message
INSUFFICIENT_ PERMISSION	You may not delete built-in items.
INSUFFICIENT_ PERMISSION	You may not delete built-in tabs.
INSUFFICIENT_ PERMISSION	You must have either 'Transactions -> Invoice' or 'Transactions -> Cash Sale' permission to bill sales orders.
INSUFFICIENT_ PERMISSION	You must have either 'Transactions -> Invoice' or 'Transactions -> Cash Sale' permission to fulfill sales orders.
INSUFFICIENT_ PERMISSION	You must have 'Transactions -> {1}'permission to build work orders.
INSUFFICIENT_ PERMISSION	You must have 'Transactions -> Fulfill Sales Orders' view permission to view sales order fulfillments.
INSUFFICIENT_ PERMISSION	You must have 'Transactions -> Fulfill Sales Orders' edit permission to fulfill sales orders.
INSUFFICIENT_ PERMISSION	You need employee access in order to delete this record.
INSUFFICIENT_ PERMISSION	{1} The {2} restrictions on your role deny you access to this record.
INSUFFICIENT_ PERMISSION	{1} The {2} restrictions on your role prevent you from seeing this record.
INSUFFICIENT_ PERMISSION	{1} The customer restrictions on your partner role prevent you from seeing this record.
INSUFFICIENT_ PERMISSION	{1} The restrictions on your role deny you access to this record.
INSUFFICIENT_ PERMISSION	{1} The restrictions on your role do not allow you to modify this record.
INSUFFICIENT_ PERMISSION	{1} You need {2} the '{3}' permission to access this page. Please contact your account administrator.
INSUFFICIENT_ PERMISSION	{1} You need a higher level of the '{2}' permission to access this page. Please contact your account administrator.
INSUFFICIENT_ PERMISSION	{1} You need a higher permission for custom record type {2} to access this page. Please contact your account administrator.
INTEGER_REQD_FOR_ QTY	Quantity must be an integer for numbered items.
INTL_FEDEX_ONE_ PACKG_ALLWD	International FedEx fulfillments allow only one package. If more than one package is required, please break up the shipment into multiple fulfillments of one package each.
INTL_SHIP_EXCEED_ MAX_ITEM	The maximum number of items for FedEx International shipping has been exceeded: {1}
INVALID_ABN	Invalid ABN registration number {1}.
INVALID_ACCT	Invalid login. No such account.
INVALID_ACCT	Invalid account number.

Error Code Returned	Long Description or Message
INVALID_ACCT_NUM_CSTM_FIELD	The account number custom field does not exist!! Consult billing cell.
INVALID_ACCT_PRD	You can not create an accounting period that is not a year or does not belong to a year.
INVALID_ACCT_TYP	Invalid account type [{1}].
INVALID_ACCT_TYP	There is no account of type: {1}
INVALID_ACCT_TYP	The account and its parent have different account type.
INVALID_ACCT_TYP	You cannot change an account to or from A/R or A/P
INVALID_ACTION	You have attempted an unsupported action.
INVALID_ADDRESS_OR_SHIPPER_NO	An error has occurred. Please ensure that the address information and shipper number are correct, then resubmit the form.
INVALID_ADJUSTMENT_ACCT	The account you selected in Adjustment Account is the same as the asset account for one of the items you are adjusting. Please go back and change the account. Normally, the adjustment account would be an expense account.
INVALID_AES_FTSR_EXEMPTN_NUM	The AES/FTSR Exemption Number is invalid.
INVALID_ALLOCTN_METHOD	You have attempted to allocate landed costs to a transaction using an allocation method that results in no allocation for any lines in the transaction. The allocation method you chose is {1}. To correct this problem, go back to the transaction and choose a different allocation method, or modify the items/lines on the transaction so that there will be some cost allocated to the lines.
INVALID_AMORTZN_ACCT	The destination account for the amortization schedule could not be determined.
INVALID_AMT	Amount applied greater than total payments and credits
INVALID_AMT	Foreign currency transactions that use Revenue Commitments cannot have a non-discount line with a negative amount. Please use a non-posting discount item instead.
INVALID_AMT	Amount Error (credit) Insufficient Funds (debit). Transaction amount is 0 or too long. Re-submit transaction with a valid amount.
INVALID_AMT	No Action Taken (credit) PIN entry necessary (debit). Reversal amount larger then original amount. No action to take, transaction may not qualify for best level of Interchange.
INVALID_AMT	Base and consumer amounts are inconsistent with the FX rate. Correct provided amounts.
INVALID_APP_ID	Invalid application id: {1}
INVALID_ASSIGN_STATUS_COMBO	Invalid assignee/status combination({1}/{2})
INVALID_ASSIGN_STATUS_COMBO	Invalid assignee/status combination (assignee {1}, status {2}, issue #{3}). No default owner for issue role?
INVALID_ASSIGN_STATUS_COMBO	Invalid assignee/status combination ({1},{2})
INVALID_AUTH	Invalid double authorization. The order is currently authorized for {1}, valid until {2}.

Error Code Returned	Long Description or Message
INVALID_AUTH_CODE	You have entered an invalid authorization code for this campaign email address. Please check the authorization code in the email message, and enter it again.
INVALID_AUTOAPPLY_VALUE	Ambiguous data: <autoApply> has been selected and lines have been selected in the <applyList> element.
INVALID_AVS_ADDR	AVS Address Length Error. Please correct the AVS Address and re-submit.
INVALID_AVS_ADDR	Invalid AVS address or zip data. Please correct the AVS data and re-submit.
INVALID_AVS_ZIP	AVS Zip Length Error. Please correct the AVS Zip and re-submit.
INVALID_BALANCE_RANGE	Your balance is not within the allowed range.
INVALID_BILLING_SCHDUL	The billing schedule definition is incompatible with this transaction. Please modify the current billing schedule or select a different one.
INVALID_BILLING_SCHDUL_DATE	Billing schedules may not extend beyond 500 years from today
INVALID_BILLING_SCHDUL_ENTRY	You cannot create a billing schedule with two entries on the same date. Please go back and edit the billing schedule or start date.
INVALID_BIN_NUM	Bin numbers may not contain the '{1}' character
INVALID_BOM_QTY	Inventory/Assembly quantities cannot be negative
INVALID_BOOLEAN_VALUE	Checkbox / boolean data must be either 'T' or 'F'
INVALID_BUG_NUM	Bug number specified was incorrect. ("{1}" isn't a number.)
INVALID_CAMPAIGN_CHANNEL	You cannot use this channel to setup this event
INVALID_CAMPAIGN_GROUP_SIZE	While in {1}, you can only send {2} emails per campaign event. Please modify one or more of your target groups to contain {2} members or less. All campaign emails will be sent to your {1} login email address.
INVALID_CAMPAIGN_STATUS	You cannot set the status of this campaign event back to 'In Progress' because it already has some activity.
INVALID_CARD	Missing Card Holder Account Data. Please provide valid card data and re-submit.
INVALID_CARD	Card not in authorizers database.
INVALID_CARD	The referencing transaction (i.e. reversal request) was not carried out with the same card as the original transaction. Re-submit with original card.
INVALID_CARD_ID	Invalid Card ID. Provide a valid card ID.
INVALID_CARD_NUM	Invalid Card Number. Please provide a valid card number and re-submit.
INVALID_CARD_NUM	Card No. Error. Card number has unknown Bank Identification Number (BIN) or fails check digit edit. Re-try card number again, verify with merchant it is a valid card with proper logos for card types the merchants accept.
INVALID_CARD_TYP	Card Type Not Accepted. Contact Merchant e-Solutions to add the card type.
INVALID_CARD_TYP	Merchant doesn't accept the transaction's card type.
INVALID_CASE_FORM	You cannot create cases using this form.

Error Code Returned	Long Description or Message
INVALID_CATGRY_TAX_AGENCY_REQ	A Vendor must be created in a cateogry with the Tax Agency checkbox checked.
INVALID_CC_EMAIL_ADDRESS	The email address to CC store orders to is invalid. Please go back and correct it.
INVALID_CC_NUM	Credit card numbers must contain between 13 and 20 digits.
INVALID_CC_NUM	Credit card number is not valid. Please check that all digits were entered correctly.
INVALID_CC_NUM	Credit card number must contain only digits.
INVALID_CERT	Intercompany Transfer Orders cannot be entered when the Accounting Preferences > Order Management > Use Item Cost as Transfer Cost preference is enabled.
INVALID_CERT	Invalid CA certificate
INVALID_CERT	Invalid certificate key
INVALID_CERT	Invalid certificate
INVALID_CERT	Failed to verify client certificate, send email to {1} for help.
INVALID_CERT_AUTH	The indicated CA is not the issuer of this certificate
INVALID_CHARGE_AMT	Charge amount too large. This order can't be charged for more than {1}
INVALID_CHARS_IN_EMAIL	Email address contains invalid characters.
INVALID_CHARS_IN_NAME	The From Name field cannot contain apostrophes, quotation marks, commas, or greater than or less than signs.
INVALID_CHARS_IN_NAME	You cannot use the colon ':' character in the topic name - please remove it .
INVALID_CHARS_IN_PARAM_FIELD	The Additional Parameters field can not contain any of the following characters: "? \<>|/!@#$%^*()+,.:;""". Please remove them and try again
INVALID_CHARS_IN_URL	Spaces are not allowed in the {1}url.<p>Examples of a valid {1}url are: http://www.mydomain.com/image.gif or https://one.two.org/user-name/test.jpg
INVALID_CHARS_IN_URL	The URL component you have chosen contains a space or one of the following prohibited character: &?\<>|/!@#$%^&*()+=,.:;'". Please remove them and try again
INVALID_CHECKOUT_EMAIL	The email address to email checkout errors to is invalid. Please go back and correct it.
INVALID_CITY	Merchant City Length Error. Reduce the city name length.
INVALID_COLUMN_NAME	Invalid column name in get_invtitem_col_sum_all_locs: {1} [{2}]
INVALID_COLUMN_VALUE	An attempt was made to set a column to an invalid value. Dynamic SQL being executed [{1}]
INVALID_CONTENT_TYPE	Invalid content type. You can only use application/json, application/xml or text/plain with RESTlets.
INVALID_COSTING_METHOD	SERIAL and LOT are the only costing methods that may be passed as parameters to this page.

Error Code Returned	Long Description or Message
INVALID_CRNCY_EXCH_RATE	Invalid currency conversion rate.
INVALID_CRYPT_KEY	{1} is not a valid cryptographic key written as a hexadecimal number.
INVALID_CSTM_FIELD_DATA_TYP	The customfield [{1}] reference object does not match its data type.
INVALID_CSTM_FIELD_RCRD_TYP	Invalid custom field record type
INVALID_CSTM_FIELD_REF	The specified custom field reference {1} is invalid.
INVALID_CSTM_FORM	{1} is an invalid custom form
INVALID_CSTM_RCRD_KEY	Invalid custom record key [{1}].
INVALID_CSTM_RCRD_QUERY	Invalid custom record object in query.
INVALID_CSTM_RCRD_TYPE_KEY	Invalid custom record type
INVALID_CSTM_RCRD_TYPE_KEY	Invalid custom record type key.
INVALID_CSTM_RCRD_TYPE_KEY	{1} refers to a custom list. To get the contents of this list, use the 'get' or 'getAll' operation with a RecordRef of type 'customList'
INVALID_CSTM_RCRD_TYP_KEY	Invalid custom record type key in query.
INVALID_CUR	You have entered an invalid currency symbol or internal ID: {1}.
INVALID_CURR_CODE	Failed to find a currency code for the currency symbol {1}. Verify your currency symbol is ISO-compliant and re-submit.
INVALID_CURRENCY_CODE	Failed to find currency code for the requested country code. Check country code and retry request.
INVALID_CURRENCY_CODE	Request currency code must match FX rate currency code. Retry request with a currency code that matches the FX currency code.
INVALID_CURRENCY_TYP	Currency Type Not Accepted. Contact Merchant e-Solutions to add the currency type.
INVALID_CUSTOMER_RCRD	This customer record {1} is not valid. Please create the customer first.
INVALID_DATA	Invalid data combination, can not set {1} to {2} and {3} to {4}
INVALID_DATA_FORMAT	Invalid data format. You should return a JavaScript object.
INVALID_DATA_FORMAT	Invalid data format. You should return TEXT.
INVALID_DATE	Date Error. Invalid date.
INVALID_DATE	The date < {1} > is invalid. You must specify a date after < {2} >.
INVALID_DATE_FORMAT	Date field not in your preferred date format
INVALID_DATE_RANGE	The date range you specified does not enclose all its child periods.

Error Code Returned	Long Description or Message
INVALID_DATE_RANGE	Invalid time range. The {1} "{2}" start time ({3}) must be earlier than its end time ({4}).
INVALID_DATE_RANGE	Invalid Date Range - the To Date value must be on or after the From Date value
INVALID_DATE_RANGE	The date range you specified does not fall inside that of the parent period.
INVALID_DEAL_RANGE	Invalid Deal Range - low must be less than projected and high must be greater than projected.
INVALID_DEAL_RANGE	Invalid Deal Range - low must be less than projected and high must be greater than projected.
INVALID_DELETE_REF	Either RecordRef or CustomRecordRef should be used for 'delete' operation.
INVALID_DESTINATION_ FLDR	The destination folder is the same as the current folder.
INVALID_DESTNTN_ COUNTRY	The destination Country is invalid or has not been set.
INVALID_DESTNTN_ POST_CODE	The destination Postal Code is invalid or has not been set.
INVALID_DESTNTN_ STATE	The destination State is invalid or has not been set.
INVALID_DETACH_ RECORD_TYP	Missing or Invalid RecordType for DetachFrom.
INVALID_DETACH_ RECORD_TYP	Detaching of record type {1} from {2} is not supported.
INVLAID_DISCOUNT_ MARKUP	Posting and non-posting discounts/markups are not allowed on the same transaction
INVALID_DOMAIN_KEY	The private domain key is invalid, please enter a valid private domain key.
INVALID_DOMAIN_NAME	Invalid domain name {1}, please enter a valid domain name.
INVALID_DUP_ISSUE_REF	Cannot set this issue to be a duplicate of itself or of an issue that is a duplicate of this issue.
INVALID_EMAIL	Email address is not valid.
INVALID_EMAIL	Your email or code is invalid. Please try again
INVALID_EMAIL	You have entered an invalid email address. Please try again.
INVALID_EMAIL_ADDR	Some of the email addresses you have entered are invalid: {1:list of invalid email addresses}
INVALID_EMAIL_ADDR	The email address for the web store is invalid. Please go back and correct it.
INVALID_END_DATE	You entered an end date ({1}) that is before the start date ({2})
INVALID_END_DATE	{1} [{2}] recurrence end date is invalid
INVALID_END_TIME	invalid 'end' time
INVALID_ENTITY_ INTERNALID	Attempt to insert entity with nkey -1 or 0
INVALID_ENTITY_STATUS	You entered an invalid entity status.

Error Code Returned	Long Description or Message
INVALID_EVENT_TIME	You cannot make the time that close to the start or end of the day, because it shifts the event across a day boundary.
INVALID_EXP_DATE	Invalid expiration date. Please correct the expiration date and re-submit.
INVALID_EXPNS_ACCT_SUB	The expense account associated with expense category '{1:expense category}' is not available in the subsidiary of customer '{2:customer}'.
INVALID_EXPRESSION	ERROR: Invalid Expression
INVALID_FAX_NUM	The Fax Number is invalid.
INVALID_FAX_PHONE_FORMAT	Invalid FaxPhoneNumber. The format of FaxPhoneNumber must contain area code plus seven digit number.
INVALID_FIELD_FOR_RCRD_TYP	Record type {1} does not support field {2}
INVALID_FIELD_NAME_FOR_NULL	The specified name [{1}] must exactly match an existing field name.
INVALID_FILE	Verify that you have a valid file to upload.
INVALID_FILE_ENCODING	The file encoding: {1} is not valid. Please refer to the documentation for a list of supported file encodings.
INVALID_FILE_TYP	Invalid file type. File is not a compressed/zip file.
INVALID_FILE_TYP	Invalid file type. File is not a compressed zip file.
INVALID_FILE_TYP	The media file type you uploaded was not recognized. Please try again.
INVALID_FLD	ERROR: Field Not Found
INVALID_FLD_RANGE	Value outside of valid min/max range for this field
INVALID_FLD_TYPE	Application error: NLField of type {1} is not supported.
INVALID_FLD_VALUE	You have entered an Invalid Field Value {1} for the following field: {2}
INVALID_FLDR_SIZE	Error in update_folder_size
INVALID_FORMAT_IN_PARAM_FIELD	The Additional Parameters field is not formatted correctly. Please reformat and try again
INVALID_FORMULA	Your formula contains a reference to an encrypted field. This is not allowed.
INVALID_FORMULA	Your formula could result in a divide by zero error. Please go back, correct the formula and resumbit.
INVALID_FORMULA_FIELD	Your formula has an unrecognized field in it. Please go back and correct the formula and resubmit.
INVALID_FROM_DATE	invalid 'from' date
INVALID_FROM_TIME	invalid 'from' time
INVALID_FULFILMNT_ITEM	You have an invalid item {1} in the fulfillment request.
INVALID_FX_BASE_CURRENCY	FX amount in base currency is required. Provide the base amount.
INVALID_FX_RATE	Exchange Rate must be 1 for vendors in your currency.
INVALID_GET_REF	Either RecordRef or CustomRecordRef should be used for 'get' operation.

Error Code Returned	Long Description or Message
INVALID_GIFT_CERT	Invalid gift certificate
INVALID_GIFT_CERT_AMT	The remaining amount on a gift certificate can not be negative
INVALID_GIFT_CERT_CODE	Gift certificate code must contain only letters and digits.
INVALID_GROUP_TYP	This type of group cannot be defined based on another group of the same type.
INVALID_GROUP_TYP	You cannot define this group type using this search.
INVALID_GROUP_TYPE	The group type {1} is invalid.
INVALID_GRP	This type of group cannot be defined based on another group.
INVALID_GST_PST_AGENCIES	The GST or PST agencies are not valid. Please review your company preferences
INVALID_ID	No order found with id {1}
INVALID_ID	Identifiers can contain only digits, alphabetic characters, or "_" with no spaces
INVALID_ID	You have provided an invalid script id or internal id: {1}
INVALID_ID	The externalId attribute is not supported for {1}
INVALID_INITIALIZE_ARG	The reference type {1} and initialize type {2} are not matched.
INVALID_INITIALIZE_ARG	InitializeRef should be used for 'initialize' operation.
INVALID_INITIALIZE_AUXREF	Invalid initialize operation argument 'auxReference'.
INVALID_INITIALIZE_REF	You can not initialize {1}: invalid reference {2}.
INVALID_INITIALIZE_REF	Can not initialize customerPayment: invalid customer reference {1}.
INVALID_INITIALIZE_REF	Can not initialize customerPayment: invalid invoice reference {1}.
INVALID_INITIALIZE_REF	You have an invalid sales order {1} or the order is already billed
INVALID_INITIALIZE_REF	You have an invalid sales order {1} or the order is already closed.
INVALID_INSURED_VALUE	The Insured Value cannot exceed the total sum of the items being shipped.
INVALID_INTERNAL_ID	The specified internal id is not allowed.
INVALID_INTERNALID	Unparseable Internal Id, did you mean to lookup this field by Name or External ID?
INVALID_INV_DATE	Invoice date on billing schedule may not be after {1}
INVALID_INVENTORY_NUM	Invalid set of inventory numbers: values must be separated by commas, spaces, tabs, or line feeds.
INVALID_INV_DATE	Invoice date on billing schedule may not be after {1}
INVALID_IP_ADDRESS_RULE	The following IP Address rule is not valid: {1}
INVALID_ISSUE_BUILD_VERSION	Cannot set issue {1} to {2} {3} and {4} {5} because that version is not associated with that build.
INVALID_ISSUE_PRIORITY	Severity 1 issues must have priority 1.
INVALID_ISSUE_PRODUCT	Cannot set issue {1} to {2} {3} and {4} {5} because that product is not associated with that build.

Error Code Returned	Long Description or Message
INVALID_ISSUE_ PRODUCT	Cannot set issue {1} to {2} {3} and {4} {5} because that product is not associated with that module.
INVALID_ISSUE_STATUS	Cannot set issue {1} to status {2} and assignee {3} because that status requires an assignee with issue role {4}.
INVALID_ITEM_OPTION	Invalid item option {1} for item {2}
INVALID_ITEM_OPTIONS	The options for item '{1}' are no longer available. Please change your order and try again.
INVALID_ITEM_SUBTYP	Invalid item subtype [{1}] for item [{2}].
INVALID_ITEM_TYP	The item [{1}] does not have a valid item type.
INVALID_ITEM_WEIGHT	The total item weight must be > 0.0
INVALID_JOB_ID	You have specified an invalid Job Id
INVALID_KEY_OR_REF	The specified key is invalid.
INVALID_KEY_OR_REF	Invalid {1} reference key {2}.
INVALID_KEY_OR_REF	Invalid {1} reference key {2} for {3} {4}.
INVALID_KEY_ PASSWORD_REQD	This key is invalid or may require a password
INVALID_LINE_ID	No line items match the entered id(s) {1}.
INVALID_LINK_SUM	Links sum to more than applied transaction amount
INVALID_LINK_SUM	Links sum to more than original transaction amount
INVALID_LIST_ID	You must specify a valid line ID. Please set {1}.
INVALID_LIST_KEY	Could not perform operation "{1}" on an invalid line [{2}].
INVALID_LIST_KEY	Could not perform operation 'add' on an existing line [{1}].
INVALID_LOC	Item Fulfillment/Item Receipt location does not match the location on the Transfer Order
INVALID_LOC_SUB_ RESTRICTN	You may not add inventory to a location that is incompatible with the subsidiary restrictions for this item.
INVALID_LOGIN	Invalid login. Online Form access is disabled.
INVALID_LOGIN	Invalid login. Supplier access is disabled.
INVALID_LOGIN_ ATTEMPT	Invalid login attempt.
INVALID_LOGIN_ CREDENTIALS	A problem occured verifying the presented email address, password, roleName or account number, please verify these pieces of information and try again
INVALID_LOGIN_ CREDENTIALS	You have entered an invalid email address or account number. Please try again.
INVALID_LOGIN_ CREDENTIALS	You have entered an invalid email address or password. Please try again.
INVALID_LOGIN_ CREDENTIALS	You have entered an invalid login password. Please try again.

Error Code Returned	Long Description or Message
INVALID_LOGIN_ CREDENTIALS	You have entered an invalid password. Please try again.
INVALID_LOGIN_IP	Invalid login. IP Address does not match any of the IP Address rules specified for this entity.
INVALID_LOT_NUM_ FORMAT	Lot numbers must be entered using this format: LOT#(Quantity).nFor example, to enter a quantity of 100 items as Lot number ABC1234, enter "ABC1234(100)" in the Lot Numbers field.
INVALID_MARKUP_ DISCOUNT	Markup/Discount % must be between -999% and 999%
INVALID_MCC	Merchant Category Code Length Error. Provide a valid MCC.
INVALID_MEMBER_ HIERARCHY	You have defined a group/kit/assembly item that contains a loop in the member hierarchy. You must remove any group/kit/assembly member items that contain this item as a member.
INVALID_MEMRZD_ TRANS	A memorized transaction may not contain any serial or lot numbers. Go back, remove the numbers, and try to re-Memorize the transaction. Posting transactions such as Bills or Cash Sales may not use serial or lot numbered items. Non-Posting transactions such as Purchase Orders or Sales Orders may use serial or lot numbered items but may not contain serial or lot numbers.
INVALID_MERCHANT_ KEY	Merchant key is not supplied or incorrect.
INVALID_MERCHANT_ NAME	Merchant Name Length Error. Reduce the merchant name length.
INVALID_NAME	Invalid savepoint name. Must start with an alphabet character and can only contain alphanumeric, underscore, dollar, and hash characters.
INVALID_NEXUS	Transaction Nexus is incorrect: it is {1} but should be {2}
INVALID_NUM	Invalid Decimal Number
INVALID_NUMBER	Invalid Decimal Number
INVALID_NUMBER	Invalid Integer
INVALID_NUMBER	Invalid integer {1}
INVALID_NUMBER	Invalid number {1}
INVALID_NUMBER	You entered "{1}" into a field where a numeric value was expected. Please go back and change this value to a number.
INVALID_NUMBER	You entered an invalid number: Go back, change this value and resubmit.
INVALID_OBJ	There are no objects of this type
INVALID_ONLINE_FORM	Online Form not found
INVALID_ONLINE_FORM	This online form is inactive or not available online.
INVALID_ONLINE_FORM_ URL	You cannot submit Online forms from this URL. Use the live version: {1} instead
INVALID_OPENID_ DOMAIN	This is not a valid domain. Please go back and enter your domain name without prefixes such as 'http://' or 'www'.

Error Code Returned	Long Description or Message
INVALID_OPERATION	That operation is not supported for this record type: {1}
INVALID_ORD_STATUS	This order has been partially or fully processed and may not be reset to 'Pending Approval'.
INVALID_ORIGIN_COUNTRY	The origin Country is invalid or has not been set.
INVALID_ORIGIN_POSTCODE	The origin Postal Code is invalid or has not been set.
INVALID_ORIGIN_STATE	The origin State is invalid or has not been set.
INVALID_PAGE_INDEX	Job {1} does not have a page {2}
INVALID_PAGE_PARAM	Invalid page parameter. Unable to view page.
INVALID_PAGER_NUM	The Pager Number is invalid.
INVALID_PARAM	Please select either {1} or {2} parameter but not both.
INVALID_PARENT	An account cannot be its own parent
INVALID_PARTNER_CODE	An account for this customer cannot be provisioned unless its partnercode ({1}) is empty or numeric.
INVALID_PARTNER_ID	Invalid partner id: {1}
INVALID_PASSWORD	Invalid key password
INVALID_PAYCHECK_DATE	Paychecks for {1} must be on or after {2}
INVALID_PERIOD	A period may be only an adjustment period, a quarter, or a year.
INVALID_PHONE	The phone number of the {1} address is invalid. Please verify the phone number is correctly formatted and includes the area code.
INVALID_PHONE_FAX_PAGER_NUM	The Phone, Fax, or Pager Number is invalid.
INVALID_PHONE_NUM	The Phone Number is invalid.
INVALID_PICKUP_POSTAL_CODE	An error has occurred. Pickup Postal Code {1} is not the postal code associated with Shipper Number {2}.
INVALID_PIN	Incorrect PIN. PIN number may have been entered incorrectly. Re-submit with proper PIN.
INVALID_PIN_DEBIT_TRANS_TYP	Invalid pin debit transaction type. Only a sale (D) is supported.
INVALID_PORTLET_TYP	unsupported portlet type [{1}], id [{2}] processed by cardMetaDataGenerator
INVALID_POST	Invalid Post
INVALID_PRESENTATION_TYP	Presentation Type not recognized
INVALID_PROBABILITY_RANGE	Probability must be between 0 and 100.
INVALID_PROFILE_ID	Invalid Profile ID or Profile Key. Correct the profile ID and profile key, and re-submit.

Error Code Returned	Long Description or Message
INVALID_PROJ_BILLING_TYP	The project billing type is incompatible with the billing schedule on the transaction. Please select a different billing schedule.
INVALID_PST_TAX_VALUE	PST tax value is not a valid number: {1}
INVALID_PSWD	Email address "{1}" has been previously registered under a different password from the new password you just provided. For security reasons, you will first need to go back and supply the correct new password for "{1}" in order to merge the accounts.
INVALID_PSWD	Invalid Password. The password must be between 6 and 10 character with at least one numeric and one alphabetic character.
INVALID_PSWD	Password must be at least 6 characters long.
INVALID_PSWD	Password must be at least 6 characters long and contain at least one number or special character.
INVALID_PSWD	Password must contain at least one letter (A-Z).
INVALID_PSWD	Password must contain at least one number or special character.
INVALID_PSWD	The current password you supplied is incorrect.
INVALID_PSWD	Your new password must be at least {1} characters, contain at least one non-letter, and be substantially different from the current password.
INVALID_PSWD	Your new password must be at least 6 characters, contain at least one non-letter, and be substantially different from the current password.
INVALID_PSWD	Your password cannot be the same as your login. Please choose a new password.
INVALID_PSWD	Your password must be at least 6 characters
INVALID_PSWD	You've used that password before. Please choose a new password.
INVALID_PSWD_HINT	Your hint is too similar to your password. Please choose something less obvious.
INVALID_PSWD_ILLEGAL_CHAR	Password contains an illegal character.
INVALID_PURCHASE_TAX_CODE	Purchase tax code not defined properly for item
INVALID_QTY	The new quantity and new value must be either both positive or both negative.
INVALID_QTY	You may not receive a larger quantity than you shipped.
INVALID_QUANTITY	Serial and lot number quantities must be integers
INVALID_QUANTITY	Serial and lot number quantities must be positive.
INVALID_QUESTION	Please select a different question.
INVALID_QUESTION	Please select a question.
INVALID_RCRD	Invalid record specification: {1}
INVALID_RCRD_CONVERSION	Only customer records can be converted to child or parent records. Please select only customer records for this duplicate merge operation.
INVALID_RCRD_HEADER_	Invalid record header: Unable to parse field name from {1}
INVALID_RCRD_HEADER_	Invalid record header: Unable to parse record id from {1}
INVALID_RCRD_HEADER_	Invalid record header: Unable to parse record type from {1}
INVALID_RCRD_ID	Invalid id {1} to create a record.

Error Code Returned	Long Description or Message
INVALID_RCRD_ INITIALIZE	You have entered an invalid default value for this record initialize operation.
INVALID_RCRD_OBJ	You do not have a valid record object.
INVALID_RCRD_REF	Invalid RecordRef internalId {1} for field {2}
INVALID_RCRD_REF	Invalid record reference.
INVALID_RCRD_REF	Invalid record reference
INVALID_RCRD_ TRANSFRM	You have entered an invalid default value for this record transformation operation.
INVALID_RCRD_ TRANSFRM	That type of record transformation is not allowed. Please see the documentation for a list of supported transformation types
INVALID_RCRD_ TRANSFRM	That is not a valid record transformation.
INVALID_RCRD_TYPE	Invalid Record Type
INVALID_RCRD_TYPE	{1}: type argument {2} is not a valid record or is not available in your account. Please see the documentation for a list of supported record types.
INVALID_RCRD_TYPE	The record type [{1}] is invalid.
INVALID_RCRD_TYPE	The record type is invalid.
INVALID_RECIPIENT	Recipient internal id does not match an existing entity.
INVALID_RECR_REF	Could not update {1} to {2} because referenced record does not exist
INVALID_RECUR_DATE_ RANGE	This event recurrence is invalid because its duration is either negative or longer than one day. {1}
INVALID_RECUR_DATE_ RANGE	This event recurrence is invalid because its end-by date is before its start date. {1}
INVALID_RECUR_DATE_ RANGE	This event recurrence is invalid because its end time and duration do not match. {1}
INVALID_RECUR_DATE_ RANGE	This event recurrence is invalid because its end time is more than one day after its start time. {1}
INVALID_RECUR_DATE_ RANGE	This event recurrence is invalid because its end time is not after its start time. {1}
INVALID_RECUR_DATE_ RANGE	This event recurrence is invalid because its start time or end time/duration is empty. {1}
INVALID_RECUR_DATE_ RANGE	This event recurrence is invalid because its times are not in order. {1}
INVALID_RECUR_DESC_ REQD	This event recurrence is invalid because it has no description. {1}
INVALID_RECUR_DOW	This event recurrence has an invalid day-of-week field. {1}
INVALID_RECUR_DOWIM	This event recurrence has an invalid day-of-week-in-month value. {1}
INVALID_RECUR_ DOWMASK	This event recurrence is invalid because its day-of-week mask is not 7 characters long. {1}

Error Code Returned	Long Description or Message
INVALID_RECUR_FREQUENCY	This event recurrence has an invalid frequency. {1}
INVALID_RECUR_PATTERN	This event does not have a valid recurrence pattern.
INVALID_RECUR_PATTERN	This event recurrence is invalid because it is not a monthly or yearly event and it has day-of-week and day-of-week-in-month field values. {1}
INVALID_RECUR_PATTERN	This event recurrence is invalid because it only has one recurrence time and it must have either none or at least two. {1}
INVALID_RECUR_PATTERN	This event recurrence is invalid because one of its times is out of the range 0 to 86399. {1}
INVALID_RECUR_PATTERN	This event recurrence is invalid because only one of the day-of-week and day-of-week-in-month fields is set. Both must be set or both must be unset. {1}
INVALID_RECUR_PATTERN	This event recurrence is invalid either because it is not weekly and it has a day-of-week mask, or it is weekly and it has no day-of-week mask. {1}
INVALID_RECUR_PATTERN	This single day event is invalid since it contains a recurrence pattern.
INVALID_RECUR_PERIOD	This event recurrence has an invalid period. {1}
INVALID_RECUR_TIME_ZONE_REQD	This event recurrence in invalid because it has no time zone. {1}
INVALID_REF_CANT_INITIALIZE	Cannot initialize customerRefund: invalid creditMemo reference {1}.
INVALID_REF_CANT_INITIALIZE	Cannot initialize customerRefund: invalid customer reference {1}.
INVALID_REF_CANT_INITIALIZE	You can not initialize {1} by referencing {2}.
INVALID_REF_KEY	Invalid externalId {1}.
INVALID_REF_KEY	Invalid reference key [{1}].
INVALID_REFFERER_EMAIL	The refferer email address you have entered is not valid. Please try again.
INVALID_REFUND_AMT	Refund amount must be between zero and the original amount. Correct amount and retry request.
INVALID_REFUND_AMT	The amount you can refund is {1} because the the order has already been refunded for {2}
INVALID_REPORT	The referenced Report and Row are no longer valid, because the layout containing them has changed. Please edit this reference row to reselect Report and Row.
INVALID_REPORT_ID	Invalid report ID.
INVALID_REPORT_ROW	Invalid Reference Row
INVALID_REPORT_ROW	Invalid Report Row Reference
INVALID_REQUEST	invalid request (failed isValid() check). Email request handler unable to service request for address= <{1}> .

Error Code Returned	Long Description or Message
INVALID_RESOURCE_TIME	Total resource time for '{1}' cannot exceed {2} planned time entries.
INVALID_RESULT_SUMMARY_FUNC	The result field {1} cannot be grouped. Please edit the search and omit this field or use a different summary function.
INVALID_REV_REC_DATE_RANGE	Rev rec end date can not be before rev rec start date.
INVALID_ROLE	The specified role is invalid.
INVALID_ROLE	Your role does not give you permission to view this page.
INVALID_ROLE_FOR_EVENT	You seem to have been invited to this {1} in a different role. Please change your role to view the {1}.
INVALID_RQST_CONTACTS_EXIST	it has associated primary contacts.
INVALID_RQST_PARENT_REQD	it has associated contact records that would be left with no parent company.
INVALID_RQST_SBCUST_JOBS_EXIST	it has associated sub-customers or jobs.
INVALID_SAVED_SRCH	Missing or invalid saved search for Custom KPI. The search must have a date column as an available filter. Please see help next to Custom KPI drop down on KPI setup page.
INVALID_SAVEDSEARCH	A saved search with the internal ID {1} does not exist.
INVALID_SAVEDSEARCH	We cannot return search columns for summary saved search {1}.
INVALID_SCHDUL_AMT	The total amount on the schedule must equal the sum of the individual recognition amounts.
INVALID_SCHDUL_AMT	The total amount on the schedule must be equal to the amount of the source transaction line.
INVALID_SCHDUL_FORMAT	To create a valid schedule, please enter the bracket values in ascending orders without gaps.
INVALID_SCRIPT_ID	A saved search with the script ID {1} does not exist.
INVALID_SEARCH	That search or mass update does not exist.
INVALID_SEARCH	You may search by {1} or {2} but not both
INVALID_SEARCH_CRITERIA	Can't search transactions: invalid cross reference key
INVALID_SEARCH_CRITERIA	Global Search supports at most three keywords and requires at least one. Keywords are composed of only letters, digits, and dashes.
INVALID_SEARCH_FIELD_KEY	search field keys are not consistent({1}/{2})
INVALID_SEARCH_FIELD_NAME	search field names are not consistent({1}/{2})
INVALID_SEARCH_FIELD_OBJ	{1} is not a valid search custom field

Error Code Returned	Long Description or Message
INVALID_SEARCH_FIELD_OBJ	{1} must be used to search custom field {2}
INVALID_SEARCH_FIELD_OBJ	Server application error: invalid search customfield object.
INVALID_SEARCH_FIELD_OBJ	Invalid search field object: {1}
INVALID_SEARCH_JOIN_ID	Invalid Search Join ID
INVALID_SEARCH_MORE	Invalid searchMore operation. Please make sure that you have had a successful search operation before you can perform any searchMore operation.
INVALID_SEARCH_OPERATOR	You need to provide a valid search field operator.
INVALID_SEARCH_OPERATOR	You can not use this operator '{1}' for internalId search.
INVALID_SEARCH_PAGE_INDEX	Invalid search page index.
INVALID_SEARCH_PAGE_SIZE	Invalid search page size.
INVALID_SEARCH_PREF	You cannot set returnSearchColumns to false while you specify search columns.
INVALID_SEARCH_PREF	You cannot set returnSearchColumns to true without specifying search columns or referencing a saved search.
INVALID_SEARCH_SELECT_OBJ	Invalid search select field object: {1}
INVALID_SEARCH_VALUE	You need to provide a search value.
INVALID_SEARCH_VALUE	You need to provide search values.
INVALID_SECONDARY_EMAIL	Invalid secondary email address. The email address must be in a valid format.
INVALID_SECPAY_CREDENTIALS	The username or password used to process the transaction with SECPay was not valid. Please make sure you have entered the correct username, password, and remote password in your SECPay account setup.
INVALID_SERIAL_NUM	No items match the entered serial number
INVALID_SERIAL_OR_LOT_NUMBER	Serial and lot numbers may not contain the '{1}' character.
INVALID_SESSION	A valid NLSession is required to generate record xml
INVALID_SESSION	A valid session is required. Please log in first.
INVALID_SHIP_DATE	The Future Ship Date is invalid. Please verify the entered Future Ship Date is no more than 7 days in the future, and resubmit the fulfillment.
INVALID_SHIP_FROM_STATE	The Ship From State/Province Code is missing or invalid. Please enter the 2 to 5 character abbreviation for the state or province of the address that contains it.

Error Code Returned	Long Description or Message
INVALID_SHIP_GRP	You cannot add shipping groups when creating a transaction that has multiple shipping routes enabled. You must first add the items, then get the transaction and update the shipping groups separately.
INVALID_SHIP_SRVC	The selected service is not valid for international shipments. Please choose an international service and retry your request.
INVALID_SHIP_TO_SATE	The Ship To State/Province Code is missing or invalid. Please enter the 2 to 5 character abbreviation for the state or province of the address that contains it.
INVALID_SHIPPER_STATE	The Shipper State/Province Code is missing or invalid. Please enter the 2 to 5 character abbreviation for the state or province of the address that contains it.
INVALID_SITE_CSTM_FILE	File is not a NetSuite site customization export file: it cannot be imported.
INVALID_SOAP_HEADER	Invalid SOAP Header: '{1}'. Value is '{2}'.
INVALID_SRCH	That search or mass update does not exist (internal id={1})
INVALID_SRCH_CRITERIA	The field rule value "{1}" is invalid for field type {2} with criterion "{3}."
INVALID_SRCH_CSTM_ FLD	This search refers to custom field with id = {1} which either is restricted or is not applied to this record type.
INVALID_SRCH_FUNCTN	An nlobjSearchColumn contains an invalid function: {1}.
INVALID_SRCH_SORT	An nlobjSearchColumn that is not sortable contains a sort specification: {1}.
INVALID_SRCH_ SUMMARY_TYP	An nlobjSearchFilter contains an invalid summary type: {1}.
INVALID_SRCH_TYP	Search Type not allowed as standalone search
INVALID_SRVC_ITEM_ SUB	The service item '{1:service item}' is not available in the subsidiary of customer '{2:customer}'.
INVALID_SSO	Invalid SuiteSignOn reference: {1}. That SuiteSignOn object does not exist or has been marked as inactive.
INVALID_SSS_DEBUG_ SESSION	You have exceeded the maximum allowable idle time for debugging scripts. To debug another script, simply reload the script debugger page and start a new debugging session.
INVALID_SSS_DEBUG_ SESSION	You have cancelled your current script debugging session.
INVALID_STATE	Merchant State Length Error. Provide a valid state.
INVALID_STATE	Signup prospect state '{1}' is invalid.
INVALID_STATUS	You may not change this issue''s status from "{1}" to "{2}".
INVALID_SUB	The subsidiary restrictions on this record are incompatible with those defined for account: {1}. Subsidiary access on this record must be a subset of those permitted by the account.
INVALID_SUB	The subsidiary restrictions on this record are incompatible with those defined for account: {1}. Subsidiary access on this record must be a superset of those permitted by the account.
INVALID_SUB	The subsidiary restrictions on this record are incompatible with those defined for department: {1}. Subsidiary access on this record must be a subset of those permitted by the department.

Error Code Returned	Long Description or Message
INVALID_SUB	The subsidiary restrictions on this record are incompatible with those defined for item: {1}. Subsidiary access on this record must be a superset of those permitted by the item.
INVALID_SUB	The subsidiary restrictions on this record are incompatible with those defined for location: {1}. Subsidiary access on this record must be a subset of those permitted by the location.
INVALID_SUB	The Subsidiary selected doesnt match the bank account selected.
INVALID_SUB	This record does not support multiple subsidary restrictions. You must choose a single subsidiary.
INVALID_SUB	Transaction references multiple subsidiaries
INVALID_SUB	You may not add members to a group/kit/assembly unless the subsidiaries for those members completely contain the subsidiaries of the group/kit/assembly.
INVALID_SUB	{1} can not be used with the selected subsidiary
INVALID_SUBLIST_DESC	Invalid sublist description - all sublists must appear exactly as they do in the WSDL (eg end with 'List')
INVALID_SUBSCRIPTION_ STATUS	You cannot change the global subscription status from its current value of {1:status name}.
INVALID_SUBSCRIPTION_ STATUS	You cannot set the global subscription status to the value {1:status name}.
INVALID_SUMMARY_ SRCH	In a summary search, you must sort by a result field with a summary function. Please go back and correct the sort by field on the results tab.
INVALID_SUPERVISOR	Employees can not be their own supervisor.
INVALID_SUPERVISOR	You can't insert this employee record as it would create a loop in the supervisor hierarchy.
INVALID_TASK_ID	The task ID: {1} is not valid. Please refer to the documentation for a list of supported task IDs.
INVALID_TASK_ID	You have specified an invalid task Id
INVALID_TAX_AMT	Invalid tax amount. Correct tax amount and retry request.
INVALID_TAX_CODE	Invalid Canadian Tax Code: {1}
INVALID_TAX_CODE_ FOR_SUB	The selected tax code is not available in subsidiary.
INVALID_TAX_CODES	Invalid Tax Code(s): {1}
INVALID_TAX_PMT	You may not commit tax payment information prior to the start date of the Payroll Service.
INVALID_TAX_VALUE	GST and PST amount cannot be negative!
INVALID_TAX_VALUE	GST tax value is not a valid number: {1}
INVALID_TIME_FORMAT	{1} is not a valid time and it should use the following format h:mm a.
INVALID_TO_DATE	invalid 'to' date
INVALID_TRACKING_ NUM	The tracking number is not valid.

Error Code Returned	Long Description or Message
INVALID_TRACKING_NUM	You have entered a tracking number that exceeds the maximum size of {1} characters: {2}. Multiple tracking numbers must be separated by spaces, tabs, or commas. Slash (/), semicolon (;), colon (:), or any other character that is not a space or a comma will be interpreted as a part of the tracking number.
INVALID_TRAN_ITEM_LINE	Item {1} can not be included on the {2} because it is not distributed as of the transaction date
INVALID_TRANS	This transaction is not valid.
INVALID_TRANS_COMPNT	You have entered an invalid component for this transaction.
INVALID_TRANS_ID	Invalid Transaction ID. Correct the transaction ID, then re-submit.
INVALID_TRANS_ID	Only sale transactions can be refunded. Provide a valid transaction ID.
INVALID_TRANS_SUB_ACCT	Transaction subsidiary {1} is not valid for account {2}. Please choose a different account.
INVALID_TRANS_SUB_CLASS	Transaction subsidiary {1} is not valid for class {2}. Please choose a different class.
INVALID_TRANS_SUB_DEPT	Transaction subsidiary {1} is not valid for department {2}. Please choose a different department.
INVALID_TRANS_SUB_ENTITY	Transaction subsidiary {1} is not valid for entity {2}. Please choose a different entity.
INVALID_TRANS_SUB_ITEM	Transaction subsidiary {1} is not valid for item {2}. Please choose a different item.
INVALID_TRANS_SUB_LOC	Transaction subsidiary {1} is not valid for location {2}. Please choose a different location.
INVALID_TRANS_TYP	Transaction type specified is incorrect.
INVALID_TRANSACTIO_DATE	There are no Accounting Periods that cover this transaction date.
INVALID_TRANSACTION_DATE	Transaction date {1} is not valid. Transaction dates may be at most {2} years in the past and {3} years in the future.
INVALID_TRIAL_TYP	The trialtype is not availabe in the product specified.
INVALID_TYP	Invalid type {1}, use {2}
INVALID_UNIT_TYP	On serialized items, you may not choose a units type that has fractional conversion rates.
INVALID_UNSUPRTD_RCRD_TYP	Invalid or unsupported record type: {1}
INVALID_UPS_ACCT	An invalid UPS Account Number was entered. Please verify you have entered the correct Shipper Number and re-submit the form.
INVALID_UPS_PACKG_WEIGHT	UPS requires a minimum package weight of .1 LBS and a maximum package weight of 150 LBS. Please adjust the package weights accordingly and resubmit the fulfillment.
INVALID_UPS_VALUES	UPS did not accept the entered values for the following fields. Please go back and correct these values:

Error Code Returned	Long Description or Message
INVALID_URL	Please begin the {1} url with http:// or https://<p>Examples of a valid {1}url are: http://www.mydomain.com/image.gif or https://one.two.org/user-name/test.jpg
INVALID_URL	Request for invalid URL: {1}
INVALID_URL_PARAM	Error: URL param {1}="{2}" - expected an integer.
INVALID_VALUE	You have entered an invalid value {1} for {2}.
INVALID_VAT_AMOUNT	VAT amount cannot be negative
INVALID_VAT_REGSTRTN_NUM	Invalid VAT registration number {1}.
INVALID_VSOE_ALLOCTN	VSOE allocations must be greater than or equal to 0
INVALID_WEBSITE_SECTION	The Web site section you entered does not exist.
INVALID_WO	You have an invalid work order {1} or the order is already closed.
INVALID_WO_ITEM	Special Work Order Items can not be Drop Ship or Special Order
INVALID_WORLDPAY_ID	Exchange source does not recognize your WorldPay ID. Please check that it is correct.
INVALID_YEAR	Invalid year {1}
INVALID_YEAR_FORMAT	Illegal year format or value. Examples: 1999, 2000, 2001, etc.
INVALID_ZIP_CODE	Merchant Zip Length Error. Provide a valid 5 digit zip.
INVALID_ZIP_FILE	Invalid archive. Zip file must contain at least one file.
INVALID_ZIP_POST_CODE	The submitted Zip/Postal Code is invalid. This field may only contain a maximum of 16 digits, spaces, and the dash character (-).
INVENTORY_NUM_DISALLWD	Inventory numbers are only allowed on items with serial numbered or lot numbered items.
INVLAID_BOOLEAN_VALUE	You have entered an invalid boolean value. Please use true, false, T, or F for boolean values and resubmit your import.
ISSUE_ASSIGNEE_DISALLWD	The specified assignee is disallowed for this issue's status.
ISSUE_PRODUCT_VERSION_MISMATCH	Cannot set issue {1} to {2} {3} and {4} {5} because that product is not associated with that version.
ISSUE_VERSION_BUILD_MISMATCH	Issue version and build do not match.
ITEM_ACCT_REQD	One of the items on this transaction has an amount but no account. Please fix the item and resubmit the transaction.
ITEM_ACCT_REQD	One of the items on this transaction has an amount but no account. Please fix the item and resubmit the transaction. It might be that you have recently elected to charge for shipping and have not assigned an account to the shipping item that is included in this transaction.
ITEM_ACCT_REQD	You must specify asset and COGS accounts for this inventory item.

Error Code Returned	Long Description or Message
ITEM_COUNT_MISMATCH	COGS_CORRECTION: 2 means of calculating the item count do not match for item: {1} vs {2})
ITEM_COUNT_MISMATCH	COGS_CORRECTION: 2 means of calculating the item count do not match for item: {1} vs {2}) There are transactions in the system in which this item is used but the asset account for that item is not the current Asset Account in the item record
ITEM_IS_UNAVAILABLE	(Item is unavailable)
ITEM_NAME_MUST_BE_UNIQUE	An item with that name already exists. Please choose another name
ITEM_NOT_UNIQUE	The item [{1}] is not unique.
ITEM_PARAM_REQD_IN_URL	Error - Item parameter (id=nnn) was not provided on the URL
ITEM_TYP_REQS_UNIT	Items of type {1} require {2} unit
ITEM_TYP_REQS_UNIT	Items of type {1} subtype {2} require {3} unit
JE_AMOUNTS_MUST_BALANCE	The amounts in the journal entry must balance.
JE_LINE_MISSING_REQD_DATA	{1} are mandatory on all lines of the journal entry.
JE_REV_REC_IN_PROGRESS	This account is currently processing Revenue Recognition Journal Entries. Only one such process is allowed at a time.
JE_UNEXPECTED_ERROR	Journal Entries failed to be created due to unexpected error.
JOB_NOT_COMPLETE	The specified job is not complete yet
JS_EXCEPTION	A JavaScript Exception was thrown
KEY_REQD	Empty key not allowed for {1}
KPI_SETUP_REQD	Please enable the Forecast ({1}) & Quota KPIs
KPI_SETUP_REQD	Please enable the Forecast & Quota KPIs
KPI_SETUP_REQD	please enable the Sales & Forecast KPIs
KPI_SETUP_REQD	Please enable the Sales & Forecast ({1}) KPIs
KPI_SETUP_REQD	Please enable the Sales & Quota KPIs
LABEL_REQD	Please enter a value for Label
LANGUAGE_SETUP_REQD	Please go to company preference to add language to translate.
LINKED_ACCT_DONT_MATCH	You are attempting to link transaction line items, but items on the lines do not match. This can happen when you create a fulfillment from a sales order, a receipt from a purchase order, an invoice from a sales order, a vendor bill from a purchase order, or a reimbursement from a purchase. Please verify that items in the transaction you are creating match the items in the originating transaction.
LINKED_ITEMS_DONT_MATCH	Linked items don't match
LIST_ID_REQD	Required field missing in a related list. You must set {1}.
LIST_KEY_REQD	There is no list key for field {1} of list {2}. Please assign a key and resubmit your task.

Error Code Returned	Long Description or Message
LOCATIONS_IN_USE	Your classes cannot be converted to locations because your existing location records are referred to by transactions or other records. These location records cannot be overwritten.
LOCATIONS_SETUP_ REQD	You must first define locations (Lists->Locations->New) before you can distribute inventory.
LOCATIONS_SETUP_ REQD	You must first define locations (Lists->Locations->New) before you can transfer inventory.
LOCATION_REQD	You must specify a location in order to use {1} numbers when Multi-Location Inventory is enabled
LOCKED_DASHBOARD	Your dashboard has been set up and locked by an administrator. Please contact them for details.
LOGIN_DISABLED	Invalid login. Customer access is disabled.
LOGIN_DISABLED	Login access has been disabled for this role.
LOGIN_DISABLED	Your access to {1} has been deactivated. Please contact the company's administrator to re-activate your access.
LOGIN_DISABLED	Your access to this account has been removed or disabled. Please contact the account adminstrator.
LOGIN_DISABLED_ PARTNER_CTR	Disabled login: Advanced Partner Center access has been disabled by the account administrator.
LOGIN_DISABLED_ PARTNER_CTR	Disabled login: Standard Partner Center access has been disabled by the account administrator.
LOGIN_EMAIL_REQD	Invalid login. You must provide an email address.
LOGIN_NAME_AND_ PSWD_REQD	Please enter both a user name and a password.
LOGIN_REQD	You must log in before accessing this page.
LOST_UPSELL_CRITERIA	Your upsell criteria were lost. This is probably due to a transient condition such as a server reboot. Click here to go back and try again.
MACHN_LIST_KEY_ NAMES_REQD	Server application error: no list key names are defined for field {1} of record of type {2}.
MANDATORY_PRD_ TYPE_REQD	Please select the mandatory period type...
MASS_UPDATE_ CRITERIA_LOST	Your mass update criteria were lost. This is probably due to a transient condition such as a server reboot. Click here to go back and try again.
MATCHING_CUR_SUB_ REQD	The parent specified must have the same currency and subsidiary as the child
MATCHING_SERIAL_ NUM_REQD	The serial numbers on a transfer order receipt must have been fulfilled
MATRIX_INFO_TEMP_ LOST	Matrix item information was lost. This was probably due to a transient condition like a server reboot. Please try again.

Error Code Returned	Long Description or Message
MATRIX_SUBITEM_NAME_TOO_LONG	The following matrix sub-item name is too long (80 character max):<p> {1} <p> Please shorten your parent item name or your option abbreviations.
MAX_16_LINES_ALLWD_PER_BILLPAY	A maximum of 16 lines per payee can be applied per online bill payment.
MAX_200_LINES_ALLWD_ON_TRANS	Journal Entries can have a maximum of 200 lines.
MAX_BARCODE_PRINT_EXCEEDED	A maximum of {1} barcodes can be printed at a time.
MAX_BULK_MERGE_RCRDS_EXCEEDED	You cannot perform a bulk merge operation with a group larger than {1} records
MAX_EMAILS_EXCEEDED	This campaign email event exceeds the number of emails ({1}) that can be sent per month without setting up a default campaign domain or specifying one on the campaign email template.
MAX_EMAILS_EXCEEDED	This merge operation exceeds the number of emails ({1}) that can be sent per month without setting up a bulk merge domain or specifying one on the email template.
MAX_RCRDS_EXCEEDED	The maximum number ({1}) of records allowed for a {2} operation has been exceeded.
MEDIA_FILE_INVALID_JSCRIPT	Media file was of type javascript and would not compile. Error on line:
MEDIA_NOT_FOUND	Media item not found {1}
MEDIA_NOT_INITIALIZED	Media Item cannot be initialized
MEMORIZED_TRANS_ERROR	failed retrieving 'chargeit' record from trancard while processing memorized tran
MEMORIZED_TRANS_ERROR	A failure occurred while trying to enter one of your automated memorized transactions. The reason for this failure is shown in the description field on this page. Because of the failure, the system has changed this memorized transaction from an automated one to a reminder. You may click the link above to view the memorized transaction in question and make any needed changes to it. When you are done with the changes, click Memorize and then Cancel. This will modify your existing memorized transaction rather than creating a new one. When filling out the Memorized Transaction Form, you may choose to select 'Automatic' to resume automated posting of this transaction.
MERGE_OPERATION_DISALLWD	You cannot perform merge operations on records that belong to your group.
MERGE_RCRD_REQD	You must specify a record to merge into
METAVANTE_ERROR	This Online Bill Payment has been {1} in your NetSuite account. However, the payment was not able to be {2} at Metavante. In order to {3} this payment at Metavante, please contact NetSuite Customer Support immediately, using the options provided in the Support > Customer Service section of your NetSuite account.
METAVANTE_SECRET_ANSWER_REQD	Missing Secret Answer. A secret answer is required by the Metavante CSP service. It cannot be null or empty.

Error Code Returned	Long Description or Message
METAVANTE_SECRET_QESTION_REQD	Missing or invalid Secret Question ID. A valid Secret Question is required by the Metavante CSP service. Please refer to Table 2, above, for a list of valid Secret Question IDs.
METAVANTE_SETUP_REQD	Your NetSuite account is not currently integrated with a Metavante Online Bill Pay account. To set up an active account, you need to reapply to Metavante. Go to Setup > Set Up Online Bill Pay and follow the instructions on that page to apply for a Metavante account.
METAVANTE_TEMP_UNAVAILBL	Metavante is temporarily unavailable. Please try again later. If you would like to print the payment to mail yourself, click Back, and then click the date of the payment on the Approve Online Bill Payments page. When the payment's detail record appears, clear the Bill Pay box and either check the To Be Printed box and click Submit or click the Print button.
MISMATCH_EVENT_ISSUE_STATUS	Event status ({1}) and issue base status ({2}) do not match
MISMATCH_ISSUE_PRODUCT_VERSION	Issue product and version do not match.
MISMATCH_SALES_CONTRIBUTION	Sales team sales rep total does not equal 100%, {1} sales reps, {2} total contribution.
MISMATCHED_CURRENCY	The transaction currency does not match the names currency
MISMATCHED_QTY_PRICING	Quantities do not match accross pricings
MISMATCHED_SEARCH_PARENTHESIS	Search error: Parentheses are unbalanced.
MISSING_ACCT_PRD	You are attempting to create an amortization or revenue recognition schedule outside the range of available accounting periods. Please adjust the periods on this transaction or go to Setup>Accounting>Manage Accounting Periods to set up more periods.
MISSING_ENUM	No Enumerated Value {1} for Enumerated Type {2}
MISSING_REQD_FLD	Missing required value for mandatory field: {1}
MISSING_REQD_FLD	The Company record does not have all required fields set. Please ensure the State, Zip/Postal Code, and Country fields are set and try your request again.
MISSNG_ACCT_PRD	Unable to find an Accounting Period for the allocation date.
MISSNG_REV_REC_RCRD	Unable to locate Revenue Recognition records.
MISSNG_SO_REV_REC_PARAMS	Unable to get Revenue Recognition parameters from originating sales order.
MISSNG_SO_START_END_DATES	Unable to acquire start and end date from Sales Order.
MLI_REQD	Multi-location Inventory Error (MLI_LOCATION_REQUIRED): this transaction or its items must have locations.
MLTPLE_TAX_LINES_DISALLWD	Multiple Tax lines for line item in transaction:

Error Code Returned	Long Description or Message
MSNG_FIELD_OWRTE_MUST_BE_TRUE	The missingFieldOverwrite attribute must be true when updating a salesOrder.
MST_UPDATE_ITEMS_THEN_RATES	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.
MULTI_ACCT_CANT_CHANGE_PSWD	The password cannot be changed here because the email address is associated with multiple accounts. The user must change their password via the link in the settings portal of the home page.
MULTI_LOC_INVT_ERROR	Multi-Location Inventory Error: You may not create an Assembly Build transaction with an assembly item that has not been distributed and member items that have been distributed. You must create an Inventory Distribution transaction for the assembly item before building it. You also may not create an Assembly Build transaction on a date prior to the distribution date of the assembly but after the distribution date of any of the member items.
MULTI_PRIMARY_PARTNER_DISALLWD	You are not allowed to select multiple primary partners.
MULTI_SHIP_ROUTES_REQD	{1} {2} has multiple shipping routes enabled, which is only supported in version 2008_2 and newer. You are not allowed to update any shipping fields on this record.
MULTISELECT_TYPE_REQD	Server application error: no multiselect type is defined for field {1} of {2} record type.
MUST_RESUBMIT_RCRD	Configuration changes have been made to your account. You must resubmit your record.
NAME_ALREADY_IN_USE	A mass update has already been saved with that name. Please use a different name.
NAME_ALREADY_IN_USE	A search has already been saved with that name. Please use a different name.
NAME_REQD	Missing Name. Name is a required field and it cannot be null or empty.
NAME_TYPE_FLDR_FIELDS_REQD	missing required fields : name, type, and folder
NARROW_KEYWORD_SEARCH	Please provide more detailed keywords so your search does not return too many results.
NEGATIVE_PAYMENT_DISALLWD	Negative payments not allowed
NEGATIVE_TAX_RATE_DISALLWD	A Tax rate cannot be negative
NEW_CONNECTION_DISALLWD	Not allowed to create new connections.
NEXUS_REQD	No tax agency defined for subsidiary: subsidiary {1} is not linked to nexus {2}
NO_DATA_FOUND	No data was found
NO_EXPENSES_FOR_PRD	The Allocation sources or destinations did not have any expenses associated with them for the selected period.
NO_ITEMS_TO_PRINT	There are no items to print

Error Code Returned	Long Description or Message
NO_MASS_UPDATES_RUNNING	There are currently no mass updates running.
NO_MTRX_ITEMS_TO_UPDATE	There are no matrix subitems to update.
NO_ORD_SHPMNT	There is no shipment on that order.
NO_RCRD_FOR_USER	There is no record for this user in the company's entity table. (emaillogin.semail='{1}', kentity={2})
NO_RCRDS_MATCH	No Records matched your request.
NO_SCHDUL_APPLIED	There were no schedules that need to applied to the given period.
NO_SCHDUL_APPLIED	There were no schedules that need to be applied to the input accounting period.
NON_ADMIN_CANNOT_INITIATE_LINK	This user cannot integrate with a partner.
NONMATCHING_EMAILS	Email addresses don't match
NONZERO_AMT_REQD	You did not enter non-zero amounts for any accounts.
NONZERO_QTY_REQD	Build quantity must be greater than zero.
NONZERO_WEIGHT_REQD	Selected service must have a weight greater than zero.
NOT_AN_INVITEE	You are not on the invitee list for event.
NOT_IN_INVT	You may not distribute {1} numbers that are not currently in inventory. You attempted to distribute the following {1} numbers that were not in inventory: {2}
NULL_CHECK_NUMBER	Null Check Number
NUM_ITEMS_GRTR_THAN_QTY	The number of {1} entered ({2}) is greater than the item quantity ({3})
NUM_ITEMS_NOT_EQUAL_TO_QTY	The number of {1} entered ({2}) is not equal to the item quantity ({3})
NUM_REQD_FOR_FIRST_LABEL	No number was specified for the first label.
NUMERIC_CHECK_NUM_REQD	Invalid Check Number. Check number must be a numeric value and can be at most 7 digits long.
NUMERIC_REF_NUM_REQD	Reference Number Must Be Numeric. Please provide a valid number and re-submit.
OI_FEATURE_REQD	You have not enabled Outlook Integration feature for your account.
OI_PERMISSION_REQD	You do not have permission to access Outlook Integration feature.
ONE_ADMIN_REQD_PER_ACCT	This operation would leave your account without an active Administrator. In order to successfully perform the mass update, please deselect at least one entity with an Administrator role.
ONE_ADMIN_REQD_PER_ACCT	You can't delete this employee. No administrators for this account would remain.
ONE_ADMIN_REQD_PER_ACCT	You can't inactivate {1}. The account would be left with no active administrators.

Error Code Returned	Long Description or Message
ONE_ADMIN_REQD_PER_ACCT	You can't remove the administrator role from this user. No administrators for this account would remain.
ONE_EMPL_REQD	At least one employee is required to process payroll
ONE_POSITIVE_VALUE_REQD	You must enter at least one positive value for at least one item.
ONE_RCRD_REQD_FOR_MASS_UPDATE	Please create at least one {1} before using this mass update.
ONE_ROLE_REQD	You can't inactivate all roles. You would not be able to log in.
ONLINE_BANK_FILE_REQD	You must first upload an Online Bank file before using the Online Bank Statement.
ONLINE_BILL_PAY_SETUP_REQD	{1} is not set up for Online Bill Pay. To set up this payee, click Go Back. When the Approve Online Bill Payments page appears, click Enable Payee in the Enabled for Billpay column. When the payee's record appears, check Enable Online Bill Pay and submit these required fields: - Legal Name - Print on Check As - Phone - Billing Address Then, go to Transactions > Approve Online Bill Payments to approve the payment.
ONLINE_FORM_DSNT_EXIST	This online form does not exist.
ONLINE_FORM_EMPTY	The online form you requested is empty.
ONLINE_FORM_ID_REQD	Missing required online form ID
ONLINE_FORM_USER_ACCESS_ONLY	This form is only accesible to online form users.
ONLINE_ORD_FEATURE_DISABLED	Can't open store for {1}. This company does not have the Use Sales Orders feature enabled. The feature is required for customers to make online purchases.
ONLY_ONE_CONTRIB_ITEM_REQD	Only one instance of a company contribution item is allowed on an employee record.
ONLY_ONE_DEDCT_ITEM_REQD	Only one instance of a deduction item is allowed on an employee record.
ONLY_ONE_DISTRIB_ALLWD	You may not distribute {1} numbers more than once. You attempted to distribute the following {1} numbers more than once: {2}
ONLY_ONE_EARNING_ITEM_REQD	Only one instance of an earning item is allowed on an employee record.
ONLY_ONE_LOT_NUM_ALLWD	You may not enter more than a single serial/lot number before an item is selected.
ONLY_ONE_PREF_BIN_ALLWD	There may be at most one preferred bin per location for an item. The following location has more than one preferred bin for this item: {1}
ONLY_ONE_PREF_BIN_ALLWD	You may not have more than one preferred bin per item.
ONLY_ONE_UNIT_AS_BASE_UNIT	Only one unit may be designated as the base unit.
ONLY_ONE_UPLOAD_ALLWD	You cannot upload more than one file at a time

Error Code Returned	Long Description or Message
ONLY_ONE_WITHLD_ ITEM_REQD	Only one instance of a withholding item is allowed on an employee record.
ORD_ALREADY_APPRVD	You cannot cancel this order because it has already been approved.
ORDER_DSNT_EXIST	That order does not exist.
OVER_FULFILL_ DISALLWD	You can not over-fulfill an item unless you have selected the 'Allow Overage on Item Fulfillments' preference.
OVER_FULFILL_RECEIV_ DISALLWD	Transfer orders can not be overfulfilled or overreceived
OVERAGE_DISALLWD	Overage is not allowed.
OVERLAPPING_PRDS_ DISALLWD	Illegal period structure. Overlapping periods.
OVERLAPPING_PRDS_ DISALLWD	There is an overlapping period. Please check your Active or Inactive Periods to ensure that there is not an existing period.
OWNER_REQD	You cannot make a contact private without an owner
PACKAGE_WEIGHT_REQD	Attempted to create a package without specifying a nonzero package weight.
PACKG_LEVEL_REF_ DISALLWD	Package level reference numbers are not allowed for shipments whose origin/ destination pair is not US/US or Puerto Rico/Puerto Rico.
PACKG_VALUE_TOO_ LARGE	Package declared value cannot be greater than $999.00 USD
PARENT_CANT_ITSELF_ BE_MEMBER	Parent item can not be a member of itself
PARENT_MUST_BE_ MATRIX_ITEM	A Child matrix item's parent must be a matrix item
PARENT_REQD	A Child matrix item must have its parent specified
PARTIAL_FULFILL_RCEIV_ DISALLWD	Transfer orders can not be partially fulfilled or partially received
PARTNER_ACCESS_ DENIED	Partners do not have access to this item.
PARTNER_CODE_ ALREADY_USED	A partner with that partner code ({1}) already exists.
PAYCHECK_ALREADY_ PAID	You are trying to edit a paycheck that is already paid by direct deposit. A paycheck cannot be edited once funds have been processed by the Automated Clearing House (ACH).
PAYCHECK_IN_USE	You cannot clear this paycheck because it is linked to by one or more liability payments. You must delete or void those transactions first
PAYEE_REQD_FOR_PMT	Your payment has been recorded, but an online bill pay payment will not be made because no payee was specified.You should return to the payment screen if you wish to print the check.
PAYPAL_FUND_SOURCE_ REQD	Please return to PayPal to select a different funding source.

Error Code Returned	Long Description or Message
PAYPAL_INVALID_PMT_METHOD	Paypal is unable to process this payment. Please select an alternate payment method.
PAYPAL_INVALID_PMT_METHOD	Your PayPal account is not configured to use Express Checkout. Please follow directions on the PayPal payment method record.
PAYPAL_PMT_NOTIFICATION	PayPal Payment Notification
PAYPAL_SETUP_REQD	The account referenced by this paypal id is not setup to use express checkout. Please return to the paypal setup page and follow directions for setting up paypal express checkout.
PAYROLL_COMMITTED	You are trying to edit a paycheck reversal that is in a committed Payroll Batch.
PAYROLL_COMMITTED	You are trying to edit a paycheck that is in a committed Payroll Batch.
PAYROLL_COMMITTED	You are trying to modify a committed payroll batch or a document on a committed payroll batch.
PAYROLL_EXPENSE_ACCT_REQD	Please select an expense account for payroll item {2}
PAYROLL_EXPENSE_ACCT_REQD	Please select a expense account for payroll item {2}
PAYROLL_FEATURE_DISABLED	You have not enabled the Payroll feature.
PAYROLL_FEATURE_UNAVAILABLE	You are trying to edit a Pay Cheque - Payroll is not available in NetSuite Canada.
PAYROLL_IN_PROCESS	A payroll process is currently running in this account. Please try again in a few minutes.
PAYROLL_ITEM_DELETE_DISALLWD	Unable to remove payroll item: {1} - There are existing transactions for this payroll item. You may mark it inactive instead.
PAYROLL_LIABILITY_ACCT_REQD	Please select a liability account for payroll item {2}
PAYROLL_MAINTENANCE	The Payroll feature in your account is undergoing routine maintenance from {1} Pacific Time on {2} to {3} Pacific Time on {4}. We apologize for any inconvenience this causes you.
PAYROLL_SETUP_REQD	The default payroll expense account is missing. Please go to Setup -> Payroll -> Set Up Payroll and set the default payroll expense account.
PAYROLL_SETUP_REQD	You are trying to create a paycheck before your payroll setup is complete. Please complete your payroll setup.
PAYROLL_SETUP_REQD	You are trying to process payroll before your payroll setup is complete. Please complete your payroll setup.
PAYROLL_UPDATE_REQD	You made changes that require you to update your payroll information. Click here to commit updates to the payroll service.
PERMISSION_VIOLATION	Permission Violation: you may not access this record.
PERMISSION_VIOLATION	Permission Violation: you may no longer edit this record.
PHONE_NUM_REQD	Please provide a phone number.

Error Code Returned	Long Description or Message
PIN_DEBIT_TRANS_DISALLWD	Non-USD pin debit transactions are not supported. PINNED Debit for USD transactions only.
PLAN_IN_USE	This plan has already been used to generate commission calculations and can't be deleted.
PLAN_OVERLAP_DISALLWD	Plan overlap is not permitted. You have attempted to assign someone to this plan for a time period that overlaps with another plan.
PMT_ALREADY_APPRVD	The payment has already been approved and sent to the bill pay carrier for processing.
PMT_ALREADY_EXISTS	A payment with the same amount and date already exists for this payee.
PMT_ALREADY_SBMTD	This payment has already been submitted for online bill pay.
PMT_EDIT_DISALLWD	Access to this Bill Pay transaction is restricted, and it cannot be modified. Transactions can only be modified until 3PM CST on the payment date.
PMT_EDIT_DISALLWD	This liability payment cannot be edited while it has an Automated Clearing House transmission in process.</TD></TR><TR><TD class=text> </TD></TR><TR><TD class=text> To view the status of payments with ACH transmissions, go to Transactions > View ACH Vendor Payments Status.
POSITIVE_QTY_REQD	Assembly member items must have positive quantities
POSTING_DISCOUNT_DISALLWD	Posting Discounts are not allowed on lines with Revenue Recognition Schedules.
POSTING_PRD_SETUP_REQD	Creation of Journal Entries require a single Accounting Period value across all Revenue Recognition events. Please setup a 'Posting Period' filter.
PRDS_DISALLWD_NAMES_NOT_UNIQUE	After adding new periods, not all names would be unique.
PRD_SETUP_REQD	You must change your period definitions to contain fiscal years. Please visit 'Setup->Manage Accounting Periods' and click 'Set Up Year'.
PRD_SETUP_REQD	You must define the periods of the prior fiscal year. Please visit 'Setup->Manage Accounting Periods' and click 'Set Up Year'.
PRDS_DISALLWD_NAMES_NOT_UNIQUE	After adding new periods, not all names would be unique.
PREF_VENDOR_COST_REQD	Drop ship/Special Order items must have a preferred vendor and a purchase price.
PREF_VENDOR_REQD	Drop ship/Special Order items must have a preferred vendor for each of the {1} the item is accessible to.
PREFERRED_TAX_AGENCY_REQD	A preferred Tax Agency has been deleted - Please choose a new one in Set Up Accounting
PREFERRED_TAX_AGENCY_REQD	Error: No preferred Tax Agencies have been set up (go to Set Up Accounting)
PRIVATE_RCRD_ACCESS_DISALLWD	You cannot view or edit this record because it is marked private
PRIVATE_STATUS_CHNG_DISALLWD	You cannot make this contact private.
PSWD_EXPIRED	Password has expired. Please change your Netsuite password before continuing.

Error Code Returned	Long Description or Message
PSWD_REQD	A password must be entered when granting login access privileges to this record.
PSWD_REQD	Password is empty.
PSWD_REQD	Please type your password into both fields.
PSWD_REQD	You must provide a password to give this person access to your account.
PWSDS_DONT_MATCH	New passwords don't match.
PWSDS_DONT_MATCH	The passwords you entered do not match. Please reenter your passwords.
PWSDS_DONT_MATCH	The Passwords you entered do not match. Please reenter your passwords.
PWSDS_DONT_MATCH	The passwords you have entered do not match.
QTY_EXCEEDED_QTY_BUCKETS	More quantities defined than there are quantity buckets
QTY_REQD	Quantities must be defined
RATE_REQUEST_SHPMNT_REQD	The rate request shipment value has not been set.
RATE_SRVC_UNAVAILBL	The rate for this service is not available for the specified source and destination addresses.
RCRD_DELETED_SINCE_RETRIEVED	Items you have requested in the record have been deleted since you retrieved the form
RCRD_DELETED_SINCE_RETRIEVED	The record has been deleted since you retrieved it.
RCRD_DSNT_EXIST	Group Record Not Found
RCRD_DSNT_EXIST	That record does not exist.{1}
RCRD_DSNT_EXIST	There are no records of this type.
RCRD_EDITED_SINCE_RETRIEVED	The record has been edited since you retrieved it. Hit the back button and click Refresh/Reload to retrieve the updated record, then resubmit your changes.
RCRD_HAS_BEEN_CHANGED	Record has been changed
RCRD_ID_NOT_INT	Record id is not integer: {1}
RCRD_LOCKED_BY_WF	This record has been locked by a user defined workflow.
RCRD_NOT_FOUND	Could not find any records by this name.
RCRD_PREVSLY_DELETED	This record has already been deleted.
RCRD_PREVSLY_DELETED	This record has been deleted since the list was generated.
RCRD_REF_RCRD_TYP_MISMATCH	The record type and its object reference are not matched.
RCRD_SUB_MISMATCH_WITH_CLASS	The subsidiary restrictions on this record are incompatible with those defined for class: {1}. Subsidiary access on this record must be a subset of those permitted by the class.
RCRD_TYPE_REQD	The record type is required.
RCRD_UNEDITABLE	That record is not editable.
REACHED_LIST_END	You have reached the end of the list.

Error Code Returned	Long Description or Message
REACHED_LIST_START	You have reached the start of the list.
REC_TYP_REQD	You must provide either standard or custom record type information.
RECALCING_PLAN_SCHDUL	Cannot inactivate a plan when schedules in the plan are recalculating. Try again when recalculation is complete.
RECUR_EVENT_DISALLWD	A yearly event cannot be on the 29th of February
RECURSV_REF_DISALLWD	ERROR: Recursive Reference
REPORT_EXPIRED	Your report request has expired. A newly created report definition will normally expire after 15 minutes of inactivity unless the definition is saved. Click back to the report definition page and re-submit your report request.
REQD_FORM_TAG_MISSING	Your HTML template file is missing some mandatory fields or a form tag. Please make the changes to the form and try again
REQD_FORM_TAG_MISSING	Your Online HTML Form template is missing a required closing FORM tag
REQD_FORM_TAG_MISSING	Your Online HTML Form template is missing one or more of the required following required tags: <HTML>, <HEAD>
REQD_FORM_TAG_MISSING	Your Online HTML template is missing the required form tag for the main form: <NLFORM>
REQD_LOC_FIELDS_MISSING	Location {1} does not have all required fields set. Please ensure the State, Zip/Postal Code, and Country fields are set and try your request again.
REQD_SUB_FIELDS_MISSING	The Subsidiary {1} does not have all required fields set. Please ensure the State, Zip/Postal Code, and Country fields are set and try your request again.
REQUEST_PARAM_REQD	This request is missing a required parameter.
REV_REC_DATE_REQD	No Revenue Recognition Start Date Specified
REV_REC_TMPLT_DATA_MISSING	One or more line items on this transaction have Variable Revenue Recognition Templates, but do not have the required {1} also populated. Please either change the Template for these items or indicate which {1} will be used to schedule the recognition of revenue.
REV_REC_UPDATE_DISALLWD	Modification of revenue recognition related information on this item is not allowed because revenue has been recognized for this or related lines.
REVERSAL_DATE_WARNING	Reversal Date is in a closed accounting period. Please go to Manage Accounting Periods and re-open the accounting period.
ROLE_REQD	Please specify a role to which access should be granted
ROLE_REQUIRED	In order to login, a role is required unless a default has been previously set.
ROUNDING_DIFF_TOO_BIG	rounding difference too big -> tax1: {1} /tax2: {2}
ROUNDING_ERROR	Rounding Error: {1}
ROUTING_NUM_REQD	Missing Routing Number. Bank routing number is a required field and it cannot be null or empty.
SALES_DISCOUNT_ACCT_REQD	Please set the Sales Discount Account Preference

Error Code Returned	Long Description or Message
SAME_ACCT_TYP_REQD_FOR_PARENT	Parent acccount must be of same account type.
SAVED_SRCH_EMAIL_ERROR	E-mail Alert Failure using saved search '{1}'. Reason: {2}
SCHDUL_EDIT_DISALLWD	This schedule cannot be edited as it has already been used for commission calculations. Please go back and select 'save as new' instead.
SCHEDULED_REPORT_ERROR	NetSuite encountered an error while trying to generate your scheduled report.
SCHEDULED_REPORT_ERROR	NetSuite encountered an error while trying to generate your scheduled report. Please click {1}here{2} to notify NetSuite Support.
SCHEDULED_REPORT_ERROR	NetSuite encountered an error while trying to generate your scheduled report. This error most likely occurred because the query was too large. To run this report successfully, try using filters, or select a different date/period range to narrow the query.
SCHEDULED_REPORT_ERROR	NetSuite encountered an error while trying to generate your scheduled report. This error most likely occurred because the query was too large. To run this report successfully, try using filters, or select a different date/period range to narrow the query. If you continue to encounter errors, please click {1}here{2} to notify NetSuite Support.
SEARCH_DATE_FILTER_REQD	The search must have a date column as an available filter
SEARCH_ERROR	Unable to search - Unexpected error in search engine.
SEARCH_INTEGER_REQD	Please enter an integer number to search on.
SEARCH_TIMED_OUT	Your search has timed out. If your search includes the '{1}' operator, try using '{2}' instead. If your search includes broad search criteria, try narrowing the criteria.
SEARCH_TIMED_OUT	Your search has timed out. This might be avoided by using a smaller page size.
SECURE_TRANS_REQD_ON_CHECKOUT	Store Server Error: As configured, this server does not permit secure transactions, required by store checkout.
SERIAL_NUM_MATCH_MULTI_ITEMS	{1} different items match this serial number. Select an item from the item dropdown.
SESSION_TERMD_2ND_LOGIN_DECTD	You can have a maximum of {1} active users at a time in {2}. If you would like to add active users, please contact your account manager to discuss your upgrade options. Or, you may choose to inactivate an existing user before adding a new one. Note that Employee Center users are not included in this total
SESSION_TERMD_2ND_LOGIN_DECTD	You can have a maximum of {1} active users at a time that are enabled for Offline Sales Client. If you would like to add active users, please contact your account manager to discuss your upgrade options. Or, you may choose to disable Offline Sales Client on an existing user before enabling a new one.
SESSION_TIMED_OUT	Your connection has timed out. Please log in again.
SESSION_TIMED_OUT	Your connection has timed out. Please log in again.
SESSION_TIMED_OUT	Your session has timed out. Please re-enter your information and try again.
SESSION_TIMED_OUT	Your session has timed out. Please re-print the document(s).

Error Code Returned	Long Description or Message
SET_SHIPPING_PICKUP_TYP	Please verify that you have selected a pickup type under setup shipping.
SETUP_METER_REQD	Please set up this meter
SHIP_ADDR_REQD	Shipping address is incomplete.
SHIP_MANIFEST_ALREADY_PRCSSD	A Shipping Manifest has already been processed for the requested date/time {1}.
SHIP_MANIFEST_ERROR	No Shipments found to generate a Shipping Manifest for close date {1} for meter {2}.
SHIP_MANIFEST_ERROR	No Shipping Manifest files found in FedEx Directory for report only request for meter {1}.
SHIP_SETUP_REQD	No {1} registration was found for the location selected. Please select a different shipping item, or go to Setup > Set Up Shipping to register a {2} account for this location.
SHIP_TALBE_UNBALNCD	The Shipping Table is not balanced. Please review the table and ensure there is a Charge for every Range Value, and that there are no duplicates
SHIPMNT_INSURANCE_NOT_AVAILABLE	Insurance is not available when shipping to the destination country: {1}
SINGLE_VALUE_REQD	Multiple values found for a dropdown field that can only take one.
SITE_DOMAIN_NAME_REQD	Notice: URL Components cannot be used until you have established a domain name for your site
SITE_TAG_ALREADY_EXISTS	This site tag already exists
SITEMAP_GEN_ERROR	An error occured while trying to generate the Sitemap.
SO_HAS_CHILD_TRANS	This salesOrder has a one or more child transactions associated with it, and cannot be updated.
SO_LINE_HAS_PO	Error: A Drop Ship/Special Order already exists for sales order {1}, line {2}.
SRVC_UNAVAILBL_FOR_LOC	The requested service is unavailable between the selected locations.
SSS_AUTHOR_MUST_BE_EMPLOYEE	The author internal id or email must match an employee.
SSS_DEBUG_DISALLWD	Script Debugging is not allowed on this server.
SSS_DUP_DRIP_EMAIL	You are attempting to send the same campaign email twice to the same recipient.
SSS_FILE_SIZE_EXCEEDED	The file you are trying to load exceeds the maximum allowed file size of {1} megabyte.
SSS_INSTRUCTION_COUNT_EXCEEDED	Script Execution Instruction Count Exceeded.
SSS_INVALID_ATTACH_RECORD_TYPE	Attaching of record type {1} to {2} is not supported.
SSS_INVALID_BCC_EMAIL	One or more bcc emails are not valid.
SSS_INVALID_CC_EMAIL	One or more cc emails are not valid.

Error Code Returned	Long Description or Message
SSS_INVALID_CMPGN_ EVENT_ID	That campaign event is invalid, disabled, or is not a Lead Nurturing Email event. Please select an active Lead Nurturing Email campaign event.
SSS_INVALID_EMAIL_ TEMPLATE	That email template is invalid, disabled, or no longer exists. Please select an active email template.
SSS_INVALID_FORM_ ELEMENT_NAME	You have entered an invalid form element name. It must be prefixed with "custpage", unique, and cannot contain any non-alphanumeric characters in order to be added to the form or sublist.
SSS_GSO_FLTR_OPRTOR	SSS Invalid GetSelectOption Filter Operator {1}, must be one of the following: {2}
SSS_INVALID_HEADER	One or more headers are not valid.
SSS_INVALID_HOST_ CERT	An untrusted, unsupported, or invalid certificate was found for this host.
SSS_INVALID_LIST_ COLUMN_NAME	You have entered an invalid list column name. It must be unique and cannot contain any non-alphanumeric characters.
SSS_INVALID_LOCK_ WAIT_TIME	You have entered an invalid wait time ({1}) for acquiring a lock. The wait time must be greater than 0ms and less than 5000ms.
SSS_INVALID_LOG_TYPE	Execution log type must be one of AUDIT, DEBUG, ERROR, or EMERGENCY.
SSS_INVALID_PORTLET_ INTERVAL	You have entered an invalid refresh interval of {1} seconds. It must not be negative.
SSS_INVALID_PORTLET_ INTERVAL	You have entered an invalid refresh interval of {1} seconds. It must be at least {2} seconds if not in testing status.
SSS_INVALID_SCRIPTLET_ ID	That Suitelet is invalid, disabled, or no longer exists.
SSS_INVALID_SRCH_COL	An nlobjSearchColumn contains an invalid column, or is not in proper syntax: {1}.
SSS_INVALID_SRCH_ COLUMN_JOIN	An nlobjSearchColumn contains an invalid column join ID, or is not in proper syntax: {1}.
SSS_INVALID_SRCH_ COLUMN_SUM	An nlobjSearchColumn contains an invalid column summary type, or is not in proper syntax: {1}.
SSS_INVALID_SRCH_ FILTER	An nlobjSearchFilter contains invalid search criteria: {1}.
SSS_INVALID_SRCH_ FILTER_JOIN	An nlobjSearchFilter contains an invalid join ID, or is not in proper syntax: {1}.
SSS_INVALID_SRCH_ OPERATOR	An nlobjSearchFilter contains an invalid operator, or is not in proper syntax: {1}.
SSS_INVALID_SUBLIST_ OPERATION	You have attempted an invalid sublist or line item operation. You are either trying to cannot access a field on a non-existent line or you are trying to add or remove lines from a static sublist.
SSS_INVALID_SUBMIT_ OPTION	You have entered an invalid submit option for this record type: {1}
SSS_INVALID_TYPE_ARG	You have entered an invalid type argument: {1}
SSS_INVALID_UI_ OBJECT_TYPE	That operation is not supported for this type of UI object: {1}. It is only supported for type: {2}.

Error Code Returned	Long Description or Message
SSS_INVALID_URL	The URL must be a fully qualified HTTP or HTTPS URL if it is referencing a non-NetSuite resource.
SSS_INVALID_URL_CATEGORY	The URL category must be one of RECORD, TASKLINK or SUITELET.
SSS_INVALID_WF_RCRD_TYPE	You have entered an invalid record type {1}. Workflow automation is not supported for this type of record.
SSS_MISSING_REQD_ARGUMENT	{1}: Missing a required argument: {2}
SSS_QUEUE_LIMIT_EXCEEDED	Script Queue Usage Limit Exceeded
SSS_RECORD_TYPE_MISMATCH	The record you are attempting to load has a different type: {1} from the type specified: {2}.
SSS_REQUEST_TIME_EXCEEDED	The host you are trying to connect to is not responding or has exceeded the maximum allowed response time.
SSS_SEARCH_TIMEOUT	Your search request has timed out. You may need to refine your search or combine the results of multiple searches to achieve the desired result.
SSS_SSO_CONFIG_REQD	The SuiteSignOn object {1} is not configured for use with this script. You must specify the script as a connection point for this SuiteSignOn.
SSS_TIME_LIMIT_EXCEEDED	Script Execution Time Exceeded.
SSS_TRANS_IN_PROGRESS	You cannot call nlapiBeginTransaction() because there is already a transaction in progress.
SSS_TRANSACTION_REQD	That operation can only be performed when there is a transaction in progress
SSS_UNKNOWN_HOST	The host you requested {1} is unknown or cannot be found.
SSS_USAGE_LIMIT_EXCEEDED	Script Execution Usage Limit Exceeded
START_DATE_AFTER_END_DATE	The start date must preceed the end date.
START_DATE_REQD	Please enter a value for {1} Start Date
STATE_ALREADY_EXISTS	A State/Province/County with the same name already exists.
STATE_REQD	State is a required field and it cannot be null or empty.
STATUS_ASSIGNEE_REQD	The issue status {1} does not define an assignee issue role. That status may not be used until this is corrected.
STORAGE_LIMIT_EXCEEDED	You entered a value that will exceed the internal storage limit of {1}. Please reduce the number.
STORE_ALIAS_UNAVAILABLE	The Store alias you chose "{1}" is already taken. Please go back and choose another.
STORE_DOMAIN_UNAVAILABLE	The store domain name you chose '{1}' is already taken. Please go back and choose another.

Error Code Returned	Long Description or Message
SUB_MISMATCH	Resources for Time and Material projects must be in the customer's subsidiary. The following resources are in a different subsidiary: {1}
SUB_RESTRICT_VIEW_REQD	Subsidiary Restrict View Required: You must first restrict your view to a subsidiary before running this report (Home > Set Preferences >Restrict View).
SUB_TAX_AGENCY_REQD	No tax agency defined for subsidiary
SUBITEM_REQD	You must first select the new subitems on the matrix tab you want to add.
SUBITEM_REQD	You must first select the subitems on the matrix tab you want to create.
SUBSIDIARY_MISMATCH	The employee and billable customer must be in the same subsidiary.
SUCCESS_TRANS	The transaction was entered {1} successfully, {2}
SUPRT_CNTR_LOGIN_ERROR	{1} Support Center login error: we are unable to find the customer record for account={2}
TAG_ALREADY_EXISTS	That tag name is already being used. Please go back and rename it.
TAG_SUBSTITUTN_ERROR	Error: Tag substitution loop encountered. Tag substitution sequence: {1}
TAG_SUBSTITUTN_ERROR	Error: Tag substitution result is too large. Tag substitution sequence: {1}
TAGATA_ALREADY_ENDORSED	The Receivable Tegata has already been endorsed.
TAX_ACCT_SETUP_REQD	Tax Accounts Not Defined.
TAX_CODE_REQD	No default tax code is defined for country {1}
TAX_CODES_SETUP_PROBLEM	The tax codes haven't been set properly
TAX_CODES_SETUP_REQD	Can't open store for {1}. This company does not have its tax codes fully set up. This is required to properly calculate taxes on international, other-province and same-province orders.
TAX_CODES_SETUP_REQD	The company is not usable. Administrator hasn't set up the tax codes.
TAX_GROUP_SETUP_REQD	You have not created tax groups in your NetSuite account. To ensure that your customers are charged the correct amount of sales tax, you must create tax groups by entering them manually at Lists > Accounting > Tax Groups > New.
TAX_PRD_REQD	No Current Tax Period is defined. Click here to create a tax period.
TAX_SETUP_REQD	The tax period range {1} has not been defined. Please visit '<A href='{2}'>Setup > Accounting > Manage Tax Periods</A>' to define this period or set up your year.
TEMPLATE_NOT_FOUND	Template not found
TEMPLATE_NOT_FOUND	Template Record not found
THIRD_PARTY_BILLING_ACCT_REQD	A 3rd Party Billing Account Number must be provided when selecting a 3rd Party Billing Type.
TICKET_NOT_LOCATED	The ticket {1} cannot be located in the error database. If this is from a customer logged case, the error may not yet be inserted into the system.
TIME_ENTRY_DISALLWD	{1} does not allow time entry.

Error Code Returned	Long Description or Message
TOPIC_REQD	You must select and add a topic to this solution.
TRAN_DATE_REQD	Missing transaction date.
TRAN_LINE_FX_AMT_REQD	Missing foreign currency amount on non-variance transaction line
TRAN_LINK_FX_AMT_REQD	Missing foreign currency amount on non-variance transaction link
TRAN_PERIOD_CLOSED	The action is causing generation of foreign exchange variance in a closed period. Please retry with {1} as the exchange rate.
TRAN_PERIOD_CLOSED	You are not allowed to change the revenue recognition status for one or more lines on this transaction as it would impact a closed period.
TRAN_PERIOD_CLOSED	You cannot change the G/L impact of a transaction in a closed period.
TRAN_PRD_CLOSED	This action cannot be completed because it requires modification of the transaction in a closed period due to foreign exchange variance. You may either open the period for this transaction or use the same rate ({1})between the transactions that will be linked.
TRANS_ALREADY_REFUNDED	Transaction Already refunded. A refund has already been performed on the transaction.
TRANS_ALREADY_SETTLED	Transaction Already Settled. Void failed because transaction has already settled, submit credit.
TRANS_ALREADY_VOIDED	Transaction Already Voided. Void failed because transaction is already voided.
TRANS_AMTS_UNBALNCD	Transaction is not in balance! amounts+taxes+shipping: {1}, total amount: {2}
TRANS_APPLIED_AMTS_UNBALNCD	Transaction is not in balance! Total to apply of ${1} does not equal sum of applied ${2} and unapplied ${3}
TRANS_APPLIED_AMTS_UNBALNCD	Transaction is not in balance! Total to apply of ${1} does not equal sum of payment ${2} and credits ${3} and deposits ${4}
TRANS_CLASS_UNBALNCD	Transaction out of balance for class {1} total = {2}.
TRANS_DEPT_UNBALNCD	Transaction out of balance for department {1} total = {2}.
TRANS_DOES_NOT_EXIST	No transaction exists for that entity.
TRANS_DSNT_EXIST	The transaction you are attempting to access does not exist.
TRANS_EDIT_DISALLWD	This transaction is in a period that has been closed. You may not edit it.
TRANS_EDIT_DISALLWD	You cannot edit this transaction. {1} does not support the imported transaction.
TRANS_FORGN_CRNCY_MISMATCH	Transaction and foreign currency account use different currencies.
TRANS_FORGN_CUR_UNBALNCD	Transaction was not in balance (Foreign currency). Posting total = {1}
TRANS_FORGN_CUR_UNBALNCD	Transaction was not in balance (Foreign currency). Total = {1}

Error Code Returned	Long Description or Message
TRANS_IN_USE	This transaction cannot be deleted because it is linked to by one or more transactions. Click <a href='/app/accounting/transactions/payments.nl?id={1}&label={2}&type={3}&alllinks=T'>here>/a> to see the list of linked transactions.
TRANS_LINE_AND_PMT_UNBALNCD	Transaction is not in balance! Line item sum of ${1} not equal to payment amount ${2}
TRANS_LINES_UNBALNCD	Transaction is not in balance! Line item sum of ${1} does not equal amount of ${2}
TRANS_LINES_UNBALNCD	Transaction is not in balance! Line item sum of ${1} does not equal applied amount of ${2}
TRANS_LOC_UNBALNCD	Transaction out of balance for location {1} total = {2}.
TRANS_NOT_CLEANED	Transaction not cleaned up.
TRANS_NOT_COMPLETED	Transaction was not complete.
TRANS_PRCSSNG_ERROR	Errors occurred while processing the selected transaction. Please process it individually for more information.
TRANS_UNBALNCD	The debits and credits are not balanced on this transaction because amounts entered include more decimal places than are supported for this currency. Please round off the amount of each line to a maximum of {1} decimal places.
TRANS_UNBALNCD	Transaction is not in balance! {1}
TRANS_UNBALNCD	Transaction is not in balance! {1},{2} othercount = {3}
TRANS_UNBALNCD	Transaction out of balance for {1} {2} total = {3}.
TRANS_UNBALNCD	Transaction was not in balance. Posting total = {1}
TRANS_UNBALNCD	Transaction was not in balance. Total = {1}
TRANSACTION_DELETED	The transaction you are attempting to access has been deleted.
TRANSORD_SHIP_REC_MISMATCH	You can not recieve more from a transfer order than you have shipped
TWO_FA_AUTH_REQD	All your other roles require a one-time key at login. Please click "Go Back" to enter a one-time key or contact the company's administrator if you have questions.
TWO_FA_REQD	Two-Factor Authentication required
UNABLE_TO_PRINT_CHECKS	Unable to print checks.
UNABLE_TO_PRINT_DEPOSITS	Unable to print deposits.
UNAUTH_CAMPAIGN_RSPNS_RQST	Unauthorized campaign response request
UNAUTH_UNSUBSCRIBE_RQST	Unauthorized unsubscribe request
UNDEFINED_ACCTNG_PRD	The accounting period range {1} has not been defined. Please visit '<A href='/app/setup/period/fiscalperiods.nl'>Setup > Accounting > Manage Accounting Periods</A>' to define this period or set up your year.

Error Code Returned	Long Description or Message
UNDEFINED_ACCTNG_ PRD	The comparison accounting period range {1} has not been defined. Please visit '<A href='/app/setup/period/fiscalperiods.nl'>Setup > Accounting > Manage Accounting Periods</A>' to define this period or set up your year.
UNDEFINED_ACCTNG_ PRD	The default accounting period for this report has not been defined. Please visit '<A href='/app/setup/period/fiscalperiods.nl'>Setup > Accounting > Manage Accounting Periods</A>' to define this period or set up your year.
UNDEFINED_CSTM_FIELD	Undefined customfield.
UNDEFINED_TAX_PRD	The default tax period for this report has not been defined. Please visit '<A href='/app/setup/period/taxperiods.nl'>Setup > Accounting > Manage Tax Periods</A>' to define this period or set up your year.
UNEXPECTED_ERROR	An error occurred while processing item options.
UNEXPECTED_ERROR	An unexpected error has occurred.
UNEXPECTED_ERROR	An unexpected error has occurred. A FedEx Shipping Label was not generated.
UNEXPECTED_ERROR	An unexpected error has occurred while generating this content.<p>Our Customer Support staff have been notified and are looking into the problem.
UNEXPECTED_ERROR	An unexpected error has occurred while synching a record. Click [OK] to skip the record and continue.
UNEXPECTED_ERROR	An unexpected error has occurred. Technical Support has been alerted to this problem.
UNEXPECTED_ERROR	An unexpected error occurred while extracting email from SMTP server
UNEXPECTED_ERROR	An unexpected error occurred while logging email request completion
UNEXPECTED_ERROR	An unexpected error occurred while logging email request failure
UNEXPECTED_ERROR	An unexpected error occurred while logging email request start
UNEXPECTED_ERROR	An unexpected error occurred while processing the payment.
UNEXPECTED_ERROR	An unexpected error occurred with the group SQL
UNEXPECTED_ERROR	An Unexpected JavaScript Error has occurred
UNEXPECTED_ERROR	Error
UNEXPECTED_ERROR	Error: {1}
UNEXPECTED_ERROR	Please specify an scompid
UNEXPECTED_ERROR	Problem during commission calculation
UNEXPECTED_ERROR	An unexpected error occurred.
UNEXPECTED_ERROR	Dto java class is not defined for {1}.
UNEXPECTED_ERROR	Server error: no dto class is defined for record of type {1}
UNEXPECTED_ERROR	Server error: missing database entries in WSRecordElement and WSNameSpace table for object of {1}
UNEXPECTED_ERROR	Application error: no form request class is defined for record of type {1}
UNIQUE_CONTACT_ NAME_REQD	. Contact names must be unique

Error Code Returned	Long Description or Message
UNIQUE_CUST_EMAIL_REQD	A customer record with this email address already exists. You must enter a unique customer email address for each record you create.
UNIQUE_CUST_EMAIL_REQD	A customer record with this email address already exists. You must enter a unique customer email address for each record you create. To correct this record, click back and enter a new customer email address in the Customer field. Then, click Submit.
UNIQUE_CUST_ID_REQD	A customer record with this ID already exists. You must enter a unique customer ID for each record you create.
UNIQUE_CUST_ID_REQD	A customer record with this ID already exists. You must enter a unique customer ID for each record you create. To correct this record, click back and enter a new customer ID in the Customer field. Then, click Submit.
UNIQUE_ENTITY_NAME_REQD	multiple sub-customers or jobs have name '{1}' which would create a naming conflict upon merge. All names must be unique. Before merging, you must change one of the subs named '{2}' to something else.
UNIQUE_GROUPID_REQD	You must specify exactly one numeric groupId
UNIQUE_PARTNER_CODE_REQD	{1:name of partner record} Code "{2:partner code}" already exists. Please select a unique code for each record.
UNIQUE_QTY_REQD	Quantities must be unique
UNIQUE_RCRD_ID_REQD	A record with this ID already exists. You must enter a unique ID in order to create or update this record.
UNIQUE_SOLUTION_CODE_REQD	A solution with this particular solution code already exists. Please assign a different code.
UNITS_TYP_IN_USE	This units type is used by {1} {2}. You must delete the {2} and all associated transactions in order to delete this units type.
UNKNOWN_CARRIER	Package Tracking is not available for id {1}. Unknown carrier.
UNKNOWN_RCRD_TYPE	Unknown record type
UNKNOWN_SCRIPT_TYP	Unknown Script Type
UNKNWN_ALLOCTN_SCHDUL_FREQ_TYP	Unable to determine allocation schedule frequency type.
UNKNWN_EMAIL_AUTHOR	The author of this email cannot be found.
UNKNWN_EXCHANGE_RATE	Unable to determine the exchange rate for currency symbol {1}. Please create a currency item with an exchange rate for this currency.
UNRECOGNIZED_METHOD	unrecognized method '{1}'
UNSUBSCRIBE_REQD	Unsubscribe is mandatory, please enter a value for this field.
UNSUPRTD_DOC_TYP	You attempted to upload an unsupported document type. Please try again with a selection from the list below:
UPDATE_DISALLWD	Update is not allowed
UPDATE_PRICE_AMT_REQD	Please specify an amount to update prices.

Error Code Returned	Long Description or Message
UPGRADE_WS_VERSION	Could not set '{1}' to field '{2}' of record number {3} due to schema enumeration restriction.
UPGRADE_WS_VERSION	Please consider upgrading to endpoint {1}
UPGRADE_WS_VERSION	Sales order <id {1}> contains item serial/lot numbers that are not supported in your client application. You are not allowed to update serial/lot numbers on this sales order. Contact your software vendor for the latest Web Services upgrade.
UPGRADE_WS_VERSION	Sales order <id {1}> has items with more than one serial/lot numbers that is not supported in your client application. The serial/lot numbers have been removed to successfully return the sales order. Contact your software vendor for the latest Web Services upgrade.
UPGRADE_WS_VERSION	This {1} has multiple {2}s. Web Services schema version {3} or greater is required to modify {2} for this {1}
UPGRADE_WS_VERSION	{1} {2} has multiple shipping routes enabled, which is only supported in version 2008_2 and newer. The shipping information has been omitted to successfully return this record.
UPS_CANT_INTEGRATE_FULFILL	The fulfillment cannot be integrated with UPS because the Shipping Integration Carrier is not set to UPS.
UPS_CONFIG_ERROR	A UPS configuration error occured. Please contact tech support.
UPS_LICENSE_AGREEMNT_REQD	You must agree to the UPS license agreement
UPS_ONLINE_RATE_UNAVAILBL	The UPS Online Realtime Rates System is temporarily unavailable. Please resubmit your rate request in a few minutes.
UPS_ONLINE_RATE_UNAVAILBL	UPS did not return any rates for the specified origin and destination addresses.
UPS_ONLINE_SHIP_UNAVAILBL	The UPS Online Shipping System is temporarily unavailable. Please resubmit your fulfillment in a few minutes.
UPS_REG_NUM_IN_USE	The submitted UPS Registration Number, {1}, is already in use. Please resubmit the registration with a different UPS registration Number.
UPS_SETUP_REQD	No UPS registration was found. Please register your UPS account with NetSuite before attempting to send a fulfillment request to UPS.
UPS_VOID_ERROR	The UPS Void failed due to a system failure.
UPS_XML_ERROR	XML Sent to UPS. UPS returned error code/text:
URL_ID_PARAM_REQD	URL is missing the id parameter. The file could not be retrieved.
URL_REQD	You must enter a URL for this media item.
USER_CONTEXT_REQD	User context is not set
USER_DISABLED	user disabled
USER_ERROR	An error occurred during your last update.
USER_ERROR	A User Error Has Occurred
USER_ERROR	Detach requires an AttachBasicReference
USER_ERROR	Either internalId or externalId is required.

Error Code Returned	Long Description or Message
USER_ERROR	Folder cannot be made a subfolder of itself.
USER_ERROR	Gift Certificate From, Recipient Name, and Recipient Email are required.
USER_ERROR	Invalid Attachment record combination
USER_ERROR	Missing Item Weight or Weight Unit.
USER_ERROR	Missing or Invalid RecordType for AttachTo
USER_ERROR	Must submit a non-abstract instance of baseRef (eg RecordRef, CustomRecordRef) NOT a baseRef
USER_ERROR	Must submit a non-abstract instance of record or searchRecord (eg customer or customerSearchBasic).
USER_ERROR	No budget inserted or updated.
USER_ERROR	{1}
USPS_ACCT_NUM_ ALREADY_EXISTS	There is an existing NetSuite registration for Endicia account number {1}.
USPS_INVALID_ INSURED_VALUE	Insured value exceeds the {1} maximum allowed by Endicia.
USPS_INVALID_ PACKAGING	The Carrier Packaging that you have selected is not valid for this item fulfillment. Usually this indicates the selected packaging cannot be used with the selected USPS shipping method, or the package weight is invalid. Please check the documentation for more details.
USPS_INVALID_PSWD	The Endicia Web Password does not match the Web Password for this USPS Registration account number.
USPS_LABEL_VOIDED	This error required 1 or more labels created for this transaction to be voided at Endicia.
USPS_LABEL_VOIDED	This error required 1 or more labels created for this transaction to be voided at Endicia.
USPS_MAX_ITEM_ EXCEEDED	International USPS fulfillments allow a maximum of 5 unique items per package, due to customs documentation. If more than one package is required, please break up the shipment into multiple fulfillments.
USPS_ONE_PACKAGE_ ALLWD	International USPS fulfillments allow only one package. If more than one package is required, please break up the shipment into multiple fulfillments of one package each.
USPS_PASS_PHRASE_ NOT_UPDATED	The Endica Pass Phrase was not updated: {1}
USPS_REFUND_FAILED	Failed Endicia Refund Request
USPS_REFUND_FAILED	The Endicia Refund Request failed due to a system error.
USPS_RETRY	A response was not received for the USPS funding request. Please try again in a few minutes.
USPS_VALIDATE_ADDR	The address you entered could not be validated. Please verify the city, state, and/ or zip code. You can validate an address by visiting the U.S. Postal Service web site, or the Endicia web site.

Error Code Returned	Long Description or Message
USPS_VERIFY_ TRACKING_NUM	Please verify that the following tracking numbers were created and voided in your Endicia account before proceeding.
USPS_VOID_ERROR	An error was detected during the Endicia Void operation:
USPS_VOID_ERROR	An error was detected during the Endicia Void operation:
VALID_EMAIL_REQD	Missing or invalid email address. Email address is a required field and it cannot be null or empty. The email address must be in a valid format.
VALID_EMAIL_REQD_ FOR_LOGIN	Please enter a valid email address when granting login access privileges to this record.
VALID_FIRST_NAME_ REQD	Missing or invalid First Name. Users first name is a required field and cannot be null or empty.
VALID_LAST_NAME_ REQD	Missing or invalid Last Name. Users last name is a required field and cannot be null or empty.
VALID_LINE_ITEM_REQD	You must have at least one valid line item for this transaction.
VALID_PHONE_NUM_ REQD	Missing or invalid Home phone number. The Home phone number is a required field and it cannot be null or empty. The format of the Home phone number must contain area code plus seven digit number.
VALID_PRD_REQD	Insert Transaction Failure: No valid, open, posting period for date - {1}. Please visit Setup > Manage Accounting Periods to set up a new accounting period.
VALID_PRD_REQD	Insert Transaction Failure: No valid, open, tax period for date - {1}. Please visit Setup > Manage Tax Periods to set up a new tax period.
VALID_PRD_REQD	Update Transaction Failure: No valid, open, {1} period for date - {2}
VALID_URL_REQD	Please go back and provide a valid URL for all five fields on the External tab.
VALID_VERSION_REQD_ IN_URL	If the version parameter is passed through the URL, it MUST contain a valid version in a phased release environment. Valid: {1}
VALID_WORK_PHONE_ REQD	Missing or invalid Work phone number. The Work phone number is a required field and it cannot be null or empty. The format of the Work phone number must contain area code plus seven digit number.
VALID_ZIPCODE_REQD	Missing or invalid ZIP code field. ZIP code is a required field and it cannot be null or empty. ZIP code and state values are checked against an internal database to make sure that ZIP code specified exists in state specified.
VENDOR_TYPE_REQD	No Vendor Type was specified. If creating a Tax Agency, please ensure that the vendor type is active and marked as a tax agency.
VERIFY_DESTNTN_ZIP_ CODE	Please verify that the destination zipcode is correctly specified.
VERIFY_PAYROLL_FUND_ ACCT	The payroll funding account has not been verified. Please verify the payroll funding account
VERIFY_ZIP_CODE_SETUP	Please verify that you have correctly set your zip code under setup company. If you have multi-location enabled, verify that you have set a correct zipcode for each location.
VISA_ERROR	Communication error with Visa. Please retry.
VOID_FAILED	Void Failed. Failed to void transaction, retry void or issue credit.

Error Code Returned	Long Description or Message
VOIDING_REVERSAL_DISALLWD	You may not create a voiding reversal for transactions with inventory impact. To reverse the inventory impact of the transaction, you will need to create an inventory adjustment.
VSOE_CANT_ADD_ITEM_GROUP	When the Is VSOE bundle box is checked, Items for Purchase cannot be added to item groups.
VSOE_REV_REC_TMPLT_REQD	All Lines in a VSOE Bundle with a VSOE Allocation must have a revenue recognition template.
VSOE_TOTAL_ALLOCATION_ERROR	The total vsoe allocation in a bundle must equal the total bundle sales amount.
VSOE_TRAN_VSOE_BUNDLE_ERROR	You have indicated that you would like this transaction to be treated as a Bundle (multi-element arrangement) for VSOE purposes. Please either uncheck the 'Transaction Is VSOE Bundle' checkbox or remove the Item Groups that have the 'Is VSOE Bundle' option specified.
WF_EXEC_USAGE_LIMIT_EXCEEDED	Workflow Execution Usage Limit Exceeded
WORK_DAYS_REQD	Select one or more working days.
WORLDPAY_ERROR	A failure occurred while attempting to connect to WorldPay.
WORLDPAY_ERROR	There was a problem with your WorldPay credentials. Please be sure you are using the correct combination of Installation ID, Merchant Code and XML Password
WRITE_OFF_ACCT_REQD	In order to receive items without restocking, you must first set a value for the write-off account. To set the value of the write-off account, go to Accounting > Accounting Preferences > Order Management > Write-Off Account for Returns.
WS_CONCUR_SESSION_DISALLWD	Someone has logged in as this user from a different web services session. Only one person may login as a given user at a time. As a consequence, this session has been terminated.
WS_CONCUR_SESSION_DISALLWD	Only one request may be made against a session at a time
WS_EXCEEDED_CONCUR_USERS_ALLWD	You can have a maximum of {1} active concurrent WS users at a time in {2}
WS_EXCEEDED_MAX_CONCUR_RQST	The maximum number of eBay Order Imports has exceeded the provisioned quantity. Please contact Customer Support for further assistance.
WS_FEATURE_REQD	You have not enabled web services feature for your account.
WS_INVALID_SEARCH_OPERATN	When using request-level credentials, you must use the {1} operation instead of {2}
WS_LOG_IN_REQD	You must log in before performing a web service operation.
WS_PERMISSION_REQD	You do not have permission to access web services feature.
ZIP_FILE_CONTAINS_VIRUS	The zip file contains a virus {1}. Upload abort.

Warning Status Codes

The code USER_ERROR is returned in the code field of the statusDetail type where the type attribute has a value of warning. A warning message is also returned which varies depending on the exact cause of the warning and is usually self-descriptive. For a complete description of warnings and how they differ from errors and faults, refer to Understanding Web Services Warnings, Errors, and Faults.

Chapter 13 Task IDs

The following table lists currently exposed TaskIDs that can be referenced in an https POST for use in single login between a Web services application and the NetSuite UI. For more information, see Invoking a UI Session for an External Application.

Task ID	Page Label in NetSuite	URL
EDIT_ACCOUNT	New Accounts	/app/accounting/account/account.nl
EDIT_ACCOUNTINGOTHERLIST	New Accounting List Element	/app/common/otherlists/accountingotherlist.nl
EDIT_ACTIVITY	New Activity	/app/crm/calendar/activity.nl
EDIT_ALLOCATION	Create Allocation Schedules	/app/accounting/transactions/allocation.nl
EDIT_AMENDW4	Form W-4	/app/common/entity/amendw4.nl
EDIT_AMORTIZATIONSCHEDULE	New Amortization Template	/app/accounting/otherlists/revrecschedule.nl?type=Amortization
EDIT_BILLINGSCHEDULE	New Billing Schedule	/app/accounting/otherlists/billingschedule.nl
EDIT_BINNUMBERRECORD	New Bin	/app/accounting/transactions/inventory/binnumberrecord.nl
EDIT_BULKOP	Edit Mass Update	/app/common/bulk/bulkop.nl
EDIT_BUNDLE	Create Bundle	/app/setup/assistants/bundlebuilder.nl?new=T
EDIT_CALENDARPREFERENCE	Calendar Preference	/app/crm/calendar/calendarpreference.nl
EDIT_CALL	New Phone Call	/app/crm/calendar/call.nl
EDIT_CAMPAIGN	New Marketing Campaign	/app/crm/marketing/campaign.nl
EDIT_CAMPAIGNAUDIENCE	New Campaign Audience	/app/crm/marketing/campaignaudience.nl
EDIT_CAMPAIGNBULK	Create Keyword Campaigns	/app/crm/marketing/campaign.nl?bulk=T
EDIT_CAMPAIGNBULKIMPORT	Import Keywords	/app/setup/assistants/nsimport/simpleimport.nl?rectype=CAMPAIGNKEYWORD
EDIT_CAMPAIGNCATEGORY	New Campaign Category	/app/crm/marketing/campaigncategory.nl
EDIT_CAMPAIGNCHANNEL	New Campaign Channel	/app/crm/marketing/campaignchannel.nl
EDIT_CAMPAIGNEMAIL	New Campaign Email Address	/app/crm/marketing/campaignemail.nl
EDIT_CAMPAIGNFAMILY	New Campaign Family	/app/crm/marketing/campaignfamily.nl
EDIT_CAMPAIGNOFFER	New Campaign Offer	/app/crm/marketing/campaignoffer.nl
EDIT_CAMPAIGNSEARCHENGINE	New Campaign Search Engine	/app/crm/marketing/campaignsearchengine.nl
EDIT_CAMPAIGNSUBSCRIPTION	New Campaign Subscription	/app/crm/marketing/campaignsubscription.nl
EDIT_CAMPAIGNVERTICAL	New Campaign Vertical	/app/crm/marketing/campaignvertical.nl
EDIT_CASEFIELDRULE	New Case Rule	/app/crm/support/casefieldrule.nl

Task ID	Page Label in NetSuite	URL
EDIT_CASEFORM	New Online Case Forms	/app/crm/support/caseform.nl
EDIT_CASEISSUE	New Case Issue	/app/crm/support/caseissue.nl
EDIT_CASEORIGIN	New Case Origin Type	/app/crm/support/caseorigin.nl
EDIT_CASEPRIORITY	New Case Priority	/app/crm/support/casepriority.nl
EDIT_CASESTATUS	New Case Status	/app/crm/support/casestatus.nl
EDIT_CASETERRITORY	New Case Territory	/app/crm/support/supportterritory.nl
EDIT_CASETYPE	New Case Type	/app/crm/support/casetype.nl
EDIT_CLASS	New Class	/app/common/otherlists/classtype.nl
EDIT_COLORTHEME	New Color Theme	/app/setup/look/colortheme.nl
EDIT_COMMISSIONSCHEDULE	New Employee Schedule	/app/crm/sales/commissions/commissionschedule.nl
EDIT_COMPETITOR	Competitor	/app/crm/sales/competitor.nl
EDIT_CONTACT	New Contacts	/app/common/entity/contact.nl
EDIT_CRMGROUP	New Groups	/app/crm/common/crmgroup.nl
EDIT_CRMMESSAGE	New Email	/app/crm/common/crmmessage.nl
EDIT_CRMOTHERLIST	New CRM List Element	/app/common/otherlists/crmotherlist.nl
EDIT_CRMTEMPLATE	New Marketing Templates	/app/crm/common/merge/marketingtemplate.nl
EDIT_CURRENCY	New Currencies	/app/common/multicurrency/currency.nl
EDIT_CURRENCYRATE	New Currency Exchange Rate	/app/common/multicurrency/currencyrate.nl
EDIT_CUSTADDRESSFORM	Address Form	/app/common/custom/custaddressform.nl?e=T
EDIT_CUSTBODYFIELD	New Transaction Body Fields	/app/common/custom/bodycustfield.nl
EDIT_CUSTCATEGORY	New Center Category	/app/common/custom/custcategory.nl
EDIT_CUSTCENTER	New Center	/app/common/custom/custcenter.nl
EDIT_CUSTCOLUMNFIELD	New Transaction Column Fields	/app/common/custom/columncustfield.nl
EDIT_CUSTEMAILLAYOUT	New Transaction Form HTML Layouts	/app/common/custom/custemaillayout.nl
EDIT_CUSTENTITYFIELD	New Entity Fields	/app/common/custom/entitycustfield.nl
EDIT_CUSTENTRYFORM	New Entry Forms	/app/common/custom/custentryform.nl
EDIT_CUSTEVENTFIELD	New CRM Fields	/app/common/custom/eventcustfield.nl
EDIT_CUSTFORM	New Transaction Forms	/app/common/custom/custform.nl
EDIT_CUSTITEMFIELD	New Item Fields	/app/common/custom/itemcustfield.nl
EDIT_CUSTITEMNUMBERFIELD	New Item Number Field	/app/common/custom/itemnumbercustfield.nl
EDIT_CUSTJOB	New Customers	/app/common/entity/custjob.nl

Task ID	Page Label in NetSuite	URL
EDIT_CUSTLAYOUT	New Transaction Form PDF Layouts	/app/common/custom/custlayout.nl
EDIT_CUSTLIST	New Lists	/app/common/custom/custlist.nl
EDIT_CUSTOMERFIELDRULE	New Set Up Sales Rules	/app/crm/sales/customerfieldrule.nl
EDIT_CUSTOMERFORM	New Online Customer Forms	/app/crm/sales/leadform.nl
EDIT_CUSTOMERSTATUS	New Customer Statuses	/app/crm/sales/customerstatus.nl
EDIT_CUSTOTHERFIELD	New Other Field	/app/common/custom/othercustfield.nl
EDIT_CUSTPROFILE	Customer Profile	/app/common/entity/custprofile.nl?category=billing&sc=4
EDIT_CUSTRECORD	New Record Types	/app/common/custom/custrecord.nl
EDIT_CUSTRECORDFIELD	Custom Record Field	/app/common/custom/custreccustfield.nl
EDIT_CUSTRECORDFORM	New Custom Record Form	/app/common/custom/custrecordform.nl
EDIT_CUSTSCRIPTFIELD	New Script Fields	/app/common/custom/scriptcustfield.nl
EDIT_CUSTSECTION	New Center Tab	/app/common/custom/custsection.nl
EDIT_CUSTTASKS	Center Links	/app/common/custom/custtasks.nl
EDIT_CUST_	Custom Record Entry	/app/common/custom/custrecordentry.nl
EDIT_DEPARTMENT	New Departments	/app/common/otherlists/departmenttype.nl
EDIT_EDITPROFILE	Employee Profile	/app/common/entity/editprofile.nl
EDIT_EMAILTEMPLATE	New Email Templates	/app/crm/common/merge/emailtemplate.nl
EDIT_EMPLCATEGORY	New Employee Directory	/app/site/setup/emplcategory.nl
EDIT_EMPLOYEE	New Employees	/app/common/entity/employee.nl
EDIT_EMPLOYEESFA	New Assign Support Reps	/app/common/entity/employeesfa.nl
EDIT_EMPOTHERLIST	New Employee Related List Element	/app/common/otherlists/empotherlist.nl
EDIT_ESCALATIONRULE	New Escalation Rules	/app/crm/support/escalationfieldrule.nl
EDIT_ESCALATIONTERRITORY	New Manage Escalation Assignment	/app/crm/support/escalationterritory.nl
EDIT_EVENT	New Event	/app/crm/calendar/event.nl
EDIT_EXPCATEGORY	New Expense Categories	/app/accounting/otherlists/expcategory.nl
EDIT_FAXMESSAGE	New Fax Message	/app/crm/common/merge/faxmessage.nl
EDIT_FAXTEMPLATE	New Fax Templates	/app/crm/common/merge/faxtemplate.nl
EDIT_FISCALPERIOD	Setup Fiscal Period	/app/setup/period/fiscalperiod.nl
EDIT_IC_ALLOCATION	Create Intercompany Allocation Schedules	/app/accounting/transactions/intercompanyallocation.nl
EDIT_INFOITEM	New Information Items	/app/site/setup/infoitem.nl?Information_TYPE=TEXT

Task ID	Page Label in NetSuite	URL
EDIT_INFOITEMFORM	New Publish Forms	/app/site/setup/infoitem.nl?Information_TYPE=FORM
EDIT_INSTALLBUNDLE	Install Bundle	/app/bundler/installbundle.nl
EDIT_ISSUE	New Issue	/app/crm/support/issuedb/issue.nl
EDIT_ISSUEPRODUCT	New Product	/app/crm/support/issuedb/issueproduct.nl
EDIT_ISSUETAG	New Issue Tag	/app/crm/support/issuedb/issuetag.nl
EDIT_ISSUEUSERTYPE	New Issue Role	/app/crm/support/issuedb/issueusertype.nl
EDIT_ITEM	New Items	/app/common/item/item.nl
EDIT_ITEMOPTION	New Transaction Item Options	/app/common/custom/itemoption.nl
EDIT_JOB	New Job	/app/common/entity/custjob.nl?job=T
EDIT_KBCATEGORY	New Knowledge Base	/app/site/setup/kbcategory.nl
EDIT_KPIREPORT	New KPI Scorecard	/app/center/enhanced/kpireportsetup.nl
EDIT_LEAD	New Leads	/app/common/entity/custjob.nl?stage=lead
EDIT_LOCATION	New Locations	/app/common/otherlists/locationtype.nl
EDIT_MAILMERGE	Bulk Merge	/app/crm/common/merge/mailmerge.nl
EDIT_MAILMESSAGE	New Word Message	/app/crm/common/merge/mailmessage.nl
EDIT_MAILTEMPLATE	New Letter Templates	/app/crm/common/merge/mailtemplate.nl
EDIT_MEDIAITEM	New Media Item	/app/common/media/mediaitem.nl
EDIT_MEDIAITEMFOLDER	New File Cabinet	/app/common/media/mediaitemfolder.nl
EDIT_MEMDOC	Enter Memorized Transactions	/app/accounting/transactions/bulkmemdoc.nl
EDIT_NEXUS	Nexus	/app/setup/nexus.nl
EDIT_OPENIDSSO	OpenID Single Sign-on	/app/setup/openidsetup.nl
EDIT_OTHERNAME	New Other Names	/app/common/entity/othername.nl
EDIT_PARTNER	New Partners	/app/common/entity/partner.nl
EDIT_PARTNERCOMMISSIONSCHED	New Partner Schedule	/app/crm/sales/commissions/partnercommissionschedule.nl
EDIT_PARTNERPLANASSIGN	New Partner Plan	/app/crm/sales/commissions/planassignpartner.nl
EDIT_PAYROLLITEM	New Payroll Items	/app/common/item/payrollitem.nl
EDIT_PDFMESSAGE	New PDF Message	/app/crm/common/merge/pdfmessage.nl
EDIT_PDFTEMPLATE	New PDF Templates	/app/crm/common/merge/pdftemplate.nl
EDIT_PERIOD	New Manage Accounting Periods	/app/setup/period/fiscalperiod.nl
EDIT_PLANASSIGN	New Employee Plan	/app/crm/sales/commissions/planassign.nl
EDIT_PRESCATEGORY	New Categories	/app/site/setup/prescategory.nl
EDIT_PROJECTTASK	New Job Tasks	/app/accounting/project/projecttask.nl

Task ID	Page Label in NetSuite	URL
EDIT_PROSPECT	New Prospects	/app/common/entity/custjob.nl?stage=prospect
EDIT_QUANTITYPRICINGSCHEDULE	New Quantity Pricing Schedule	/app/accounting/otherlists/quantitypricingschedule.nl
EDIT_REDIRECT	New Redirect	/app/site/setup/redirect.nl
EDIT_REFERRALCODE	New Promotion Codes	/app/crm/sales/referralcode.nl
EDIT_RELATEDITEM	New Related Items Category	/app/site/setup/relateditem.nl
EDIT_REPORTSECTIONTYPE	New Financial Financial Type	/app/reporting/reportsectiontypes.nl
EDIT_RESOLVECONFLICTS	Resolve Conflicts	/app/common/entity/conflictresolution.nl
EDIT_RESOURCE	Resource	/app/crm/calendar/resource.nl
EDIT_REVRECSCHEDULE	New Revenue Recognition Template	/app/accounting/otherlists/revrecschedule.nl
EDIT_ROLE	New Role	/app/setup/role.nl
EDIT_RPTGROUP	New Financial Statement Section	/app/reporting/reportgroup.nl
EDIT_RPTLAYOUT	New Financial Statement Layout	/app/reporting/reportsectionlayout.nl
EDIT_RSSFEED	New RSS Feed	/app/site/hosting/rssfeed.nl
EDIT_SALESCAMPAIGN	New Sales Campaign	/app/crm/marketing/salescampaign.nl
EDIT_SALESTEAM	New Sales Teams	/app/crm/common/crmgroup.nl?grouptype=SalesTeam
EDIT_SALESTERRITORY	New Manage Sales Territories	/app/crm/sales/salesterritory.nl
EDIT_SAVEDSEARCH	New Saved Search	/app/common/search/search.nl?cu=T&e=F
EDIT_SCRIPT	New Script	/app/common/scripting/script.nl
EDIT_SCRIPTEDRECORD	Scripted Record	/app/common/scripting/scriptedrecord.nl
EDIT_SCRIPTRECORD	New Script Deployment	/app/common/scripting/scriptrecord.nl
EDIT_SEARCH	New Search	/app/common/search/search.nl
EDIT_SHIPITEM	New Shipping Items	/app/common/item/shipitem.nl
EDIT_SITEITEMTEMPLAT	Item Template	/app/site/setup/siteitemtemplate.nl
EDIT_SITEMEDIA	Site Media	/app/site/media/sitemedia.nl
EDIT_SITETAG	Web Site Tag	/app/site/setup/sitetag.nl
EDIT_SITETHEME	Web Site Theme	/app/site/setup/sitetheme.nl
EDIT_SOLUTION	New Solutions	/app/crm/support/kb/solution.nl
EDIT_SSCATEGORY	New Publish Saved Search	/app/site/setup/sscategory.nl
EDIT_STATE	New State/Province/County	/app/setup/state.nl
EDIT_STOREITEMLISTLAYOUT	New Layouts	/app/site/setup/storeitemlistlayout.nl
EDIT_STORETAB	New Tabs	/app/site/setup/storetab.nl

Task ID	Page Label in NetSuite	URL
EDIT_SUBSIDIARY	New Subsidiaries	/app/common/otherlists/subsidiarytype.nl
EDIT_SUITESIGNON	SuiteSignOn	/app/setup/ssoapp.nl
EDIT_SUPPORTCASE	New Cases	/app/crm/support/supportcase.nl
EDIT_SYSALERT	System Alert	/app/common/otherlists/systemalert.nl
EDIT_TASK	New Tasks	/app/crm/calendar/task.nl
EDIT_TAXACCT	Tax Control Account	/app/setup/taxacct.nl
EDIT_TAXGROUP	New Tax Groups	/app/common/item/taxgroup.nl
EDIT_TAXITEM	New Tax Codes	/app/common/item/taxitem.nl
EDIT_TAXPERIOD	New Manage Tax Periods	/app/setup/period/taxperiod.nl
EDIT_TAXSCHEDULE	New Tax Schedule	/app/common/item/taxschedule.nl
EDIT_TAXTYPE	Tax Type	/app/setup/taxtype.nl
EDIT_THIRDPARTYTACKIN GPIXEL	Third Party Conversion Tracking URLs	/app/accounting/otherlists/trackingpixel.nl
EDIT_TOPIC	New Topics	/app/crm/support/kb/topic.nl
EDIT_TRANSACTIONLIST	Paycheck History	/app/accounting/transactions/transactionlist. nl?Transaction_TYPE=Paycheck
EDIT_TRAN_ADJJOURNAL	Make Currency Adjustment Journal Entries	/app/accounting/transactions/journal.nl? adjustmentjournal=T
EDIT_TRAN_BINTRNFR	Bin Transfer	/app/accounting/transactions/bintrnfr.nl
EDIT_TRAN_BINWKSHT	Bin Putaway Worksheet	/app/accounting/transactions/binwksht.nl
EDIT_TRAN_BUILD	Build Assemblies	/app/accounting/transactions/build.nl
EDIT_TRAN_CARDCHRG	Use Credit Card	/app/accounting/transactions/cardchrg.nl
EDIT_TRAN_CASHRFND	Refund Cash Sales	/app/accounting/transactions/cashrfnd.nl
EDIT_TRAN_CASHSALE	Enter Cash Sales	/app/accounting/transactions/cashsale.nl
EDIT_TRAN_CHECK	Write Checks	/app/accounting/transactions/check.nl
EDIT_TRAN_COMMISSN	Individual Employee Commission	/app/accounting/transactions/commissn.nl
EDIT_TRAN_CUSTCHRG	Create Statement Charges	/app/accounting/transactions/custchrg.nl
EDIT_TRAN_CUSTCRED	Issue Credit Memos	/app/accounting/transactions/custcred.nl
EDIT_TRAN_CUSTDEP	Record Customer Deposits	/app/accounting/transactions/custdep.nl
EDIT_TRAN_CUSTINVC	Create Invoices	/app/accounting/transactions/custinvc.nl
EDIT_TRAN_CUSTPYMT	Accept Customer Payments	/app/accounting/transactions/custpymt.nl
EDIT_TRAN_CUSTRFND	Issue Customer Refund	/app/accounting/transactions/custrfnd.nl
EDIT_TRAN_DEPAPPL	Apply Customer Deposits	/app/accounting/transactions/depappl.nl
EDIT_TRAN_DEPOSIT	Make Deposits	/app/accounting/transactions/deposit.nl
EDIT_TRAN_ESTIMATE	Prepare Estimates	/app/accounting/transactions/estimate.nl
EDIT_TRAN_EXPREPT	Enter Expense Reports	/app/accounting/transactions/exprept.nl

Task ID	Page Label in NetSuite	URL
EDIT_TRAN_FXREVAL	Revalue Open Currency Balances	/app/accounting/transactions/fxrevalsetup.nl
EDIT_TRAN_ICJOURNAL	Make Intercompany Journal Entries	/app/accounting/transactions/journal.nl?icj=T
EDIT_TRAN_ICJOURNALIMPORT	Import Intercompany Journal Entries	/app/setup/assistants/nsimport/simpleimport.nl?rectype=INTERCOMPANYJOURNALENTRY
EDIT_TRAN_ICTRNFRORD	Enter Intercompany Transfer Orders	/app/accounting/transactions/trnfrord.nl?icto=T
EDIT_TRAN_INVADJST	Adjust Inventory	/app/accounting/transactions/invadjst.nl
EDIT_TRAN_INVDISTR	Distribute Inventory	/app/accounting/transactions/invdistr.nl
EDIT_TRAN_INVTRNFR	Transfer Inventory	/app/accounting/transactions/invtrnfr.nl
EDIT_TRAN_INVWKSHT	Adjust Inventory Worksheet	/app/accounting/transactions/invwksht.nl
EDIT_TRAN_ITEMRCPT	New Item Receipt	/app/accounting/transactions/itemrcpt.nl
EDIT_TRAN_ITEMSHIP	New Item Fulfillment	/app/accounting/transactions/itemship.nl
EDIT_TRAN_JOURNAL	Make Journal Entries	/app/accounting/transactions/journal.nl
EDIT_TRAN_JOURNALIMPORT	Import Journal Entries	/app/setup/assistants/nsimport/simpleimport.nl?rectype=JOURNALENTRY
EDIT_TRAN_LIAADJST	Liability Adjustment	/app/accounting/transactions/liaadjst.nl
EDIT_TRAN_LIABPYMT	Pay Payroll Liabilities	/app/accounting/transactions/liabpymt.nl
EDIT_TRAN_OPPRTNTY	Create Opportunities	/app/accounting/transactions/opprtnty.nl
EDIT_TRAN_PARTNERCOMMISSN	Individual Partner Commission	/app/accounting/transactions/partnercommissn.nl
EDIT_TRAN_PAYCHECK	Create Individual Paycheck	/app/accounting/transactions/paycheck.nl
EDIT_TRAN_PURCHORD	Enter Purchase Orders	/app/accounting/transactions/purchord.nl
EDIT_TRAN_PURCHORD_REQ	Enter Purchase Requests	/app/accounting/transactions/purchord.nl
EDIT_TRAN_REPLENISHLOC	Replenish Location	/app/accounting/transactions/replenishloc.nl
EDIT_TRAN_REVCOMM	RevComm	/app/accounting/transactions/revcomm.nl
EDIT_TRAN_REVCOMRV	RevComRv	/app/accounting/transactions/revcomrv.nl
EDIT_TRAN_RTNAUTH	Issue Return Authorizations	/app/accounting/transactions/rtnauth.nl
EDIT_TRAN_SALESORD	Enter Sales Orders	/app/accounting/transactions/salesord.nl
EDIT_TRAN_TAXLIAB	Write GST Liability	/app/accounting/transactions/vatliab.nl
EDIT_TRAN_TAXLIAB2	Write Tax Liability	/app/accounting/transactions/vatliab.nl
EDIT_TRAN_TAXPYMT	Pay Sales Tax	/app/accounting/transactions/taxpymt.nl
EDIT_TRAN_TAXPYMT2	Pay Sales Tax	/app/accounting/transactions/taxpymt.nl
EDIT_TRAN_TAXPYMTCA	Pay PST	/app/accounting/transactions/taxpymt.nl
EDIT_TRAN_TEGCOLLECT	Collect Tegata	/app/accounting/transactions/tegcollect.nl
EDIT_TRAN_TEGPAY	Pay Tegata	/app/accounting/transactions/tegpay.nl

Task ID	Page Label in NetSuite	URL
EDIT_TRAN_TEGPYBL	Issue Tegata	/app/accounting/transactions/tegpybl.nl
EDIT_TRAN_TEGRCVBL	Receive Tegata	/app/accounting/transactions/tegrcvbl.nl
EDIT_TRAN_TRANSFER	Transfer Funds	/app/accounting/transactions/transfer.nl
EDIT_TRAN_TRNFRORD	Enter Transfer Orders	/app/accounting/transactions/trnfrord.nl
EDIT_TRAN_UNBUILD	Unbuild Assemblies	/app/accounting/transactions/unbuild.nl
EDIT_TRAN_VATLIABAU	Pay Tax Liability	/app/accounting/transactions/vatliab.nl
EDIT_TRAN_VENDAUTH	Enter Vendor Return Authorizations	/app/accounting/transactions/vendauth.nl
EDIT_TRAN_VENDBILL	Enter Bills	/app/accounting/transactions/vendbill.nl
EDIT_TRAN_VENDCRED	Enter Vendor Credits	/app/accounting/transactions/vendcred.nl
EDIT_TRAN_VENDPYMT	Pay Single Vendor	/app/accounting/transactions/vendpymt.nl
EDIT_TRAN_WORKORD	Enter Work Orders	/app/accounting/transactions/workord.nl
EDIT_TRAN_YTDADJST	Create Payroll Adjustment	/app/accounting/transactions/ytdadjst.nl
EDIT_UNITSTYPE	New Units Of Measure	/app/common/units/unitstype.nl
EDIT_UPGRADEBUNDLE	Managed Bundles	/app/bundler/bundlelist.nl?type=S&subtype=m
EDIT_URLALIAS	New Promotional URL	/app/setup/urlalias.nl
EDIT_VENDOR	New Vendors	/app/common/entity/vendor.nl
EDIT_WORKCALENDAR	New Work Calendar	/app/accounting/project/workcalendar.nl
EDIT_WORKFLOW	New Workflow	/app/common/workflow/setup/workflowmanager.nl
EDIT_WORKPLACE	New Workplaces	/app/common/otherlists/workplacetype.nl
INTL_SECTION	New Tabs	/internal/admin/section.nl
INTL_SECTIONS	Tabs	/internal/admin/sections.nl
INTL_TASKLINK	New Tasklinks	/internal/admin/tasklink.nl
INTL_TASKLINKS	Tasklinks	/internal/admin/tasklinks.nl
LIST_ACCOUNT	Accounts	/app/accounting/account/accounts.nl
LIST_ACCOUNTINGOTHERLIST	Accounting Lists	/app/common/otherlists/accountingotherlists.nl
LIST_ACTIVITY	Activity List	/app/crm/calendar/activitylist.nl
LIST_ADP_BATCHES	Payroll Batches	/app/payroll/payrollbatchlist.nl
LIST_ADP_PAYCHECK	Individual Paychecks	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Paycheck
LIST_ALLOCATION	Allocation Schedules	/app/accounting/transactions/allocschedulelist.nl
LIST_AMORTIZATION	Amortization Templates	/app/accounting/otherlists/amortizationtemplates.nl

Task ID	Page Label in NetSuite	URL
LIST_AMORTIZATIONSCHEDULE	Amortization Schedules	/app/accounting/otherlists/amortizationschedules.nl
LIST_APPROVEACH	Approve Direct Deposits	/app/accounting/transactions/approveach.nl
LIST_APPROVEEFT	Approve Electronic Funds Transfers	/app/accounting/transactions/approveeft.nl
LIST_APPROVEVP	Approve Vendor Payment Transfers	/app/accounting/transactions/approvevp.nl
LIST_BILLINGSCHEDULE	Billing Schedules	/app/accounting/otherlists/billingschedules.nl
LIST_BIN	Bins	/app/accounting/transactions/inventory/binlist.nl
LIST_BUDGET	Budgets	/app/accounting/transactions/budgetlist.nl
LIST_BUDGETRATES	Budget Exchange Rates	/app/accounting/otherlists/budgetrates.nl
LIST_BULKOP	Mass Updates	/app/common/bulk/bulkops.nl
LIST_BULKRESULTS	Mass Update Results	/app/common/bulk/bulkresults.nl
LIST_BUNDLE	List Bundles	/app/bundler/bundlelist.nl?type=S
LIST_CALENDAR	Calendar	/app/crm/calendar/calendar.nl
LIST_CALL	Phone Calls	/app/crm/calendar/calllist.nl
LIST_CAMPAIGN	Marketing Campaigns	/app/crm/marketing/campaignlist.nl?Campaign_ISSALESCAMPAIGN=F
LIST_CAMPAIGNAUDIENCE	Campaign Audiences	/app/crm/marketing/campaignaudiences.nl
LIST_CAMPAIGNCALENDAR	Campaign Calendar	/app/crm/marketing/campaigncalendar.nl
LIST_CAMPAIGNCATEGORY	Campaign Categories	/app/crm/marketing/campaigncategories.nl
LIST_CAMPAIGNCHANNEL	Campaign Channels	/app/crm/marketing/campaignchannels.nl
LIST_CAMPAIGNEMAIL	Campaign Email Addresses	/app/crm/marketing/campaignemails.nl
LIST_CAMPAIGNFAMILY	Campaign Families	/app/crm/marketing/campaignfamilies.nl
LIST_CAMPAIGNOFFER	Campaign Offers	/app/crm/marketing/campaignoffers.nl
LIST_CAMPAIGNSEARCHENGINE	Campaign Search Engines	/app/crm/marketing/campaignsearchengines.nl
LIST_CAMPAIGNSUBSCRIPTION	Campaign Subscriptions	/app/crm/marketing/campaignsubscriptions.nl
LIST_CAMPAIGNVERTICAL	Campaign Verticals	/app/crm/marketing/campaignverticals.nl
LIST_CASEFIELDRULE	Set Up Case Rules	/app/crm/support/casefieldrules.nl
LIST_CASEFORM	Online Case Forms	/app/crm/support/caseforms.nl
LIST_CASEISSUE	Case Issues	/app/crm/support/caseissuelist.nl
LIST_CASEORIGIN	Case Origin Types	/app/crm/support/caseoriginlist.nl
LIST_CASEPRIORITY	Case Priorities	/app/crm/support/caseprioritylist.nl
LIST_CASESTATUS	Case Statuses	/app/crm/support/casestatuslist.nl
LIST_CASETERRITORIES	Case Territory List	/app/crm/support/supportterritorylist.nl

Task ID	Page Label in NetSuite	URL
LIST_CASETERRITORY	Manage Case Territories	/app/crm/common/automation/territorymanager.nl?case=T
LIST_CASETERRITORYASSIGN	Territory Reassignment	/app/common/bulk/bulkop.nl?searchtype=Case&opcode=ReAssign
LIST_CASETYPE	Case Types	/app/crm/support/casetypelist.nl
LIST_CCTRAN	View Credit Card Transactions	/app/accounting/transactions/cardtrans.nl
LIST_CHART_ACCOUNT	Chart of Accounts	/app/accounting/account/accounts.nl?report=T&code=COA
LIST_CLASS	Classes	/app/common/otherlists/classlist.nl
LIST_COLORTHEME	Color Themes	/app/setup/look/colorthemes.nl
LIST_COMMISSIONSCHEDULE	Employee Schedules	/app/crm/sales/commissions/commissionscheds.nl
LIST_COMPETITOR	Competitors	/app/crm/sales/competitorlist.nl
LIST_CONSOLRATES	Consolidated Exchange Rates	/app/accounting/otherlists/consolidatedrates.nl
LIST_CONTACT	Contacts	/app/common/entity/contactlist.nl
LIST_COUNTRY	Set Up Countries	/app/setup/countries.nl
LIST_CRMGROUP	Groups	/app/crm/common/crmgrouplist.nl
LIST_CRMOTHERLIST	CRM Lists	/app/common/otherlists/crmotherlists.nl
LIST_CRMTEMPLATE	Marketing Templates	/app/crm/common/merge/marketingtemplates.nl
LIST_CURRENCY	Currencies	/app/common/multicurrency/currencylist.nl
LIST_CURRENCYRATE	Currency Exchange Rates	/app/common/multicurrency/currencyratelist.nl
LIST_CUSTBODYFIELD	Transaction Body Fields	/app/common/custom/bodycustfields.nl
LIST_CUSTCATEGORY	Center Categories	/app/common/custom/custcategories.nl
LIST_CUSTCENTER	Centers	/app/common/custom/custcenters.nl
LIST_CUSTCOLUMNFIELD	Transaction Column Fields	/app/common/custom/columncustfields.nl
LIST_CUSTEMAILLAYOUT	Transaction Form HTML Layouts	/app/common/custom/custemaillayouts.nl
LIST_CUSTENTITYFIELD	Entity Fields	/app/common/custom/entitycustfields.nl
LIST_CUSTENTRYFORM	Entry Forms	/app/common/custom/custentryforms.nl
LIST_CUSTEVENTFIELD	CRM Fields	/app/common/custom/eventcustfields.nl
LIST_CUSTFIELDTAB	Subtabs	/app/common/custom/custfieldtabs.nl
LIST_CUSTFORM	Transaction Forms	/app/common/custom/custforms.nl
LIST_CUSTITEMFIELD	Item Fields	/app/common/custom/itemcustfields.nl
LIST_CUSTITEMNUMBERFIELD	Item Number Fields	/app/common/custom/itemnumbercustfields.nl

Task ID	Page Label in NetSuite	URL
LIST_CUSTJOB	Customers	/app/common/entity/custjoblist.nl?Customer_STAGE=CUSTOMER
LIST_CUSTLAYOUT	Transaction Form PDF Layouts	/app/common/custom/custlayouts.nl
LIST_CUSTLIST	Lists	/app/common/custom/custlists.nl
LIST_CUSTOMCODEFILES	SuiteScripts	/app/common/media/mediaitemfolders.nl?folder=-15
LIST_CUSTOMERFIELDRULE	Set Up Sales Rules	/app/crm/sales/customerfieldrules.nl
LIST_CUSTOMERFORM	Online Customer Forms	/app/crm/sales/leadforms.nl
LIST_CUSTOMERSTATUS	Customer Statuses	/app/crm/sales/customerstatuslist.nl
LIST_CUSTOMSUBLIST	Sublists	/app/common/custom/customsublists.nl
LIST_CUSTOTHERFIELD	Other Custom Fields	/app/common/custom/othercustfields.nl
LIST_CUSTRECORD	Record Types	/app/common/custom/custrecords.nl
LIST_CUSTRECORDFORM	Online Custom Record Forms	/app/common/custom/custrecordforms.nl
LIST_CUSTSECTION	Center Tabs	/app/common/custom/custsections.nl
LIST_DEPARTMENT	Departments	/app/common/otherlists/departmentlist.nl
LIST_DIFFROLE	Show Role Differences	/app/setup/diffroles.nl
LIST_EMAILTEMPLATE	Email Templates	/app/crm/common/merge/emailtemplates.nl
LIST_EMPLCATEGORY	Employee Directory	/app/site/setup/emplcategories.nl
LIST_EMPLOYEE	Employees	/app/common/entity/employeelist.nl
LIST_EMPOTHERLIST	Employee Related Lists	/app/common/otherlists/empotherlists.nl
LIST_ENTITY	All Entities	/app/common/entity/entitylist.nl
LIST_ENTITY_DUPLICATES	Entity Duplicate Resolution	/app/common/entity/manageduplicates.nl
LIST_ESCALATIONRULE	Set Up Escalation Rules	/app/crm/support/escalationfieldrules.nl
LIST_ESCALATIONTERRITORY	Manage Escalation Assignment	/app/crm/common/automation/territorymanager.nl?escalation=T
LIST_EVENT	Events	/app/crm/calendar/eventlist.nl
LIST_EXPCATEGORY	Expense Categories	/app/accounting/otherlists/expcategories.nl
LIST_FAXTEMPLATE	Fax Templates	/app/crm/common/merge/faxtemplates.nl
LIST_FCSITEFOLDER	Web Site Hosting Files	/app/common/media/mediaitemfolders.nl?folder=-100
LIST_FILECABINET	Media Items	/app/common/media/mediaitems.nl
LIST_FINCHRG	Assess Finance Charges	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustInvc&Transaction_FINCHRG=T
LIST_FORECAST	List Sales Rep Forecast	/app/crm/sales/forecastlist.nl?Forecast_ISTEAM=F
LIST_GIFTCERTIFICATES	Gift Certificates	/app/accounting/otherlists/giftcertificates.nl

Task ID	Page Label in NetSuite	URL
LIST_IMAGE	Images	/app/common/media/mediaitemfolders.nl?folder=-4
LIST_INFOITEM	Information Items	/app/site/setup/infoitemlist.nl?searchid=-2540
LIST_INFOITEMFORM	Publish Forms	/app/site/setup/infoitemlist.nl?searchid=-2541
LIST_INSTALLEDBUNDLE	Installed Bundles	/app/bundler/bundlelist.nl?type=I
LIST_ISSUE	Issues	/app/crm/support/issuedb/issuelist.nl
LIST_ITEM	Items	/app/common/item/itemlist.nl
LIST_ITEMOPTION	Transaction Item Options	/app/common/custom/itemoptions.nl
LIST_JOB	Jobs	/app/common/entity/joblist.nl
LIST_KBCATEGORY	Knowledge Base	/app/site/setup/kbcategories.nl
LIST_KPIREPORT	KPI Scorecards	/app/center/enhanced/kpireports.nl
LIST_LEAD	Leads	/app/common/entity/custjoblist.nl?Customer_STAGE=LEAD
LIST_LOCATION	Locations	/app/common/otherlists/locationlist.nl
LIST_MAILMERGE	Merge History	/app/crm/common/merge/mailmergehistory.nl
LIST_MAILTEMPLATE	Letter Templates	/app/crm/common/merge/mailtemplates.nl
LIST_MEDIAITEMFOLDER	File Cabinet	/app/common/media/mediaitemfolders.nl
LIST_MEDIAITEMFOLDER_LOG	Job Status	/app/external/xml/upload/uploadlog.nl?displayType=FILECABINET
LIST_MEMDOC	Memorized Transactions	/app/accounting/otherlists/memdoclist.nl
LIST_MGRFORECAST	List Sales Manager Forecast	/app/crm/sales/forecastlist.nl?Forecast_ISTEAM=T
LIST_MYROLES	My Roles	/app/center/myroles.nl
LIST_OTHERNAME	Other Names	/app/common/entity/othernames.nl
LIST_PARTNER	Partners	/app/common/entity/partnerlist.nl
LIST_PARTNERCOMMISSIONSCHED	Partner Schedules	/app/crm/sales/commissions/partnercommissionscheds.nl
LIST_PARTNERPLANASSIGN	Partner Plans	/app/crm/sales/commissions/partnercommissionplans.nl
LIST_PARTNERPLANASSIGNS	Partner Plan Assignments	/app/crm/sales/commissions/partnerplanassigns.nl
LIST_PAYROLLBATCH	Payroll Batches	/app/payroll/payrollbatchlist.nl
LIST_PAYROLLISSUES	Payroll Issues	/app/payroll/cdstatus.nl
LIST_PAYROLLITEM	Payroll Items	/app/common/item/payrollitems.nl
LIST_PDFTEMPLATE	PDF Templates	/app/crm/common/merge/pdftemplates.nl
LIST_PERIOD	Manage Accounting Periods	/app/setup/period/fiscalperiods.nl

Task ID	Page Label in NetSuite	URL
LIST_PLANASSIGN	Employee Plans	/app/crm/sales/commissions/commissionplans.nl
LIST_PLANASSIGNS	Employee Plan Assignments	/app/crm/sales/commissions/planassigns.nl
LIST_PRESCATEGORY	Categories	/app/site/setup/prescategories.nl?ctype=PRES
LIST_PROJECTTASK	Project Tasks	/app/accounting/project/projecttasks.nl
LIST_PROJECTTASKIMPORT	Import Project Tasks	/app/setup/assistants/nsimport/simpleimport.nl?rectype=PROJECTTASK
LIST_PROSPECT	Prospects	/app/common/entity/custjoblist.nl?Customer_STAGE=PROSPECT
LIST_QUANTITYPRICINGSCHEDULE	Quantity Pricing Schedules	/app/accounting/otherlists/quantitypricingschedules.nl
LIST_QUOTA	Quotas	/app/crm/sales/quotalist.nl
LIST_RECENTRECORDS	Recent Records	/app/common/otherlists/recentrecords.nl
LIST_RECVDFILES	Attachments Received	/app/common/media/mediaitemfolders.nl?folder=-10
LIST_REFERRALCODE	Promotion Codes	/app/crm/sales/referralcodelist.nl
LIST_RELATEDITEM	Related Items Categories	/app/site/setup/relateditems.nl
LIST_REPORESULT	Report Results	/app/reporting/workqueue/reportresultlist.nl
LIST_REPOSCHEDULE	Report Schedules	/app/reporting/workqueue/reportschedulelist.nl
LIST_RESOURCE	Resources	/app/crm/calendar/resources.nl
LIST_REVRECSCHEDS	Revenue Recognition Schedules	/app/accounting/otherlists/revrecschedules.nl
LIST_REVRECSCHEDULE	Revenue Recognition Templates	/app/accounting/otherlists/revrectemplates.nl
LIST_ROLE	Manage Roles	/app/setup/rolelist.nl
LIST_RSSFEED	RSS Feeds	/app/site/hosting/rssfeeds.nl
LIST_SALESCAMPAIGN	Sales Campaigns	/app/crm/marketing/salescampaignlist.nl
LIST_SALESTEAM	Sales Teams	/app/crm/common/salesteamlist.nl
LIST_SALESTERRITORIES	Sales Territory List	/app/crm/sales/salesterritorylist.nl
LIST_SALESTERRITORY	Manage Sales Territories	/app/crm/common/automation/territorymanager.nl?sales=T
LIST_SALESTERRITORYASSIGN	Territory Reassignment	/app/common/bulk/bulkop.nl?searchtype=Customer&opcode=ReAssign
LIST_SAVEDASHBOARD	Published Dashboards	/app/center/setup/savedashboards.nl
LIST_SAVEDBULKOP	Saved Mass Updates	/app/common/bulk/savedbulkops.nl
LIST_SAVEDSEARCH	Saved Searches	/app/common/search/savedsearches.nl
LIST_SCRIPT	Scripts	/app/common/scripting/scriptlist.nl
LIST_SCRIPTCALENDAR	Script Calendar	/app/common/scripting/scriptcalendar.nl
LIST_SCRIPTEDRECORD	Scripted Records	/app/common/scripting/scriptedrecords.nl

Task ID	Page Label in NetSuite	URL
LIST_SCRIPTRECORD	Script Deployments	/app/common/scripting/scriptrecordlist.nl
LIST_SCRIPTSTATUS	Scheduled Script Status	/app/common/scripting/scriptstatus.nl
LIST_SEARCHRESULTS	Search Results	/app/common/search/searchresults.nl
LIST_SENTFILES	Attachments Sent	/app/common/media/mediaitemfolders.nl?folder=-14
LIST_SHIPITEM	Shipping Items	/app/common/item/shipitems.nl
LIST_SHIPPINGMANIFEST	Shipping Manifest	/app/common/shipping/fedex/shippingmanifest.nl
LIST_SITEITEMTEMPLAT	Item/Category Templates	/app/site/setup/siteitemtemplates.nl
LIST_SITETAGS	Web Site Tags	/app/site/setup/sitetags.nl
LIST_SITETHEMES	Web Site Themes	/app/site/setup/sitethemes.nl
LIST_SMBIMPORT	SMB Import	/app/external/xml/upload/upload.nl
LIST_SOLUTION	Solutions	/app/crm/support/kb/solutions.nl
LIST_SSCATEGORY	Publish Saved Search	/app/site/setup/sscategories.nl
LIST_STATE	Set Up States/Provinces/Counties	/app/setup/states.nl
LIST_STATUSACH	View Direct Deposit Status	/app/accounting/transactions/statusach.nl
LIST_STATUSEFT	View Electronic Funds Transfer Status	/app/accounting/transactions/statuseft.nl
LIST_STATUSVP	View Vendor Payment Status	/app/accounting/transactions/statusvp.nl
LIST_STOREITEMLISTLAYOUT	Layouts	/app/site/setup/storeitemlistlayouts.nl
LIST_STORETAB	Tabs	/app/site/setup/storetabs.nl
LIST_SUBSIDIARY	Subsidiaries	/app/common/otherlists/subsidiarylist.nl
LIST_SUPPORTCASE	Cases	/app/crm/support/caselist.nl
LIST_SYSALERT	System Alerts	/app/common/otherlists/systemalerts.nl
LIST_SYSTEMEMAILTEMPLATE	Set Up System Email Templates	/app/crm/common/merge/systememailtemplates.nl
LIST_TASK	Tasks	/app/crm/calendar/tasklist.nl
LIST_TAXGROUP	Tax Groups	/app/common/item/taxgroups.nl
LIST_TAXITEM	Tax Codes	/app/common/item/taxitems.nl
LIST_TAXPERIOD	Manage Tax Periods	/app/setup/period/taxperiods.nl
LIST_TAXSCHEDULE	Tax Schedules	/app/common/item/taxschedules.nl
LIST_TEMPLATEFILES	Template Files	/app/common/media/mediaitemfolders.nl?folder=-9
LIST_THIRDPARTYTRACKINGPIXEL	Third Party Conversion Tracking URLs	/app/site/setup/google/trackingpixels.nl
LIST_TOPIC	Topics	/app/crm/support/kb/topics.nl

Task ID	Page Label in NetSuite	URL
LIST_TRANSACTION	Transactions	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=@ALL@
LIST_TRAN_BINTRNFR	Bin Transfers	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=BinTrnfr
LIST_TRAN_BINWKSHT	Bin Putaway Worksheets	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=BinWksht
LIST_TRAN_BUILD	Assembly Builds	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Build
LIST_TRAN_CARDCHRG	Credit Card Charges	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CardChrg
LIST_TRAN_CASHRFND	Cash Refunds	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CashRfnd
LIST_TRAN_CASHSALE	Cash Sales	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CashSale
LIST_TRAN_CHECK	Checks	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Check
LIST_TRAN_COMMISSN	Individual Employee Commissions	/app/accounting/transactions/commissns.nl
LIST_TRAN_CUSTCHRG	Statement Charges	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustChrg
LIST_TRAN_CUSTCRED	Credit Memos	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustCred
LIST_TRAN_CUSTDEP	Customer Deposits	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustDep
LIST_TRAN_CUSTINVC	Invoices	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustInvc
LIST_TRAN_CUSTPYMT	Customer Payments	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustPymt
LIST_TRAN_CUSTRFND	Customer Refunds	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustRfnd
LIST_TRAN_DEPAPPL	Deposit Applications	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=DepAppl
LIST_TRAN_DEPOSIT	Deposits	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Deposit
LIST_TRAN_DOWNLOAD	My Downloads	/app/common/entity/downloads.nl
LIST_TRAN_ESTIMATE	Estimates	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Estimate
LIST_TRAN_EXPREPT	Expense Reports	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=ExpRept
LIST_TRAN_FXREVAL	Currency Revaluations	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=FxReval

Task ID	Page Label in NetSuite	URL
LIST_TRAN_ICJOURNAL	Intercompany Journal Entries	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Journal&icj=T
LIST_TRAN_ICTRNFRORD	Intercompany Transfer Orders	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TrnfrOrd&icto=T
LIST_TRAN_INVADJST	Inventory Adjustments	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=InvAdjst
LIST_TRAN_INVDISTR	Inventory Distributions	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=InvDistr
LIST_TRAN_INVTRNFR	Inventory Transfers	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=InvTrnfr
LIST_TRAN_INVWKSHT	Inventory Worksheets	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=InvWksht
LIST_TRAN_ITEMRCPT	Item Receipts	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=ItemRcpt
LIST_TRAN_ITEMSHIP	Item Fulfillments	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=ItemShip
LIST_TRAN_JOURNAL	Journal Entries	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Journal
LIST_TRAN_LIABPYMT	Liability Payments	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=LiabPymt
LIST_TRAN_OPPRTNTY	Opportunities	/app/accounting/transactions/opprtntylist.nl
LIST_TRAN_PARTNERCOMMISSN	Individual Partner Commissions	/app/accounting/transactions/partnercommissns.nl
LIST_TRAN_PAYCHECK	Paychecks	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Paycheck
LIST_TRAN_PURCHORD	Purchase Orders	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=PurchOrd
LIST_TRAN_PURCHORD_REQ	View Purchase Requests/Orders	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=PurchOrd
LIST_TRAN_REORDER	Items Ordered	/app/common/item/itemsordered.nl
LIST_TRAN_REVCOMM	Revenue Commitments	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=RevComm
LIST_TRAN_REVCOMRV	View Revenue Commitment Reversals	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=RevComRv
LIST_TRAN_RTNAUTH	Return Authorizations	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=RtnAuth
LIST_TRAN_SALESORD	Sales Orders	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=SalesOrd
LIST_TRAN_TAXLIAB	Tax Liabilities	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TaxLiab
LIST_TRAN_TAXPYMT	Tax Payments	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TaxPymt

Task ID	Page Label in NetSuite	URL
LIST_TRAN_TEGPYBL	Issued Tegatas	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TegPybl
LIST_TRAN_TEGRCVBL	Received Tegatas	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TegRcvbl
LIST_TRAN_TRANSFER	Bank Transfers	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Transfer
LIST_TRAN_TRNFRORD	Transfer Orders	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TrnfrOrd
LIST_TRAN_UNBUILD	Assembly Unbuilds	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Unbuild
LIST_TRAN_VATLIAB	Tax Liabilities	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TaxLiab
LIST_TRAN_VATLIABAU	GST Liabilities	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TaxLiab
LIST_TRAN_VATLIABUK	VAT Liabilities	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TaxLiab
LIST_TRAN_VENDAUTH	Vendor Return Authorizations	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=VendAuth
LIST_TRAN_VENDBILL	Bills	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=VendBill
LIST_TRAN_VENDCRED	Bill Credits	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=VendCred
LIST_TRAN_VENDPYMT	Bill Payments	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=VendPymt
LIST_TRAN_WORKORD	Work Orders	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=WorkOrd
LIST_TRAN_YTDADJST	Payroll Adjustment	/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=YtdAdjst
LIST_UNCATSITEITEM	View Uncategorized Items	/app/setup/uncatsiteitems.nl
LIST_UNITSTYPE	Units Of Measure	/app/common/units/unitstypelist.nl
LIST_UPSELLPOPUP	Upsell Popup	/app/crm/sales/upsell/upsellpopup.nl
LIST_UPSELLWIZARD	Upsell Manager	/app/crm/sales/upsell/upsellmanager.nl
LIST_URLALIASES	Promotional URLs	/app/setup/urlaliases.nl
LIST_USER	Manage Users	/app/setup/listusers.nl
LIST_VENDOR	Vendors	/app/common/entity/vendorlist.nl
LIST_WORKCALENDAR	Work Calendars	/app/accounting/project/workcalendars.nl
LIST_WORKFLOW	Workflows	/app/common/workflow/setup/workflowlist.nl
LIST_WORKPLACE	Workplaces	/app/common/otherlists/workplacelist.nl
SRCH_ACCOUNT	Search Account	/app/common/search/search.nl?searchtype=Account

Task ID	Page Label in NetSuite	URL
SRCH_ACTIVITY	Search Activities	/app/common/search/search.nl?searchtype=Activity
SRCH_AMORTIZATIONSCHEDULE	Search Amortization Schedules	/app/common/search/search.nl?searchtype=AmortizationSchedule
SRCH_BIN	Search Bin	/app/common/search/search.nl?searchtype=BinNumber
SRCH_BUDGET	Search Set Up Budget	/app/common/search/search.nl?searchtype=Budget
SRCH_BUDGETRATES	Search Budget Rates	/app/common/search/search.nl?searchtype=BudgetExchangeRate
SRCH_CALENDAR	Search Calendar/Event	/app/common/search/search.nl?searchtype=Calendar
SRCH_CALL	Search Calls	/app/common/search/search.nl?searchtype=Call
SRCH_CAMPAIGN	Search Campaigns	/app/common/search/search.nl?searchtype=Campaign
SRCH_CASE	Search Case	/app/common/search/search.nl?searchtype=Case
SRCH_CLASS	Search Class	/app/common/search/search.nl?searchtype=Class
SRCH_COMPANY	Search All Companies	/app/common/search/search.nl?searchtype=Company
SRCH_COMPETITOR	Search Competitors	/app/common/search/search.nl?searchtype=Competitor
SRCH_CONSOLEXCHANGERATE	Consolidated Exchange Rates	/app/common/search/search.nl?searchtype=ConsolExchangeRate
SRCH_CONSOLRATES	Search Consolidated Rates	/app/common/search/search.nl?searchtype=ConsolExchangeRate
SRCH_CONTACT	Search Contact	/app/common/search/search.nl?searchtype=Contact
SRCH_CRMGROUP	Search Group	/app/common/search/search.nl?searchtype=CRMGroup
SRCH_CUSTOMER	Search Customer	/app/common/search/search.nl?searchtype=Customer
SRCH_DEPARTMENT	Search Department	/app/common/search/search.nl?searchtype=Department
SRCH_DOCUMENT	Search File Cabinet	/app/common/search/search.nl?searchtype=Document
SRCH_EMPLOYEE	Search Employee	/app/common/search/search.nl?searchtype=Employee
SRCH_ENTITY	Search All Entities	/app/common/search/search.nl?searchtype=Entity

Task ID	Page Label in NetSuite	URL
SRCH_FIRSTVISIT	Search First Site Visit	/app/common/search/search.nl?searchtype=FirstVisit
SRCH_GIFTCERTIFICATE	Search Gift Certificates	/app/common/search/search.nl?searchtype=GiftCertificate
SRCH_INFOITEM	Search Information Items	/app/common/search/search.nl?id=-2540&e=T
SRCH_INFOITEMFORM	Search Published Forms	/app/common/search/search.nl?id=-2541&e=T
SRCH_INVENTORYNUMBER	Search Inventory Number	/app/common/search/search.nl?searchtype=InventoryNumber
SRCH_ISSUE	Search Issue	/app/common/search/search.nl?searchtype=Issue
SRCH_ITEM	Search Item	/app/common/search/search.nl?searchtype=Item
SRCH_JOB	Search Job	/app/common/search/search.nl?searchtype=Job
SRCH_LEAD	Search Leads	/app/common/search/search.nl?searchtype=Customer&Customer_STAGE=LEAD
SRCH_LOCATION	Search Location	/app/common/search/search.nl?searchtype=Location
SRCH_LOGINAUDIT	Search Login Audit Trail	/app/common/search/search.nl?searchtype=LoginAudit
SRCH_MEDIAITEM	Search Media Items	/app/common/search/search.nl?searchtype=Document
SRCH_MEMDOC	Search Memorized Transactions	/app/common/search/search.nl?searchtype=MemDoc
SRCH_PARTNER	Search Partner	/app/common/search/search.nl?searchtype=Partner
SRCH_PRESCATEGORY	Search Categories	/app/common/search/search.nl?searchtype=SiteCategory
SRCH_PRICING	Search Pricing	/app/common/search/search.nl?searchtype=Pricing
SRCH_PROJECTRESOURCE	Search Project Resources	/app/common/search/search.nl?searchtype=ProjectResource
SRCH_PROJECTTASK	Search Project Tasks	/app/common/search/search.nl?searchtype=ProjectTask
SRCH_PROJECTTASKANDCRMTASK	Search Project Tasks and CRM Tasks	/app/common/search/search.nl?searchtype=ProjectTaskAndCrmTask
SRCH_PROMOTION	Search Promotion Code	/app/common/search/search.nl?searchtype=Promotion
SRCH_PROSPECT	Search Prospects	/app/common/search/search.nl?searchtype=Customer&Customer_STAGE=PROSPECT
SRCH_QUOTA	Search Establish Quota	/app/common/search/search.nl?searchtype=Quota

Task ID	Page Label in NetSuite	URL
SRCH_REVREC	Search Revenue Recgonition Schedules	/app/common/search/search.nl?searchtype=RevRec
SRCH_REVRECOGNITIONSCHED	Search Revenue Recgonition Schedules	/app/common/search/search.nl?searchtype=RevRecognitionSched
SRCH_ROLE	Search Role	/app/common/search/search.nl?searchtype=Role
SRCH_SALESCAMPAIGN	Search Sales Campaigns	/app/common/search/search.nl?searchtype=SalesCampaign
SRCH_SALESTERRITORIES	Sales Territory Search	/app/common/search/search.nl?searchtype=SalesTerritory
SRCH_SAVEDSEARCH	All Saved Searches	/app/common/search/savedsearches.nl
SRCH_SCRIPTNOTE	Search Script Execution Log	/app/common/search/search.nl?searchtype=ScriptNote
SRCH_SHOPPINGCART	Search Shopping Cart	/app/common/search/search.nl?searchtype=ShoppingCart
SRCH_SOLUTION	Search Solution	/app/common/search/search.nl?searchtype=Solution
SRCH_SUBSIDIARY	Search Subsidiary	/app/common/search/search.nl?searchtype=Subsidiary
SRCH_SYSTEMNOTE	Search System Notes	/app/common/search/search.nl?searchtype=SystemNote
SRCH_TASK	Search Tasks	/app/common/search/search.nl?searchtype=Task
SRCH_TAXGROUP	Search Tax Group	/app/common/search/search.nl?searchtype=TaxGroup
SRCH_TAXITEM	Search Tax Code	/app/common/search/search.nl?searchtype=TaxItem
SRCH_TIME	Search Track Time	/app/common/search/search.nl?searchtype=Time
SRCH_TOPIC	Search Topic	/app/common/search/search.nl?searchtype=Topic
SRCH_TRANSACTION	Search Transactions	/app/common/search/search.nl?searchtype=Transaction
SRCH_TRAN_BINTRNFR	Search Bin Transfers	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=BinTrnfr
SRCH_TRAN_BINWKSHT	Search Bin Putaway Worksheets	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=BinWksht
SRCH_TRAN_BUILD	Search Assembly Builds	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Build
SRCH_TRAN_CARDCHRG	Search Credit Card Charges	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=CardChrg

Task ID	Page Label in NetSuite	URL
SRCH_TRAN_CASHRFND	Search Cash Refunds	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=CashRfnd
SRCH_TRAN_CASHSALE	Search Cash Sales	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=CashSale
SRCH_TRAN_CHECK	Search Checks	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Check
SRCH_TRAN_COMMISSN	Search Individual Employee Commissions	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Commissn
SRCH_TRAN_CUSTCHRG	Search Statement Charges	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=CustChrg
SRCH_TRAN_CUSTCRED	Search Credit Memos	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=CustCred
SRCH_TRAN_CUSTDEP	Search Customer Deposits	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=CustDep
SRCH_TRAN_CUSTINVC	Search Invoices	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=CustInvc
SRCH_TRAN_CUSTPYMT	Search Customer Payments	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=CustPymt
SRCH_TRAN_CUSTRFND	Search Customer Refunds	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=CustRfnd
SRCH_TRAN_DEPAPPL	Search Deposit Applications	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=DepAppl
SRCH_TRAN_DEPOSIT	Search Deposits	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Deposit
SRCH_TRAN_ESTIMATE	Search Estimates	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Estimate
SRCH_TRAN_EXPREPT	Search Expense Reports	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=ExpRept
SRCH_TRAN_FXREVAL	Search Currency Revaluations	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=FxReval
SRCH_TRAN_INVADJST	Search Inventory Adjustments	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=InvAdjst
SRCH_TRAN_INVDISTR	Search Inventory Distributions	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=InvDistr
SRCH_TRAN_INVTRNFR	Search Inventory Transfers	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=InvTrnfr
SRCH_TRAN_INVWKSHT	Search Inventory Worksheets	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=InvWksht
SRCH_TRAN_ITEMRCPT	Search Item Receipts	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=ItemRcpt
SRCH_TRAN_ITEMSHIP	Search Item Fulfillments	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=ItemShip

Task ID	Page Label in NetSuite	URL
SRCH_TRAN_JOURNAL	Search Journal Entries	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Journal
SRCH_TRAN_LIABPYMT	Search Liability Payments	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=LiabPymt
SRCH_TRAN_OPPRTNTY	Search Opportunities	/app/common/search/search.nl?searchtype=Opprtnty
SRCH_TRAN_PAYCHECK	Search Paychecks	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Paycheck
SRCH_TRAN_PURCHORD	Search Purchase Orders	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=PurchOrd
SRCH_TRAN_REVCOMM	Search Revenue Commitments	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=RevComm
SRCH_TRAN_REVCOMRV	Search View Revenue Commitment Reversals	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=RevComRv
SRCH_TRAN_RTNAUTH	Search Return Authorizations	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=RtnAuth
SRCH_TRAN_SALESORD	Search Sales Orders	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=SalesOrd
SRCH_TRAN_TAXLIAB	Search Tax Liabilities	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=TaxLiab
SRCH_TRAN_TAXPYMT	Search Tax Payments	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=TaxPymt
SRCH_TRAN_TEGPYBL	Search Issued Tegatas	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=TegPybl
SRCH_TRAN_TEGRCVBL	Search Received Tegatas	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=TegRcvbl
SRCH_TRAN_TRANSFER	Search Bank Transfers	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Transfer
SRCH_TRAN_TRNFRORD	Search Transfer Orders	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=TrnfrOrd
SRCH_TRAN_UNBUILD	Search Assembly Unbuilds	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=Unbuild
SRCH_TRAN_VENDAUTH	Search Vendor Return Authorizations	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=VendAuth
SRCH_TRAN_VENDBILL	Search Bills	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=VendBill
SRCH_TRAN_VENDCRED	Search Bill Credits	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=VendCred
SRCH_TRAN_VENDPYMT	Search Bill Payments	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=VendPymt
SRCH_TRAN_WORKORD	Search Work Orders	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=WorkOrd

Task ID	Page Label in NetSuite	URL
SRCH_TRAN_YTDADJST	Search Payroll Adjustment	/app/common/search/search.nl?searchtype=Transaction&Transaction_TYPE=YtdAdjst
SRCH_UNITSTYPE	Search Unit of Measure	/app/common/search/search.nl?searchtype=UnitsType
SRCH_USERNOTE	Search User Notes	/app/common/search/search.nl?searchtype=UserNote
SRCH_VENDOR	Search Vendor	/app/common/search/search.nl?searchtype=Vendor
SRCH_WORKCALENDAR	Search Work Calendar	/app/common/search/search.nl?searchtype=WorkCalendar
SRCH_WORKFLOW	Search Workflow	/app/common/search/search.nl?searchtype=Workflow
SRCH_WORKFLOWINSTANCE	Search Workflow Instances	/app/common/search/search.nl?searchtype=WorkflowInstance
SRCH_WORKPLACE	Search Workplace	/app/common/search/search.nl?searchtype=Workplace
SUPT_ALLOW_LOGIN	NetSuite Support Login	/app/crm/support/allowsupportlogin.nl
SUPT_BUG_BILLING	Billing Questions	"javascript:window.open(""/app/crm/support/nlcorpsupport.nl?type=billing"","" messagepopup"",""resizable=yes,scrollbars=yes,width=800,height=700"");void(0)"
SUPT_BUG_SUPPORT	Contact Support Online	"javascript:window.open(""/app/crm/support/nlcorpsupport.nl?type=support"","" messagepopup"",""resizable=yes,scrollbars=yes,width=800,height=700"");void(0)"
SUPT_BUG_SUPPORT_POPUP	Contact Support By Phone	"javascript:window.open(""/app/crm/support/contactus.nl?l=T"",""messagepopup"","" scrollbars,width=450,height=400"");void(0)"
SUPT_CENTER_ROLE	NetSuite Support Center	/app/login/dashboard.nl?id=
SUPT_GENERICHELP	Generic Help for related topics	
SUPT_GLOSSARY	Glossary	"javascript:nlPopupHelp(""UG_Glossary"", ""userGuides"")"
SUPT_GUIDES	User Guides	"javascript:nlPopupHelp(""userGuides"","" userGuides"")"
SUPT_HELP	Help	"javascript:nlPopupHelp(""help"",""help"")"
SUPT_MENUOPTIONS	Customer Support Menu Options	"javascript:window.open(""/core/help/helppdf.nl?id=309"");void(0)"
SUPT_NSCENTRAL	NetSuite Central	"javascript:window.open(""/app/site/backend/login/shoppingsecureloginredirect.nl?company=NLCORP&nscentral=T&redirect=https://central.netsuite.com/index.html""); void(0);"

Task ID	Page Label in NetSuite	URL
SUPT_POLICYFORM	Support Policies	"javascript:nlPopupHelp(""SUPPORT_POLICY"",""help"")"
SUPT_RELEASENOTES	Release Notes	"javascript:nlPopupHelp(""relNotes"",""help"")"
SUPT_SNEAKPEAK9_5	Version 9.5 Updates	"javascript:showSneakPeek(""9_5"");void(0)"
SUPT_VERSION_10_5	Version 10.6	"javascript:nlPopupHelp(""sneakPeeks"",""help"")"
TRAN_ACTIVITY	Activity	/app/crm/calendar/activity.nl
TRAN_ADDCONTENT	Add Content	/app/center/setup/addcontent.nl
TRAN_ADDSHORTCUT	Short Cuts	/core/pages/addShortcut.nl
TRAN_ALLOCATEPAYCHECKSTOJOBS	Allocate Paycheck Expenses to Jobs	/app/accounting/transactions/allocatepaycheckstojobs.nl
TRAN_APPROVAL_EXPREPT	Approve Expense Reports	/app/accounting/transactions/approval.nl?type=exprept&label=Expense%20Report
TRAN_APPROVAL_PURCHORD	Approve Purchase Requests	/app/accounting/transactions/approval.nl?type=purchord&label=Purchase%20Request
TRAN_APPROVECOMMISSN	Approve Employee Commissions	/app/accounting/transactions/approvecommissn.nl
TRAN_APPROVEPARTNERCOMMISSN	Approve Partner Commissions	/app/accounting/transactions/approvepartnercommissn.nl
TRAN_AUDIT	View Audit Trail	/app/accounting/transactions/audit.nl
TRAN_BANKRECON	Find Matching Transactions	/app/accounting/transactions/bankrecon.nl
TRAN_BANKVIEW	Online Banking Statement	/app/accounting/transactions/bankview.nl
TRAN_BAS	Business Activity Statement	/app/accounting/reports/intl/bas.nl
TRAN_BILLPAY_LOG	Job Status	/app/external/xml/upload/uploadlog.nl?displayType=BILLPAY
TRAN_BUDGET	Set Up Budgets	/app/accounting/transactions/budgets.nl
TRAN_BULKBILL_LOG	Job Status	/app/external/xml/upload/uploadlog.nl?displayType=BULKBILL
TRAN_BULKCOMMITREVENUE_LOG	Job Status	/app/external/xml/upload/uploadlog.nl?displayType=BULKCOMMITREVENUE
TRAN_BULKFULFILL_LOG	Job Status	/app/external/xml/upload/uploadlog.nl?displayType=BULKFULFILL
TRAN_CALENDARPREFERENCE	Calendar Preferences	/app/crm/calendar/calendarpreference.nl
TRAN_CAMPAIGNSETUP	Marketing Preferences	/app/setup/campaignsetup.nl
TRAN_CLEARHOLD	Manage Payment Holds	/app/accounting/transactions/salesordermanager.nl?type=clearholds
TRAN_COPY_BUDGET	Copy Budgets	/app/accounting/transactions/copybudget.nl
TRAN_CREATEAMORTIZATIONJE	Create Amortization Journal Entries	/app/accounting/transactions/createamortizationje.nl

Task ID	Page Label in NetSuite	URL
TRAN_CREATEAMORTIZATIONJE_LOG	Create Journal Entries Status	/app/accounting/transactions/createrecognitionjestatus.nl?type=AMORTIZATION
TRAN_CREATEINTERCOADJJE	Create Intercompany Adjustment Journal Entries	/app/accounting/transactions/intercoadj/createintercoadjje.nl
TRAN_CREATEINTERCOADJJE_LOG	Create Journal Entries Status	/app/accounting/transactions/intercoadj/createintercoadjjestatus.nl
TRAN_CREATEREVRECJE	Create Revenue Recognition Journal Entries	/app/accounting/transactions/createrevrecje.nl
TRAN_CREATEREVRECJE_LOG	Create Journal Entries Status	/app/accounting/transactions/createrecognitionjestatus.nl?type=REVREC
TRAN_CREATE_JOBS_FROM_ORDERS	Create Jobs From Sales Orders	/app/accounting/transactions/jobcreationmanager.nl
TRAN_CREATE_WORK_ORDERS_FOR_STOCK	Mass Create Work Orders	/app/accounting/transactions/createworkordersforstock.nl
TRAN_CUSTOMIZEITEMLIST	Customize Item List	/app/accounting/transactions/customizeitemlist.nl
TRAN_CUSTOMIZETRANLIST	Customize Transaction List	/app/accounting/transactions/customizetranlist.nl
TRAN_DEPOSITSUMMARY	Deposit Summary	/app/accounting/transactions/depositsummary.nl
TRAN_DOMAINS	Set Up Domains	/app/setup/domains.nl
TRAN_EMAILPWD	Change Email Password	/app/center/emailpwd.nl
TRAN_EMPLOYEESFA	Assign Reps	/app/common/entity/employeesfa.nl
TRAN_FINCHRG	Assess Finance Charges	/app/accounting/transactions/finchrg.nl
TRAN_FORECAST	Edit Sales Rep Forecast	/app/crm/sales/forecast.nl
TRAN_GENERATEFISCALPERIODS	Generate Fiscal Periods	/app/setup/period/generatefiscalperiods.nl
TRAN_GENERATETAXPERIODS	Generate Tax Periods	/app/setup/period/generatetaxperiods.nl
TRAN_GIFTCERTCREATEJE	Recognize Gift Certificate Income	/app/accounting/transactions/giftcertcreateje.nl
TRAN_GSTREFUND	Process GST/HST Refund	/app/accounting/transactions/gstrefund.nl
TRAN_HISTORY	History	/app/accounting/transactions/history.nl
TRAN_IMPACT	GL Impact	/app/accounting/transactions/impact.nl
TRAN_INVOICECUSTOMERS	Invoice Billable Customers	/app/accounting/transactions/invoicecustomers.nl
TRAN_ITEMSHIPPACK	Mark Orders Packed	/app/accounting/transactions/itemshipmanager.nl?type=pack
TRAN_ITEMSHIPSHIP	Mark Orders Shipped	/app/accounting/transactions/itemshipmanager.nl?type=ship

Task ID	Page Label in NetSuite	URL
TRAN_JOURNALAPPROVAL	Approve Journal Entries	/app/accounting/transactions/journalapproval.nl
TRAN_LOGINAUDIT	Login Audit	/app/setup/loginAudit.nl
TRAN_MARKVSOEDELIVERED	Mark VSOE Delivered Status	/app/accounting/transactions/revrec/markvsoedelivered.nl
TRAN_MGRFORECAST	Edit Sales Manager Forecast	/app/crm/sales/mgrforecast.nl
TRAN_NLVENDOR	Vendor Center	/app/center/nlvendor.nl.nl
TRAN_OPENBAL	Enter Opening Balances	/app/accounting/transactions/openbal.nl
TRAN_ORDERITEMS	Order Items	/app/accounting/transactions/orderitems.nl
TRAN_PAYMENTS	Payments	/app/accounting/transactions/payments.nl
TRAN_PAYROLLBATCH	Payroll Batch	/app/payroll/payrollbatch.nl
TRAN_PAYROLLRUN	Create Payroll	/app/accounting/transactions/payroll/payrollrun.nl
TRAN_PAYROLLSTATUS	View Payroll Status	/app/payroll/cdstatus.nl
TRAN_PDF_F940	Annual Federal Unemployment (940)	/app/accounting/reports/taxformfederal.nl?formtype=f940
TRAN_PDF_F941	Quarterly Federal Tax Return (941)	/app/accounting/reports/taxformfederal.nl?formtype=f941
TRAN_POSTVENDORBILLVARIANCES	Post Vendor Bill Variances	/app/accounting/transactions/vendorbillvariance/postvendorbillvariances.nl
TRAN_POSTVENDORBILLVARIANCES_STATUS	Post Vendor Bill Variances Status	/app/accounting/transactions/vendorbillvariance/postvendorbillvariancesstatus.nl
TRAN_PREVIEWW2	Form W-2 Preview	/app/accounting/reports/w2preview.nl
TRAN_PRINT	Print Checks and Forms	/app/accounting/print/print.nl
TRAN_PRINT1096	Form 1096	/app/accounting/reports/NLPrint1096s.nl?mode=frame
TRAN_PRINT1099	Form 1099-MISC	/app/accounting/reports/NLPrint1099s.nl?mode=frame
TRAN_PRINTBARCODES	Generate Barcodes	/app/accounting/print/printbarcodes.nl?printtype=null&method=print
TRAN_PRINTMAILINGLABELS	Print Mailing Labels	/app/accounting/print/printmailinglabels.nl?printtype=null&method=print
TRAN_PRINTPRICELIST	Individual Price List	/app/accounting/print/printpricelist.nl
TRAN_PRINTSTATEMENT	Individual Statement	/app/accounting/print/printstatement.nl
TRAN_PRINTW2	Form W-2	/app/accounting/reports/NLPrintW2s.nl?mode=frame
TRAN_PRINTW2AUDIT	W-2 and 1099 Audit Information	/app/accounting/reports/printw2audit.nl?mode=frame

Task ID	Page Label in NetSuite	URL
TRAN_PRINTW3	Form W-3	/app/accounting/reports/NLPrintW3s.nl?mode=frame
TRAN_PRINT_BOM	Print Bills of Materials	/app/accounting/print/printframe.nl?trantype=workord&printtype=bom&method=print
TRAN_PRINT_CASHSALE	Print Receipts	/app/accounting/print/printframe.nl?trantype=cashsale&printtype=transaction&method=print
TRAN_PRINT_CHECK	Print Checks	/app/accounting/print/printframe.nl?trantype=check&printtype=transaction&method=print
TRAN_PRINT_COMMERCIALINVOICE	Print Commerical Invoice	/app/accounting/print/commericalinvoice.nl
TRAN_PRINT_CUSTCRED	Print Credits Memos	/app/accounting/print/printframe.nl?trantype=custcred&printtype=transaction&method=print
TRAN_PRINT_CUSTINVC	Print Invoices	/app/accounting/print/printframe.nl?trantype=custinvc&printtype=transaction&method=print
TRAN_PRINT_ESTIMATE	Print Estimates	/app/accounting/print/printframe.nl?trantype=estimate&printtype=transaction&method=print
TRAN_PRINT_INTEGRATEDSHIPPINGLABEL	Print Intetgrated Shipping Labels	/app/accounting/print/printlabels.nl?printtype=integratedshippinglabel&method=print&title=Integrated Shipping Labels
TRAN_PRINT_ITEM_DETAIL_STATEMENT	Generate Item Detail Statements	/app/accounting/print/printframe.nl?trantype=&printtype=itemdetailstatement&method=print
TRAN_PRINT_ONE_ITEM_DETAIL_STATEMENT	Individual Item Detail Statement	/app/accounting/print/printitemdetailstatement.nl
TRAN_PRINT_PACKINGSLIP	Print Packing Slips	/app/accounting/print/printframe.nl?trantype=&printtype=packingslip&method=print
TRAN_PRINT_PAYCHECK	Print Paychecks	/app/accounting/print/printframe.nl?trantype=paycheck&printtype=transaction&method=print
TRAN_PRINT_PAYMENTVOUCHER	Print Payment Vouchers	/app/accounting/print/printframe.nl?trantype=vendpymt&printtype=paymentvoucher&method=print
TRAN_PRINT_PICKINGTICKET	Print Picking Tickets	/app/accounting/print/printframe.nl?trantype=salesord&printtype=pickingticket&method=print
TRAN_PRINT_PRICELIST	Generate Price Lists	/app/accounting/print/printframe.nl?trantype=&printtype=pricelist&method=print

Task ID	Page Label in NetSuite	URL
TRAN_PRINT_PURCHORD	Print Purchase Orders	/app/accounting/print/printframe.nl?trantype=purchord&printtype=transaction&method=print
TRAN_PRINT_RTNAUTH	Return Authorizations	/app/accounting/print/printform.nl?printtype=transaction&trantype=rtnauth&method=print&title=Return%20Authorizations
TRAN_PRINT_SALESORD	Print Sales Orders	/app/accounting/print/printframe.nl?trantype=salesord&printtype=transaction&method=print
TRAN_PRINT_SHIPPINGLABEL	Print Shipping Labels	/app/accounting/print/printframe.nl?trantype=&printtype=shippinglabel&method=print
TRAN_PRINT_STATEMENT	Generate Statements	/app/accounting/print/printframe.nl?trantype=&printtype=statement&method=print
TRAN_PROCESSCOMMISSN	Authorize Employee Commissions	/app/accounting/transactions/processcommissn.nl
TRAN_PROCESSORDER	Process Individual Order	/app/accounting/transactions/processorder.nl
TRAN_PROCESSPARTNERCOMMISSN	Authorize Partner Commissions	/app/accounting/transactions/processpartnercommissn.nl
TRAN_PURCHORDPROC	Bill Purchase Orders	/app/accounting/transactions/purchordermanager.nl?type=proc
TRAN_PURCHORDRECEIVE	Receive Order	/app/accounting/transactions/purchordermanager.nl?type=receive
TRAN_QUOTA	Establish Quotas	/app/crm/sales/quota.nl
TRAN_REALLOCITEMS	Reallocate Items	/app/accounting/transactions/reallocitems.nl
TRAN_RECONCILE	Reconcile Bank Statement	/app/accounting/transactions/reconcile.nl
TRAN_RECONCILE_CC	Reconcile Credit Card Statement	/app/accounting/transactions/reconcile.nl?page_type=cc
TRAN_REIMBURSEMENTS	Reimbursements	/app/accounting/transactions/reimbursements.nl
TRAN_REMINDERS	Setup Reminders	/app/center/setup/reminders.nl
TRAN_REVIEWNEGATIVEINVENTORY	Review Negative Inventory	/app/accounting/transactions/inventory/reviewnegativeinventory.nl
TRAN_REVRECCREATEJE	Revenue Recognition Schedules	/app/accounting/transactions/revreccreateje.nl
TRAN_RTNAUTHAPPRV	Approve Return Authorizations	/app/accounting/transactions/returnauthmanager.nl?type=apprv
TRAN_RTNAUTHCREDIT	Refund Returns	/app/accounting/transactions/returnauthmanager.nl?type=credit
TRAN_RTNAUTHRECEIVE	Receive Returned Order	/app/accounting/transactions/returnauthmanager.nl?type=receive

Task ID	Page Label in NetSuite	URL
TRAN_RTNAUTHREVERSEREVCOMMITMENT	Generate Revenue Commitment Reversals	/app/accounting/transactions/returnauthmanager.nl?type=reverserevcommitment
TRAN_SALESORDAPPRV	Approve Sales Orders	/app/accounting/transactions/salesordermanager.nl?type=apprv
TRAN_SALESORDCOMMITREVENUE	Generate Revenue Commitments	/app/accounting/transactions/salesordermanager.nl?type=commitrevenue
TRAN_SALESORDFULFILL	Fulfill Orders	/app/accounting/transactions/salesordermanager.nl?type=fulfill
TRAN_SALESORDPROC	Bill Sales Orders	/app/accounting/transactions/salesordermanager.nl?type=proc
TRAN_SEARCH	Search	/app/common/search/search.nl
TRAN_SHORTCUTS	Add Shortcuts	/app/center/shortcuts.nl
TRAN_SNAPSHOTCOMPOSER	Custom Snapshot Report	/app/reporting/snapshotcomposer.nl
TRAN_SNAPSHOTS	Setup Snaphots	/app/center/setup/snapshots.nl
TRAN_TAXPERIODS	Generate Tax Reporting Periods	/app/setup/period/generatetaxperiods.nl
TRAN_TIMEAPPROVAL	Approve Time	/app/accounting/transactions/timeapproval.nl
TRAN_TIMEBILL	Track Time	/app/accounting/transactions/timebill.nl
TRAN_TIMEBILL_WEEKLY	Weekly Time Sheet	/app/accounting/transactions/timebill.nl?weekly=T
TRAN_TIMECALC	Calculate Time	/core/pages/timecalc.nl
TRAN_TIMER	Timer	/core/pages/timer.nl
TRAN_TRNFRORDAPPRV	Approve Transfer Orders	/app/accounting/transactions/transferordermanager.nl?type=apprv
TRAN_USERPREFS	Set Preferences	/app/center/userprefs.nl
TRAN_VAT100	VAT 100	/app/accounting/reports/intl/vat100.nl
TRAN_VENDAUTHAPPRV	Approve Vendor Returns	/app/accounting/transactions/vendauthmanager.nl?type=apprv
TRAN_VENDAUTHCREDIT	Credit Vendor Returns	/app/accounting/transactions/vendauthmanager.nl?type=credit
TRAN_VENDAUTHRETURN	Ship Vendor Returns	/app/accounting/transactions/vendauthmanager.nl?type=return
TRAN_VENDBILLPURCHORD	New Purchase Order	/app/accounting/transactions/vendbillpurchord.nl
TRAN_VENDPYMTS	Pay Bills	/app/accounting/transactions/vendpymts.nl
TRAN_WORKORDBUILD	Build Work Orders	/app/accounting/transactions/salesordermanager.nl?type=build