## Contents

### Chapter 1  Welcome

- In this Guide .................................................. 8
- Supported Operations ........................................ 8
- Unsupported Features .......................................... 10
- Understanding Web Services Governance .................. 10
  - Understanding Usage Reporting ............................. 10
  - Understanding Record Limiting ............................ 11
  - Understanding Request Limiting ........................... 12
  - Understanding Governance Errors ........................ 12
- Vocabulary ...................................................... 13

### Chapter 2  Getting Started

- Understanding Web Services ................................ 14
- Quick Start ...................................................... 15
  - Enabling the Web Services Feature .......................... 15
  - Building an Application with Microsoft .NET .............. 15
  - Building an Application with Java using Apache Axis ... 17
  - Running the Sample Application .............................. 19
  - Using Web Services with PHP ................................ 19
  - Using SOAPScope .............................................. 19
- Setting Your Web Services Preferences ..................... 21
  - Setting Company-Wide Preferences .......................... 21
  - Setting Preferences at the Request Level ................... 24
  - Displaying Internal IDs ...................................... 25
- General Concepts .............................................. 25
  - Understanding Features ...................................... 25
  - Working with Roles and Permissions .......................... 26
  - Using Web Services Specific Forms .......................... 28
  - Effects of Account Configuration on Web Services .......... 28
  - Working with Lists ............................................ 29
  - Working with Hidden Fields ................................... 31

### Chapter 3  Platform Features

- Understanding the WSDL and XSD Structure ................ 32
  - Versioning ..................................................... 33
  - NetSuite Messaging XSD Files ................................ 33
  - NetSuite Business Records XSD Files ....................... 33
  - System Constants XSD Files .................................. 34
  - Example ......................................................... 35
- Security .......................................................... 36
  - Authentication ................................................ 36

SuiteTalk Platform - Version 2.5.0
Chapter 4 Types

Built-in Types .......................................................... 51
NetSuite Complex Types ............................................. 52
  Passport .............................................................. 52
  Record ............................................................... 53
  RecordList .......................................................... 54
  BaseRef ............................................................... 54
  ListOrRecordRef ..................................................... 54
  Status ................................................................. 55
  StatusDetail .......................................................... 55
  NullField ............................................................... 56
  ReadResponse .......................................................... 56
  ListReadResponse .................................................... 56
  ListWriteResponse .................................................... 56
  WriteResponse ........................................................ 56

Custom Field Types .................................................... 57
  CustomFieldRef ....................................................... 57
  IntCustomFieldRef ..................................................... 57
  DoubleCustomFieldRef ................................................. 58
  BooleanCustomFieldRef .............................................. 58
  StringCustomFieldRef .............................................. 58
  DateCustomFieldRef ................................................. 58
  SelectCustomFieldRef ................................................ 58
  MultiSelectCustomFieldRef ........................................... 58
  CustomFieldList ....................................................... 59

Session Management ...................................................... 36
Authorization ............................................................ 38
Encryption ............................................................... 38

Using Internal IDs, External IDs, and References .................. 38
  Understanding ExternalIDs ........................................... 39
  List of Operations and Internal ID Requirements .................... 39
  Example Internal ID Usage ............................................. 40

Processing Requests Asynchronously .................................. 44
  checkAsyncStatus ...................................................... 45
  getAsyncResult ......................................................... 46
  Monitoring Asynchronous Jobs in the UI ............................ 46

Web Services to UI Single Login ....................................... 46

Handling Errors ........................................................ 48
  Warnings ............................................................... 48
  Errors ................................................................. 49
  Faults ................................................................. 49

Understanding ExternalIDs ............................................... 39

Encryption ............................................................... 38
Authorization ............................................................ 38
Session Management ...................................................... 36

SuiteTalk Platform - Version 2.5.0
Chapter 5 Operations

login .............................................................. 70
  Request ....................................................... 71
  Response ...................................................... 71
  Faults .......................................................... 72
  Sample Code .................................................. 72
mapSso .......................................................... 75
  Request ....................................................... 75
  Response ...................................................... 76
  Faults .......................................................... 76
  Sample Code: SOAP ........................................... 76
  Sample Code: Java ............................................ 77
logout ............................................................ 77
  Request ....................................................... 77
  Response ...................................................... 77
  Faults .......................................................... 77
  Sample Code .................................................. 77
addList .......................................................... 79
  Request ....................................................... 79
  Response ...................................................... 79
  Faults .......................................................... 79
  Sample Code .................................................. 80
add .............................................................. 80
  Request ....................................................... 80
  Response ...................................................... 81
  Faults .......................................................... 81
  Sample Code .................................................. 81
updateList ...................................................... 85
  Request ....................................................... 86
  Response ...................................................... 86
  Faults .......................................................... 86
  Sample Code .................................................. 86
update ........................................................... 89
  Request ....................................................... 90
  Response ...................................................... 90
  Faults .......................................................... 90
  Sample Code .................................................. 90
  Updating Record Lists ...................................... 93
deleteList ....................................................... 93
  Request ....................................................... 93
  Response ...................................................... 93
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>delete</td>
<td>96</td>
</tr>
<tr>
<td>Request</td>
<td>96</td>
</tr>
<tr>
<td>Response</td>
<td>96</td>
</tr>
<tr>
<td>Faults</td>
<td>97</td>
</tr>
<tr>
<td>Sample Code</td>
<td>97</td>
</tr>
<tr>
<td>search</td>
<td>98</td>
</tr>
<tr>
<td>Working with Searches and Joined Searches</td>
<td>99</td>
</tr>
<tr>
<td>Returning Associated Lists</td>
<td>101</td>
</tr>
<tr>
<td>Search Preferences</td>
<td>102</td>
</tr>
<tr>
<td>Filtering Lists that Contain Null Values</td>
<td>103</td>
</tr>
<tr>
<td>Request</td>
<td>104</td>
</tr>
<tr>
<td>Response</td>
<td>104</td>
</tr>
<tr>
<td>Faults</td>
<td>104</td>
</tr>
<tr>
<td>Sample Code</td>
<td>105</td>
</tr>
<tr>
<td>searchMore</td>
<td>109</td>
</tr>
<tr>
<td>Request</td>
<td>110</td>
</tr>
<tr>
<td>Response</td>
<td>110</td>
</tr>
<tr>
<td>Faults</td>
<td>111</td>
</tr>
<tr>
<td>Sample Code</td>
<td>111</td>
</tr>
<tr>
<td>searchNext</td>
<td>113</td>
</tr>
<tr>
<td>Request</td>
<td>113</td>
</tr>
<tr>
<td>Response</td>
<td>113</td>
</tr>
<tr>
<td>Faults</td>
<td>114</td>
</tr>
<tr>
<td>Sample Code</td>
<td>114</td>
</tr>
<tr>
<td>getList</td>
<td>115</td>
</tr>
<tr>
<td>Request</td>
<td>115</td>
</tr>
<tr>
<td>Response</td>
<td>115</td>
</tr>
<tr>
<td>Faults</td>
<td>115</td>
</tr>
<tr>
<td>Sample Code</td>
<td>116</td>
</tr>
<tr>
<td>get</td>
<td>117</td>
</tr>
<tr>
<td>Request</td>
<td>117</td>
</tr>
<tr>
<td>Response</td>
<td>117</td>
</tr>
<tr>
<td>Faults</td>
<td>117</td>
</tr>
<tr>
<td>Sample Code</td>
<td>118</td>
</tr>
<tr>
<td>getAll</td>
<td>120</td>
</tr>
<tr>
<td>Request</td>
<td>120</td>
</tr>
<tr>
<td>Response</td>
<td>120</td>
</tr>
<tr>
<td>Faults</td>
<td>121</td>
</tr>
<tr>
<td>Sample Code</td>
<td>121</td>
</tr>
<tr>
<td>getSelectValue</td>
<td>125</td>
</tr>
<tr>
<td>Request</td>
<td>125</td>
</tr>
<tr>
<td>Response</td>
<td>126</td>
</tr>
<tr>
<td>Faults</td>
<td>126</td>
</tr>
<tr>
<td>Sample Code</td>
<td>126</td>
</tr>
<tr>
<td>GetSelectValue Types</td>
<td>127</td>
</tr>
</tbody>
</table>
Handling Errors

Soap Faults for each Operation .................................................. 136
Soap Fault Status Codes .............................................................. 138
Error Status Codes ..................................................................... 139
Warning Status Codes ................................................................. 159

Task IDs

FAQs
Chapter 1  Welcome

Welcome to SuiteTalk. The SuiteTalk Platform provides programmatic access to your Netsuite data and business processes through an XML-based application programming interface (API). This guide provides an overview of the SuiteTalk Platform with detailed requirements and example code for each of the core Web services operations. For detailed information on each record available, refer to the SuiteTalk Records Guide.

In this Guide

This manual contains the following sections:

Chapter 1 "Welcome": provides an overview of this guide, what operations are supported, what features are currently unsupported, and an overview of Web services governance.

Chapter 2 "Getting Started": provides an overview of Web services and how to quickly get your web services development environment set up to create your first service. It also describes how to modify the default Web services preferences and provides some general information useful when working with many different operations or record types.

Chapter 3 "Platform Features": describes basic features of the NetSuite Web Services Platform including the WSDL and XSD structure, session management and error handling.

Chapter 4 "Types": describes the various types available in the SuiteTalk Platform.

Chapter 5 "Operations": describes each operation that can be performed through web services and provides SOAP, C# and Java code samples.

Handling Errors: provides tables of possible SOAP faults, fault status codes and error status codes that can be thrown for a given request.

FAQs: provides a list of common questions with possible solutions.

Supported Operations

In this version of SuiteTalk Platform the following user actions or operations are supported:

• 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 an account number. All other Web Services operations require a successful login. Once successfully completed, a login creates a session that allows subsequent operations to be performed without having to login again until the session expires or the logout operation is invoked. For more information on how sessions are controlled, refer to “Session Management” on page 36.

• mapSso: Use to automate the mapping between external applications credentials and NetSuite’s credentials for a user.
- **logout:** Use to logout from the system. The logout operation invalidates the current session.

- **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.

- **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 the internal ID and record type.

- **delete / deleteList:** Use to delete one or more records in the system. The records to be deleted are identified by the internal ID and record type.

- **get / getList:** Use to query the system for one or more records. You must provide the internal ID and record type for each query item.

- **getAll:** Use to return the values of user defined lists in the system that are categorized as Other Lists.

- **search / searchNext / searchMore:** Use to search for a set of records based on specific search criteria. This operation has most of the features of the Advanced Search feature within the NetSuite UI. It also supports pagination, so that large result sets can be retrieved in smaller sets. For more information on how to control pagination, refer to “Setting Your Web Services Preferences” on page 21.

- **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.

- **getCustomization:** Use to dynamically retrieve and manage the metadata for Custom Fields, Lists, and Record Types

- **getItemAvailability:** Use to retrieve the inventory availability for a given list of items.

- **attachTo / attachedRecord:** Use to attach a contact to a customer record.

- **changePasswordOrEmail:** Use to change a user's email or password.

- **getDeleted:** Use to retrieve a list of deleted records of a given type during a specified period.

- **initialize:** Use to emulate the UI workflow by prepopulating fields on transaction line items with values from a related record.

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, deleteList and getList) 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.
Unsupported Features

The current version of SuiteTalk does not support the following features. Records affected by these features may not behave as expected when using web services applications to define them.

- **ADVANCEDJOBS**: When enabled, detailed information about jobs can be tracked and jobs are managed separately from customer records.
- **GIFTCERTIFICATES**: When enabled, gift certificate items of set or variable amounts can be sold to customers.
- **PARTNERCOMMISSIONS**: When enabled, schedules that calculate commission for partners and pay commission when it is earned can be created.
- **REVREC**: When enabled, revenue from your sales can be amortized.
- **VSOE**: When enabled, revenue can be deferred on products with multiple elements using Vendor Specific Objective Evidence (VSOE) capabilities.
- **ISSUEDB**: When enabled, the Issue Management database can be used.

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.

WS governance is made up of two primary areas:

- Record Limiting
- Request Limiting

Understanding Usage Reporting

All Web services usage is tracked and categorized based on a number of factors, including:

- Type of operation (e.g. add, addList, delete, etc.)
- Number of records processed in an operation
- Who performed the operation
- Time the operation was performed

You can monitor Web services usage at any time by accessing the Web Services Integration reports. These reports also provide detailed information such as the type of operations performed, the records involved, the users submitting the requests and performance statistics for each request.
**Important:** Integration Reports permission is required to view these reports.

To access these reports, go to Reports > Integration and select any of the following:

- Integration Usage Detail By Job
- Integration Usage Summary By Job
- Integration Usage Summary By Record

Each report can also 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 additional details, refer to the online help.

**Understanding Record Limiting**

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

Unless your License Agreement specifies otherwise, the record limits are as follows:

### Synchronous Operations

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

#### Peak Hours

<table>
<thead>
<tr>
<th>Operation</th>
<th>Record Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>100</td>
</tr>
<tr>
<td>Update</td>
<td>50</td>
</tr>
<tr>
<td>Delete</td>
<td>100</td>
</tr>
<tr>
<td>Search Page Size</td>
<td>500</td>
</tr>
</tbody>
</table>

#### Off-Peak Hours

<table>
<thead>
<tr>
<th>Operation</th>
<th>Record Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>200</td>
</tr>
<tr>
<td>Update</td>
<td>100</td>
</tr>
<tr>
<td>Delete</td>
<td>200</td>
</tr>
<tr>
<td>Search Page Size</td>
<td>1000</td>
</tr>
</tbody>
</table>

### Asynchronous Operations

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

Request limiting refers to the restrictions on the number of concurrent requests that can be made. The number of concurrent requests allowed varies depending on request size and time of day (peak or off-peak hours).

Request Types

Requests are classified as Long, Medium or Short depending on the number of records submitted in the request, as follows:

<table>
<thead>
<tr>
<th>Request Type</th>
<th># of Records (Add/Delete)</th>
<th># of Records (Update)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long</td>
<td>100+</td>
<td>50+</td>
</tr>
<tr>
<td>Medium</td>
<td>4 - 100</td>
<td>4 - 50</td>
</tr>
<tr>
<td>Short</td>
<td>&lt; 4</td>
<td>&lt; 4</td>
</tr>
</tbody>
</table>

Request limiting is also governed by your License agreement, but in general, no more than 1 long, 2 medium or 3 short requests can occur concurrently on a single account.

Understanding Governance Errors

The following exceptions can be thrown as a result of other Governance violations.

- **ExceededRecordCountFault**: fault thrown if a request exceeds the allowed Record count (see Understanding Record Limiting)
- **ExceededRequestLimitFault**: fault thrown if the allowed number of concurrent requests is exceeded (see Understanding Request Limiting)
- **MAX_PAGE_SIZE_EXCEEDED**: warning code generated if a request is made whose result set returned will be greater than the Search Page Size. The request is paginated based on the Search Page Size defined by the preference settings.

Note: The above warning codes may actually produce errors depending on the preference settings as described in “Setting Your Web Services Preferences” on page 21.

For more information on exceptions, refer to “Handling Errors” on page 48.
Vocabulary

Following are some terms you should be familiar with before implementing SuiteTalk integration technology.

- **Type**: A type is an element defined in the XML Schema.
- **Record Type**: A NetSuite business record type, such as customer, event, or custom record, that has an entry form in the user interface (UI).
- **Search Record Type**: A type that encapsulates the available search criteria for a given NetSuite business record type.
- **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.
- **Read operations**: A Web services operation that retrieves data from a NetSuite account. These include get, getList, getAll, search, searchNext and searchMore operations.
Chapter 2  Getting Started

This section provides the basic information you need to get started using the SuiteTalk Platform and includes the following sections:

- **Understanding Web Services**: provides a basic description of Web services.
- **Quick Start**: provides step-by-step instructions on how to setup your web services environment and start building applications to interface with the SuiteTalk Platform. Also includes a section that describes how to run a sample application provided by NetSuite.
- **Setting Your Web Services Preferences**: describes how to adjust preferences specific to the SuiteTalk Platform.
- **General Concepts**: provides information about the NetSuite data model that is useful to consider when building your own applications to interface with the SuiteTalk Platform.

Before building a web services application to interface with the SuiteTalk Platform, the Web Services feature must first be enabled in the user interface (UI). To do this, go to Setup > Company > Enable Features > General. Check the Web Services box and click Save. Once you have enabled the feature, you can set your Web Services preferences (see “Setting Your Web Services Preferences” on page 21).

**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.

Quick Start

This section provides details on how to use Microsoft .NET or Java to build an application that uses the NetSuite Web services and how to run the sample application provided 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. In the General tab, enable Web Services.
3. Click Save.

Building an Application with Microsoft .NET

This section provides details on how to use the Microsoft .NET platform to build an application that uses NetSuite Web services.

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 version 1.1. 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 — 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 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:
   a. Open Microsoft Visual Studio .NET.
   b. Create a new project by selecting the appropriate language and an application type.
c. Once the project has been created, select the Add Web Reference option from the Project menu.

d. When prompted, enter the NetSuite WSDL URL and then click Go.
https://webservices.netsuite.com/wsdl/v2_5_0/netsuite.wsdl
Visual Studio inspects the WSDL and displays a summary of the available operations.

e. Click Add Reference to generate the classes.
Once complete, you can view these classes in the Solution Explorer by expanding the Web Reference icon.
The generated proxy classes are available in a file named Reference.cs. In order to view this file, you may first need to enable the Show All Files option under the Project menu.

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

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

b. Open a command prompt, and type the following to generate the proxy classes:
wsdl /language:<language>cs <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/v2_5_0/netsuite.wsdl
A C# file called NetSuiteService.cs is generated.

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

d. 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#.

a. Instantiate the NetSuite service.
NetSuiteService service = new NetSuiteService();

b. Enable support for multiple cookie management.
service.CookieContainer = new CookieContainer();

c. Create a valid session by populating the Passport object.
invoking 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 = "foobar1";
Status status = service.login( passport ).status;

d. 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 );

e. 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 version 1.1 and higher to build an application that uses the NetSuite Web services.

**Note:** The Java based sample application available on the beta portal provides an in-depth look on how to build an application using Apache Axis and NetSuite Web services.

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.
   a. Download and install Apache Axis, version 1.1 or higher, from http://ws.apache.org/axis/.
   b. Download and install the Apache Axis patch for cookie management from NetSuite: http://www.netsuite.com/portal/partners/netflex/webservices.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.
   c. 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. Please 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/v2_5_0/netsuite.wsdl
   ```

   **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.
   a. Locate the NetSuite service.
      ```
      NetSuiteServiceLocator service = new NetSuiteServiceLocator();
      ```
   b. Enable support for multiple cookie management.
      ```
      service.setMaintainSession( true );
      ```
   c. Get the NetSuite port.
      ```
      NetSuitePortType port = service.getNetSuitePort();
      ```
   d. Create a valid session by populating the Passport object and then invoking the login operation.
      ```
      passport.setEmail( "username@netsuite.com" );
      passport.setPassword( "foobar1" );
      role.setId( "3" );
      passport.setRole( role );
      ```
passport.setAccount( "TSTDRV96" );
Status status = port.login( passport ).getStatus();

1. 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 );

2. Logout to invalidate the current session
   port.logout();

Running the Sample Application

A sample C# based application is available on the NetSuite Web services portal. This is a functional command line web services application that provides the ability to run several operations on a customer record. Before creating your own applications, it is recommended that you run this application and familiarize yourself with the associated code.

Using Web Services with PHP

**Important:** NetSuite has not tested using NetSuite Web Services with PHP. The following information is to provide general guidelines only.

When using NetSuite Web services with PHP, you can do one of the following:

- Call Web Services Directly through PHP.
  This is the preferred option. PHP itself does not directly support SOAP based Web services. However, this functionality is available through 3rd party libraries such as the following:
  - PEAR::SOAP: [http://pear.php.net/package/SOAP](http://pear.php.net/package/SOAP)
    For an overview of how to create a SOAP client in PHP using one of the above libraries, refer to the following article:
    [http://www.onlamp.com/lpt/a/3968](http://www.onlamp.com/lpt/a/3968)

- Call Web Services Using Java through a PHP/Java bridge
  The PHP/Java bridge is a PHP module which connects the PHP object system with the Java object system. The advantage to using this method is that the Java code that encapsulates the Web services calls has been tested previously by NetSuite.
  More information regarding a PHP/Java bridge, refer to the following article:

Using SOAPScope

Mindreef’s SOAPScope utility allows you to capture and analyze SOAP request and response messages associated with an operation call. To use SOAPScope with a NetSuite Web services client, follow the instructions in this section to set up SOAPScope’s SSL forwarding feature.
Mindreef’s SOAPscope 4.1’s SSL forwarding feature allows you to use SOAPscope to capture SOAP traffic over SSL/TLS/HTTPS, logging the messages in cleartext while still using encryption between the client and server.

**Note:** You can also use SOAPScope to manually invoke requests for a specific operation. When invoking requests from within SOAPScope, remember that you must first submit the login operation just as with any other web services application. Also, to deselect all optional elements for a record, hit **CTRL/Click** on any element to deselect it’s siblings.

**To set up SOAPScope:**

1. **Install SOAPscope 4.1**
   
   You can obtain a trial version of SOAPscope at:
   

2. **Configure SOAPScope.**
   
   To use the SSL Forwarding feature, you must configure SOAPscope to listen for SSL traffic on a specified port. When your client connects to that port, SOAPscope decrypts the data, logs it, re-encrypts it and forwards the data to and from the Web service. SOAPscope uses a self-signed SSL certificate to validate the forwarding socket. However, most client applications treat a self-signed certificate as untrusted, because it has not been signed by a trusted Certificate Authority, and will refuse the connection. Therefore your client app may need to be modified slightly to allow it to connect.

**Modifying a .NET/C# client application:**

- Add the following class to your code:

  ```csharp
  class OverrideCertificatePolicy : ICertificatePolicy
  {
      public bool CheckValidationResult(ServicePoint srvPoint,
          WebRequest request, int certificateProblem)
      {
          return true;
      }
  }
  
  The class **OverrideCertificatePolicy** implements the **ICertificatePolicy** interface, which defines how SSL certificates are to be validated. This implementation of CheckValidationResult unconditionally validates any server certificate that is presented.

- Before you instantiate the service object in your code, add the following line:

  ```csharp
  
  This creates an instance of this class and assigns it as the global certificate policy for the application.
  ```

**Modifying an Axis client application:**

Add the following line of code in your client before the code that instantiates the service object.

```java
System.setProperty("axis.socketSecureFactory",
    "org.apache.axis.components.net.SunFakeTrustSocketFactory");
```

This disables certificate checking by the Axis client.
3. Change your SOAP endpoint in the client code to point to port 8443 on your local machine which allows SOAPscope to intercept the message and capture it, and then forward to the original destination. So the client code should bind to the following endpoint (where _2_5 is the namespace version):

https://localhost:8443/services/NetSuitePort_2_5

The following files will need to be modified:

- For Microsoft .NET: Reference.cs
- For Java: com.netsuite.webservices.platform.NetSuiteServiceLocator.java

4. Use ssconfig.exe to tell SOAPscope to listen for HTTPS on port 8443, and forward messages to the original destination host and port. In order to do this, open up a command prompt and change directories to the SOAPscope bin directory that contains the ssconfig.exe file (should be in C:\Program Files\Mindreef\SOAPscope 4.1\bin). Type the following:

   ssconfig -s 8443,webservices.netsuite.com:443

5. Restart SOAPscope.

6. Compile your Web Services client and run.

### Setting Your Web Services Preferences

Before utilizing Web services with your NetSuite account, ensure that the appropriate preferences are enabled as defined in the following sections. These preferences control how certain services are executed.

#### 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 1000. This value must be greater than 5 and less than 1000. The page size entered here can be overridden at the individual request level. However, the following two exceptions can be thrown:
  - **MAX_PAGE_SIZE_EXCEEDED**: the page size is set to greater than the record size limit and the actual request includes more than the maximum allowed records. This is normally a warning and the response is simply limited to the record size limit but if the **Treat Warnings as Errors** preference is also enabled, the request fails.
  - **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.

- **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 “Handling Errors” on page 48.

- **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 Preferences at the Request Level” on page 24 for more details on setting request level preferences):

  ```xml
  <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 Custom Code**: When enabled, custom code is not run during a Web services operation. This may be important in cases where custom code is dependent on user interaction or slaving characteristics not available during a Web services call. The default setting is **TRUE**.

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

  **Important**: Use caution when disabling custom code when your forms have associated custom code since this directly impacts the execution of client side business logic and may result in undesirable outcome.

### Setting Default Behavior

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

- **Record fields can be automatically populated with default values.**

- **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.**

- **Records can be populated with a calculated value depending on the values set in a particular field.**
When using Web Services, however, 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.

- **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.

**Important**: You can NOT 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**

For 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. Therefore, in a Web services implementation, if the Probability field has already been adjusted based on information unknown to the NetSuite System, it may be undesirable to have the field automatically populated with the conditional default value.

**Note**: Endpoints prior to version 2.0.0 also included 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 from an earlier endpoint, you must adjust your code accordingly since this preference is no longer available.
Setting Preferences at the Request Level

In addition to setting preferences to be used for all requests submitted to NetSuite Web services, 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. If not used correctly, an error is not thrown but your settings will be ignored.
- Create the Preference Object: contains the elements you are allowed to set. Elements include:
  - `warningAsError`
  - `useConditionalDefaultsOnAdd`
  - `useConditionalDefaultsOnUpdate`
  - `disableMandatoryCustomFieldValidation`
  - `disableSystemNotesForCustomFields`
- Set the header (Java Only)

Sample Code

**SOAP Request**

```xml
<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>
```

**Java**

In this example, a SearchPreference Object is set.

```java
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.setObjectName(searchPrefs);
stub.setHeader(searchPrefHeader);
```
C#  

In this example, a SearchPreference object is set as well as a warnAsError object.

```csharp
// 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 );
```

**Displaying Internal IDs**

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 and then enable *Show internal IDs*. When 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).

When working with custom fields, the custom code feature must also be enabled to view Internal IDs for custom fields.

**General Concepts**

**Understanding Features**

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 which 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 Web services feature:

1. As administrator, click Setup > Company > Enable Features.
2. In the General tab, enable Web Services.
3. Click Save.

**Working with Roles and Permissions**

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.

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.

**Customer and Vendor Center Roles**

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

**Note:** When creating a custom Customer Center or Vendor Center, you do NOT need to define a Web services permission for that role. The ability to define permissions associated with the Setup tab is unavailable to these roles but the Web services permission is automatically assigned.

**Default Web Services Roles**

The Web services role used to specify the permissions granted to the logged in user is determined as follows:

- First, any role specified in the Passport of the request is used. The role defined here must be a valid role contained in the Employee record of the given user.
- 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 default roles have the Web services permission by default when the feature is enabled, but custom roles have to be explicitly set to have Web services role.

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

1. Click Setup > Integration > Web Services.
2. Select the desired user from the Name drop-down list.
3. Select the default role to use for Web services requests for this user.
   Note: The internal ID for the selected role automatically populates the ID field.
4. Click Add.

Web Services Permissions

- Regardless of which role is specified in the request or is defaulted, in order for the user to be granted access to the system via Web Services, the Web Services permission must be specified in the role.
- By default, all standard roles have the Web Services permission. You must customize the role and remove the permission if you wish to prevent the role from being used for Web Services.
- Role permissions are set through the UI in Setup > Users/Roles > Manage Roles. On the Setup subtab of the Permissions tab, a Web services permission can be assigned to the desired roles with the desired level of access for Web service requests for the selected role.

IDs Associated with Roles

The following table lists the standard NetSuite roles and the associated internal ID values.

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).

<table>
<thead>
<tr>
<th>ID</th>
<th>Is Settable</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Y</td>
<td>Accountant</td>
</tr>
<tr>
<td>2</td>
<td>Y</td>
<td>Accountant (Reviewer)</td>
</tr>
<tr>
<td>3</td>
<td>Y</td>
<td>Administrator</td>
</tr>
<tr>
<td>4</td>
<td>Y</td>
<td>A/P Clerk</td>
</tr>
<tr>
<td>5</td>
<td>Y</td>
<td>A/R Clerk</td>
</tr>
<tr>
<td>6</td>
<td>Y</td>
<td>Bookkeeper</td>
</tr>
<tr>
<td>7</td>
<td>Y</td>
<td>CEO (Hands Off)</td>
</tr>
<tr>
<td>8</td>
<td>Y</td>
<td>CEO</td>
</tr>
<tr>
<td>9</td>
<td>Y</td>
<td>Sales Manager</td>
</tr>
<tr>
<td>10</td>
<td>Y</td>
<td>Sales Person</td>
</tr>
<tr>
<td>11</td>
<td>Y</td>
<td>Store Manager</td>
</tr>
</tbody>
</table>
Using Web Services Specific Forms

In order to ensure that your applications are generic and stable, we recommend that you use custom Web Services specific forms. Use the custom Web services form for Web services requests by specifying the form in the customForm field. This will alleviate problems associated with customer specific customization to forms used in the UI which may break your Web services application code.

**Note:** If a form is specified in the WS request then that form is used for validation. If a form is NOT specified then the default preferred form for that record or transaction 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.

Effects of Account Configuration on Web Services

When customizaton or feature enabling occurs in NetSuite, it is possible that required fields have been added to a form or deleted. For each Web service request, fields available for the specified record are validated against the data being submitted and errors 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.

**Working with Lists**

Many NetSuite record types contain sublists of items. When working with these sublists, the method for searching on, updating, or replacing items varies depending on the type of list. There are two types of lists, **Keyed** or **Non-keyed**.

*Note:* When searching records, by default sublists are NOT returned in the result set. You can change this behavior by setting the search preference for the request. For details, see the description for the search operation in "search" on page 98.

**Keyed Lists**

These lists have a **line** field (typically on transaction lists). Individual lines in the list can be searched for by referencing the line value. For example, the OpportunityItemList on the Opportunity record is a keyed list.

The value of the replaceAll attribute on **Keyed** sublists determines the behaviour of your request for these lists.

- replaceAll = TRUE: When set to TRUE, the existing list is **replaced** with the line items submitted in your web services request. If you specify a value for the **line** field on a specific line item, then the list is replaced with the list submitted but the line referenced specifically is updated with the additional information.
- replaceAll = FALSE: When set to FALSE, line items in your web services request are **added** to the existing list. If you specify a value for the **line** field on a specific line item, then that line is updated in the existing list.

*Note:* The line field simply references the existing lines numerically, starting at one.

The default value for replaceAll is TRUE.

**Example**

For example, suppose you want 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 field. In the following code, 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_2.5.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 internalId="380" type="inventoryItem" />
    <s0:quantity>1.0</s0:quantity>
    <s0:amount>20.0</s0:amount>
```

Non-keyed Lists

These lists have no line field. The line items are recorded in flat lists. For example, salesTeamList or contactList on the Customer record are non-keyed lists.

Since these sublists are NOT keyed, you can NOT update a specific line in the list. Instead you must interact with the list as a whole. In this way an update operation is similar to the add operation with respect to sublists.

**Important:** For Non-keyed lists, the replaceAll attribute is ignored and behaves as if it were TRUE for all requests.

Example

For example, suppose you want to modify the addressbookList associated with a Customer record.

- To update a single address in the list, retrieve the entire list of addresses, change a single address as desired and then resubmit the entire list.
- To replace the addressbookList with a subset of addresses, retrieve the entire list, and then resubmit the subset of addresses. The original addressbookList is purged and replaced by the new list containing the subset of addresses.
- To delete the addressbookList, submit an empty list.

**Note:** When adding or updating an addressbookList entry ensure that you do NOT set the internalId field as this is read-only.

Example Code

The addressbookList item refers to the ContactAddressbookList type:

```xml
<xs:complexType name="Contact">
  <xs:complexContent>
    <xs:extension base="platformCore:Record">
      <xs:sequence>
        <xs:element name="entityId" type="xsd:string" minOccurs="0"/>
        ...
        <xs:element name="addressbookList" type="listRel:ContactAddressbookList" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="id" type="xsd:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```
Working with Hidden Fields

The Web services interface includes fields that are not visible in the NetSuite UI. These fields are hidden in the UI and are used primarily for storing system-generated information such as dates. For example, each time a contact record is created, a hidden datecreated or createdDate field is populated with a system generated timestamp.

Hidden fields have read and write permissions and can be populated with values submitted via Web services. If a hidden field is not explicitly populated by a Web service, it is automatically populated with the system generated data as it would be through the UI.

When submitting a request, if a hidden field is available in the schema but not available for the specified form, then whether the field is returned or not depends on the type of field:

- **Body Fields**: all fields listed in the schema are returned regardless of whether they are available on the chosen form.
- **List Machine Fields**: only fields available in the schema and on the chosen form are returned.

**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 can NOT be overridden.
Chapter 3  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 Platform and includes the following sections:

- **Understanding the WSDL and XSD Structure**: provides an overview of the WSDL and XSD files with information on where each record type is defined.
- **Security**: provides information on the NetSuite security model.
- **Using Internal IDs, External IDs, and References**: describes IDs and references and how they are used.
- **Processing Requests Asynchronously**: describes how to process requests synchronously versus asynchronously.
- **Web Services to UI Single Login**: describes how to submit an https POST for single login between web services applications and the NetSuite UI.
- **Handling Errors**: describes the three different types of exceptions that can be thrown during a web service call — warnings, errors, and faults.

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.

**Understanding the WSDL and XSD Structure**

The SuiteTalk Platform has a single WSDL file that describes all of the supported operations and messages. You can view that file here:

https://webservices.netsuite.com/wsdl/v2_5_0/netsuite.wsdl

(where v2_5_0 is the version of the wsdl)

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 tables below show the organization of the XSD files.
Versioning

The NetSuite Web Services platform defines versioning for the WSDL, the location of schemas, the namespaces, and end point as follows:

WSDL: https://webservices.netsuite.com/wsd1/v2_5_0/netsuite.wsdl
<xsd:import namespace="urn:core_2_5.platform.webservices.netsuite.com"
  schemaLocation="https://webservices.netsuite.com/xsd/platform/v2_5_0/core.xsd"/>
<port name="NetSuitePort" binding="tns:NetSuiteBinding">
  <soap:address location="https://webservices.netsuite.com/services
NetSuitePort_2_5" />
</port>
(where v2_5_0 and _2_5 reflect the current version)

The NetSuite Web services platform maintains backwards compatibility wherever possible. Additions such new Operations or Record types, result in a minor revision but the namespace will not be incremented. New minor revisions support all prior functions.

However, when backward incompatible changes occur, such as adding a new required field, changing the structure of a data type or removing an operation or element, the version of the namespace is incremented and your applications must be recompiled to reflect the new version. There is no guarantee that a major new release will support all previous functions.

In the event that an unsupported version is sent in a Web services request, an InvalidVersionFault is returned.

NetSuite Messaging XSD Files

These files provide descriptions for the base web services functions used by all operations.

<table>
<thead>
<tr>
<th>URL</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/platform/v2_5_0/core.xsd">https://webservices.netsuite.com/xsd/platform/v2_5_0/core.xsd</a></td>
<td>platformCore</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/platform/v2_5_0/messages.xsd">https://webservices.netsuite.com/xsd/platform/v2_5_0/messages.xsd</a></td>
<td>platformMsgs</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/platform/v2_5_0/faults.xsd">https://webservices.netsuite.com/xsd/platform/v2_5_0/faults.xsd</a></td>
<td>platformFaults</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/platform/v2_5_0/common.xsd">https://webservices.netsuite.com/xsd/platform/v2_5_0/common.xsd</a></td>
<td>platformCore</td>
</tr>
</tbody>
</table>

NetSuite Business Records XSD Files

These files provide descriptions for each record type in NetSuite.

<table>
<thead>
<tr>
<th>URL</th>
<th>Alias</th>
<th>Records</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/activities/v2_5_0/scheduling.xsd">https://webservices.netsuite.com/xsd/activities/v2_5_0/scheduling.xsd</a></td>
<td>actSched</td>
<td>Events, Tasks, Phone Calls</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/general/v2_5_0/communication.xsd">https://webservices.netsuite.com/xsd/general/v2_5_0/communication.xsd</a></td>
<td>generalCom</td>
<td>Messages, Notes</td>
</tr>
<tr>
<td>URL</td>
<td>Alias</td>
<td>Records</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/transactions/v2_5_0/sales.xsd">https://webservices.netsuite.com/xsd/transactions/v2_5_0/sales.xsd</a></td>
<td>tranSales</td>
<td>Opportunities Sales Orders Item Fulfillment Invoice CashSale Estimate</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/transactions/v2_5_0/general.xsd">https://webservices.netsuite.com/xsd/transactions/v2_5_0/general.xsd</a></td>
<td>tranGeneral</td>
<td>Journal Entries</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/lists/v2_5_0/accounting.xsd">https://webservices.netsuite.com/xsd/lists/v2_5_0/accounting.xsd</a></td>
<td>listAcct</td>
<td>Other Lists Items</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/lists/v2_5_0/relationships.xsd">https://webservices.netsuite.com/xsd/lists/v2_5_0/relationships.xsd</a></td>
<td>listRel</td>
<td>Customers Contacts Partner Vendor</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/lists/v2_5_0/support.xsd">https://webservices.netsuite.com/xsd/lists/v2_5_0/support.xsd</a></td>
<td>listSupport</td>
<td>Cases</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/setup/v2_5_0/customization.xsd">https://webservices.netsuite.com/xsd/setup/v2_5_0/customization.xsd</a></td>
<td>setupCustom</td>
<td>Custom Records Custom Fields (All)</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/lists/v2_5_0/employees.xsd">https://webservices.netsuite.com/xsd/lists/v2_5_0/employees.xsd</a></td>
<td>listEmp</td>
<td>Employee</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/documents/v2_5_0/fileCabinet.xsd">https://webservices.netsuite.com/xsd/documents/v2_5_0/fileCabinet.xsd</a></td>
<td>docfileCab</td>
<td>File</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/transactions/v2_5_0/bank.xsd">https://webservices.netsuite.com/xsd/transactions/v2_5_0/bank.xsd</a></td>
<td>tranBank</td>
<td>Check</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/transactions/v2_5_0/inventory.xsd">https://webservices.netsuite.com/xsd/transactions/v2_5_0/inventory.xsd</a></td>
<td>tranInvt</td>
<td>Inventory Adjustment</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/transactions/v2_5_0/purchases.xsd">https://webservices.netsuite.com/xsd/transactions/v2_5_0/purchases.xsd</a></td>
<td>tranPurch</td>
<td>VendorBill Purchase Order</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/transactions/v2_5_0/customers.xsd">https://webservices.netsuite.com/xsd/transactions/v2_5_0/customers.xsd</a></td>
<td>tranCust</td>
<td>CashRefund CustomerPayment ReturnAuthorization CreditMemo</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/lists/v2_5_0/website.xsd">https://webservices.netsuite.com/xsd/lists/v2_5_0/website.xsd</a></td>
<td>listSite</td>
<td>SiteCategory</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/transactions/v2_5_0/employees.xsd">https://webservices.netsuite.com/xsd/transactions/v2_5_0/employees.xsd</a></td>
<td>tranEmp</td>
<td>TimeBill</td>
</tr>
</tbody>
</table>

### System Constants XSD Files

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

<table>
<thead>
<tr>
<th>URL</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/activities/v2_5_0/schedulingTypes.xsd">https://webservices.netsuite.com/xsd/activities/v2_5_0/schedulingTypes.xsd</a></td>
<td>actSchedTyp</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/general/v2_5_0/communicationTypes.xsd">https://webservices.netsuite.com/xsd/general/v2_5_0/communicationTypes.xsd</a></td>
<td>generalCommTyp</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/lists/v2_5_0/relationshipsTypes.xsd">https://webservices.netsuite.com/xsd/lists/v2_5_0/relationshipsTypes.xsd</a></td>
<td>listRelTyp</td>
</tr>
<tr>
<td><a href="https://webservices.netsuite.com/xsd/lists/v2_5_0/supportTypes.xsd">https://webservices.netsuite.com/xsd/lists/v2_5_0/supportTypes.xsd</a></td>
<td>listSupportTyp</td>
</tr>
</tbody>
</table>
Example

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

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

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

Notice that the element called `platformMsgs:add` is not contained in the WSDL itself but is referenced from the related platformMsgs XSD file. In this case, the platformMsgs alias refers to the xsd file at:

https://webservices.netSuite.com/xsd/platform/v2_5_0/messages.xsd

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

```xml
<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/v2_5_0/core.xsd
The abstract type Record is defined as:

```xml
<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 record types.

## 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 including:

- Encryption
- Authentication and Authorization
- Session Management

### Authentication

Authentication is the process of determining the identity of the requester by verifying that they are who they claim to be based upon the credentials they present. The SuiteTalk Platform requires a valid user name, password and account number for authentication. These are provided via the login operation using the Passport type.

```xml
<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. All Web services operations require authentication.

**Note:** In order 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:

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

### Session Management

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

Sessions are controlled by the following:

- **Session Timeouts:** Similar to the UI, Web services sessions automatically time-out after 15 minutes of inactivity, requiring a login to resume activity. However, if the server resubmits the cookie during the first 15 minutes, the inactivity timer is reset.
**Note:** If you need shorter sessions for security reasons, you should create a login routine for the client that automatically re-logins after a specified idle time.

If you explicitly logout of a session, and then attempt to utilize 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.

If you need to eliminate open sessions for security reasons, you should call logout upon completion of an operate.

- **Session Limits:** A given login (username/password) is currently limited to two sessions, one through the browser and one through Web services.

  For example, if a user attempts to log in to his/her account from two separate browsers, the second browser login terminates the first browser session (assuming the same credentials).

  Likewise, if two Web services clients attempt to establish two concurrent sessions, the first session is terminated. However, UI sessions and Web services sessions do not cause termination of each other.

  Note that more than two independent concurrent sessions are possible with the purchase of certain products such as Offline Client or Outlook Integration.

  **Important:** Using the same login from multiple client applications, or multiple instances of the same application is NOT supported.

**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 2005 18:43:27 GMT
Server: Oracle-Application-Server-10g/9.0.4.0.0 Oracle-HTTP-Server
Set-Cookie: NS_VER=11.0.0; domain=joe.corp.netsuite.com; path=/
Vary: User-Agent
Set-Cookie: JSESSIONID=ac101faelf4312dfxxx062fc829447eaa0c3ddc70af41d; Path=/
Set-Cookie: lastUser=TS3TRV198400_935_3; Expires=Wed, 25-May-2005 17:42:24 GMT;
Path=/
Cache-Control: private
Keep-Alive: timeout=150, max=1000
```
For each subsequent request, all of the cookies must be returned in the HTTP header to maintain the session as follows:

**Sample Code (C#)**
```csharp
NetSuiteService nss = new NetSuiteService();
nss.Url = https://webservices.netsuite.com/services/NetSuitePort_2_0;
nss.CookieContainer = new CookieContainer();
```

**Sample Code (Java with Axis 1.2 or Axis 1.3)**
```java
NetSuiteServiceLocator nss = new NetSuiteServiceLocator();
nss.setMaintainSession(true);
```

### Authorization

Authorization is the process of ensuring that the requester has the appropriate entitlement to perform the requested operation. When a user requests to be authenticated, they also provide a role so that they are logged in under that 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:

```xml
<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>
```

For detailed information on roles and permissions and how the SuiteTalk Platform implements roles and permissions rules refer to “Working with Roles and Permissions” on page 26.

### Encryption

All 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 nonsecure Web service requests are granted.

### Using Internal IDs, External IDs, and References

Each record in NetSuite is uniquely identified by its record type in combination with either a system-generated NetSuite internal ID or an external ID that has been provided at the time of record creation or during an update. Internal and External IDs are NOT unique across different record types. Therefore, both the record type and either the internal ID or the external ID are required to perform an operation on an existing record and both are returned for each record in all operations on records.

**Note:** 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.
The term *reference* or *ref* is used to define a reference to any existing record (record type/id combination) in the system. References are implemented through the RecordRef type described in the following XSD file:

https://webservices.netsuite.com/xsd/platform/v2_5_0/core.xsd

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

```xml
<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>
  </complexContent>
</complexType>
```

**Important:** When referencing records, you must provide the internalID OR the externalID attribute for all update operations but not both in the same operations. If both are provided, the internalID is used to locate the record and the externalID is ignored.

### Understanding ExternalIDs

The externalID attribute of a recordRef provides a means to reference an object by its foreign key in an external database. ExternalIDs can be set during an Add or Update. ExternalIDs are useful for the following two situations:

- **Maintaining client ID relationships**
  In cases where a client application already maintains references between record, set the externalID 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 externalID is used, you would need to import the customer records, determine the IDs of the related sales reps employee records and then reimport the customer records with the sales reps ID references. By providing an externalID you can import the customer records in a single API call using the externalID references to the sales reps.

### List of Operations and 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, since 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.
Example Internal ID Usage

The following are partial SOAP messages of operations demonstrating the use of internal IDs.

The 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)

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

Using Internal IDs, External IDs, and References

(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>

The 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>

The deleteList operation

In the following delete request, the internal IDs are required for the request and returned in the response. In this case 3 Event records are deleted (100101, 100102 and 100103).

(Request)

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

(Response)

<soapenv:Body>
  <deleteListResponse xmlns="urn:messages_2_5.platform.webservices.netSuite.com">
  </deleteListResponse>
</soapenv:Body>
<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>
  </writeResponseList>

The getList operation

In this case, 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:RecordRef"/>
    <platformMsgs:baseRef internalId="100105" type="calendarEvent" xsi:type="platformCore:RecordRef"/>
    <platformMsgs:baseRef internalId="100106" type="calendarEvent" xsi:type="platformCore:RecordRef"/>
  </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>
            ........[more fields]
          </record>
        </writeResponse>
      </readResponse>
    </readResponseList>
  </getListResponse>
</soapenv:Body>
The 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)

```xml
<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)

```xml
<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>
```
Processing Requests Asynchronously

Web services requests can be processed synchronously or asynchronously.

In synchronous requests, your client application sends a request to the SuiteTalk Platform where it is processed and a response is returned. The client application handles the response as appropriate.

In asynchronous requests, your client application sends a request to the NetSuite Web Services Platform where it is placed in a processing queue and handled asynchronously with other requests. 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.

**Note:** Asynchronous request JobIDs are valid for 30 days.

Asynchronous equivalents are available for the following operations:
- addList: asyncAddList
- updateList: asyncUpdateList
- deleteList: asyncDeleteList
- getList: asyncGetList
- search: asyncSearch

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

**Request**

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

The response, however differs in that only information about the request is returned instead of the actual record. You can then use the jobID to reference the request later.

**Response**

```
<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>
    </asyncStatusResult>
  </asyncGetListResponse>
</soapenv:Body>
```
Use the getAsyncResult or checkAsyncStatus operations to track the asynchronous request.

**checkAsyncStatus**

The checkAsyncStatus operation can be used to check the status of an asynchronous web services submission. When a jobId is submitted, the status, percent complete and estimated remaining duration is returned.

Possible status values that can be returned include:

- failed
- finishedWithErrors
- pending
- processing
- finished

If the status is failed, finishedWithErrors, or finished, you can use the getAsyncResult operation to provide the detailed of the response whether that be the record results, or error and fault messages.

**Note:** The percent complete and estimated remaining duration, although generally accurate, may vary depending on increases or decreases in database activity during the web services processing.

**Request**

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

**Response**

```xml
<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>
```

**Faults**

This operation can throw one of the following faults. See “Soap Fault Status Codes” on page 138 for more information on faults.

- InvalidSessionFault
• AsyncFault
• UnexpectedErrorFault

getAsyncResult

The getAsyncResult operation can be used to retrieve the results of an asynchronous web services submission.

Faults

This operation can throw one of the following faults. See “Soap Fault Status Codes” on page 138 for more information on faults.

• InvalidSessionFault
• ExceededRequestLimitFault
• ExceededUsageLimitFault
• ExceededRecordCountFault
• AsyncFault
• UnexpectedErrorFault

Monitoring Asynchronous Jobs in the UI

Web services jobs submitted asynchronously can be monitored on the Web Services Job Status page at Setup > Import/Export > Web Services Job Status.

On this page you can do the following:

• View the status of the job such as percent complete and estimated time.
• View the Job Id. The Job Id, which is also returned in the asynch SOAP response, can be used to programmatically retrieve job status information.
• View the SOAP request or response associated with a job. The response can only be viewed once a job has successfully completed.
• Cancel a Job. A job can only be cancelled if it has not yet successfully completed.

Note: The Job Status page does not automatically refresh. Click refresh in your browser window to get the current status of a job.

Web Services to UI Single Login

With NetFlex Web Services, a single login can be provided and access to both the web services application and the NetSuite UI maintained. The web services application submits a POST to a specific wslogin page URL with credentials and a taskId identifying the desired NetSuite page. The wslogin page then authenticates the request, provides a new session and redirects to the requested page.
Important: Using the POST mechanism ensures security because the user's credentials will be encrypted using https and, therefore, will NOT be vulnerable during transmission.

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Required for Authentication</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>email</td>
<td>text</td>
<td>Yes</td>
<td>Must reflect a valid email address of an entity in your account.</td>
</tr>
<tr>
<td>account</td>
<td>text</td>
<td>Yes</td>
<td>The account ID. To view the account ID, go to Setup &gt; Integration &gt; Web Services.</td>
</tr>
<tr>
<td>password</td>
<td>password</td>
<td>Yes</td>
<td>The password of the entity associated with the email provided.</td>
</tr>
<tr>
<td>taskid</td>
<td>text</td>
<td>Yes</td>
<td>For a list of currently exposed NetSuite TaskIDs, refer to &quot;Task IDs&quot; on page 160. To find the TaskID for a given page while in NetSuite, view the source of the page and search for main_help_anchor. A snippet similar to the following is found: <code>main_help_anchor' href=&quot;#&quot; onclick=&quot;nlPopupHelp('LIST_SAVEDSEARCH','Full');</code> (where LIST_SAVEDSEARCH is the taskID of the page)</td>
</tr>
<tr>
<td>id</td>
<td>text</td>
<td>Yes</td>
<td>The ID of a record type available in your NetSuite account.</td>
</tr>
<tr>
<td>role</td>
<td>text</td>
<td>Optional</td>
<td>Sets the role for the login. This must be a valid role for the user logging in.</td>
</tr>
<tr>
<td>e</td>
<td>text</td>
<td>Optional</td>
<td>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.</td>
</tr>
</tbody>
</table>

The post URL is: /app/webservices/wslogin.nl.

Note: Since every NetSuite user is restricted to a single UI session at any given time, initiating a new browser session using this mechanism will invalidate an existing session.

Sample HTML

The following HTML provides a simple login page where clicking Submit sends the user to the Item page in edit mode with administrator privileges. Edit the email, account, and password input fields with values from your account to successfully run this sample.

```html
<html>
<body>
<form action="https://system.netsuite.com/app/webservices/wslogin.nl" method="POST"/>
<p>
email <input type="text" value="name@netsuite.com" name="email"/>
<br/>account <input type="text" value="#####" name="account"/>
<br/>password <input type="password" value="foobaaa!" name="password"/>
<br/>role <input type="text" size="50" value="3" name="role"/>
<br/>taskid <input type="text" size="50" value="EDIT_ITEM" name="taskid"/>
<br/>record id <input type="text" size="3" value="45" name="id"/>
<br/>edit flag <input type="text" size="1" value="" name="e"/>
```
Handling Errors

The SuiteTalk Platform supports three different exception types that are returned in the SOAP response in the event of a problem. The client can then take the appropriate action based on the error code or fault that is received.

- **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” on page 21.
- **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 Your Web Services Preferences” on page 21 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):

```xml
<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” on page 159.

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.

```xml
<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” on page 139.

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.

```xml
<soapenv:Fault>
  <faultcode>soapenv:Server.userException</faultcode>
  <faultstring>com.netledger.dto.faults.InvalidSessionFault: Your connection has timed out. Please log in again.</faultstring>
</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.

```xml
<soapenv:Fault>
  <faultcode>soapenv:Server.userException</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” on page 138.
Chapter 4  Types

This chapter 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 in order 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 Platform 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 Platform implementation uses the following built-in types. For detailed information on XML Schema built-in types, refer to [http://www.w3.org/TR/xmlschema-2/](http://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 either be set to true or false. If your Web service does not explicitly set the field as true or false the field returns 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.

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.

**NetSuite 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.
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.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>xsd:string (attribute)</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>xsi:type</td>
<td>xsi:type (attribute)</td>
<td>N</td>
<td>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.</td>
</tr>
<tr>
<td>nullFieldList</td>
<td>NullFields[]</td>
<td>N</td>
<td>A list of fields that are to be set to null explicitly in the update operation.</td>
</tr>
</tbody>
</table>

**RecordList**

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

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>record</td>
<td>Record []</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>BaseRef</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>internalId</td>
<td>xsd:string (attribute)</td>
<td>Y</td>
<td>See “Using Internal IDs, External IDs, and References” on page 38.</td>
</tr>
<tr>
<td>type</td>
<td>xsd:string (attribute)</td>
<td>N</td>
<td>Reference to a value in a Web services only system list. If the type is known by context, the type is NOT required.</td>
</tr>
<tr>
<td>name</td>
<td>xsd:string</td>
<td>N</td>
<td>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.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>internalId</td>
<td>xsd:string (attribute)</td>
<td>Y</td>
<td>References the primary record internal Id. This Id corresponds to the type of custom record. For a list of possible values, see “Record Internal IDs” on page 13 in the Web Services Records Guide.</td>
</tr>
<tr>
<td>typeId</td>
<td>xsd:string</td>
<td>Y</td>
<td>References the custom record type Id.</td>
</tr>
<tr>
<td>type</td>
<td>xsd:string (attribute)</td>
<td>N</td>
<td>Reference to a value in a Web services only system list.</td>
</tr>
<tr>
<td>name</td>
<td>xsd:string</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

ListOrRecordRef

The listOrRecordRef type is defined in core.xsd.
### Status

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

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>internalId</td>
<td>xsd:string (attribute)</td>
<td>Y</td>
<td>See “Using Internal IDs, External IDs, and References” on page 38.</td>
</tr>
<tr>
<td>typeId</td>
<td>xsd:string (attribute)</td>
<td>N</td>
<td>Reference to a value in a Web services only system list.</td>
</tr>
<tr>
<td>name</td>
<td>xsd:string</td>
<td>N</td>
<td>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.</td>
</tr>
</tbody>
</table>

### StatusDetail

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

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

### NullField

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

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>xsd:string</td>
<td>Y</td>
<td>Name of the field to be null. The specified name must exactly match an existing field name.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>Status</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>recordRef</td>
<td>RecordRef</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>response</td>
<td>ListReadResponse[]</td>
<td>Y</td>
<td>An array of ReadResponse types.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>response</td>
<td>WriteResponse[]</td>
<td>Y</td>
<td>An array of WriteResponse types.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>Status</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>recordRef</td>
<td>RecordRef</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>XML Schema Type</th>
<th>Custom Field Type in UI</th>
</tr>
</thead>
<tbody>
<tr>
<td>IntCustomFieldRef</td>
<td>Integer Number</td>
</tr>
<tr>
<td>DoubleCustomFieldRef</td>
<td>Decimal Number</td>
</tr>
<tr>
<td>StringCustomFieldRef</td>
<td>Free-Form Text, Text Area, Phone Number, E-mail Address, Hyperlink, Rich Text</td>
</tr>
<tr>
<td>BooleanCustomFieldRef</td>
<td>Check Box</td>
</tr>
<tr>
<td>DateCustomFieldRef</td>
<td>Date</td>
</tr>
<tr>
<td>SelectCustomFieldRef</td>
<td>List/Record</td>
</tr>
<tr>
<td>MultiSelectCustomFieldRef</td>
<td>Multiple Select</td>
</tr>
</tbody>
</table>

**CustomFieldRef**

The CustomFieldRef type is an abstract type.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>internalId</td>
<td>xsd:string (attribute)</td>
<td>Y</td>
<td>References a unique instance of a custom field type.</td>
</tr>
</tbody>
</table>

To locate the internal ID for a given custom field in the UI, go to Setup > Customization > [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.

**IntCustomFieldRef**

The IntCustomFieldRef type extends the CustomFieldRef abstract type.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>xsd:int</td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>
**DoubleCustomFieldRef**

The DoubleCustomFieldRef type extends the CustomFieldRef abstract type.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>xsd:double</td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>

**BooleanCustomFieldRef**

The BooleanCustomFieldRef type extends the CustomFieldRef abstract type.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>xsd:boolean</td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>

**StringCustomFieldRef**

The StringCustomFieldRef type extends the CustomFieldRef abstract type.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>xsd:string</td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>

**DateCustomFieldRef**

The DateCustomFieldRef type extends the CustomFieldRef abstract type.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>xsd:datetime</td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>ListOrRecordRef</td>
<td>Y</td>
<td>A single ListOrRecordRef.</td>
</tr>
</tbody>
</table>

**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.
CustomFieldList

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>customFieldRef[]</td>
<td>Y</td>
<td>An array of type customFieldRef. The actual entries in the array will be of a concrete type that extends customFieldRef.</td>
</tr>
</tbody>
</table>

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

```xml
<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="IntCustomFieldRef" 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

Searching within Web services is implemented such that every implemented record type has a corresponding search record type. For example, there is a CustomSearch type in addition to a Customer record type.

Search XML Schema Types

The following sections define the available search types. Every search field within a search object type must belong to one of these search types.
### SearchPreferences

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>bodyFields Only</td>
<td>boolean</td>
<td>Y</td>
<td>Defaults to True</td>
</tr>
<tr>
<td>pageSize</td>
<td>int</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SearchRequest

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>preferences</td>
<td>boolean</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>searchRecord</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SearchResult

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>totalRecords</td>
<td>int</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>pageSize</td>
<td>int</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>totalPages</td>
<td>int</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>pageIndex</td>
<td>int</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

### SearchStringField

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>operator</td>
<td>platformCoreTyp:</td>
<td>Y</td>
<td>Reference to a value in a system list. For more information on available values, see “Platform Enumerations” on page 66.</td>
</tr>
<tr>
<td></td>
<td>SearchStringFieldOperator (attribute)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>searchValue</td>
<td>xsd:string</td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>

### SearchBooleanField

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>operator</td>
<td>xsd:boolean</td>
<td>Y</td>
<td>The available values are true or false.</td>
</tr>
</tbody>
</table>

### SearchDoubleField

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>operator</td>
<td>platformCoreTyp: SearchDoubleFieldOperator (attribute)</td>
<td>Y</td>
<td>Reference to a value in a system list. For more information on available values, see “Platform Enumerations” on page 66.</td>
</tr>
</tbody>
</table>
### SearchIntField

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>operator</td>
<td>platformCoreTyp: SearchIntFieldOperator (attribute)</td>
<td>Y</td>
<td>Reference to a value in a system list. For more information on available values, see “Platform Enumerations” on page 66.</td>
</tr>
<tr>
<td>searchValue</td>
<td>xsd:int</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>searchValue2</td>
<td>xsd:int</td>
<td>N</td>
<td>If the operator is between or notBetween searchValue2 must be populated.</td>
</tr>
</tbody>
</table>

### SearchDateField

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>operator</td>
<td>platformCoreTyp: SearchDate (attribute)</td>
<td>Y</td>
<td>Reference to a value in a system list. For more information on available values, see “Platform Enumerations” on page 66.</td>
</tr>
<tr>
<td>predefinedSearchValue</td>
<td>platformCoreTyp: SearchDate</td>
<td>N</td>
<td>Reference to a value in a system list. For more information on available values, see “Platform Enumerations” on page 66.</td>
</tr>
<tr>
<td>searchValue</td>
<td>xsd:date</td>
<td>N</td>
<td>Either predefinedSearchValue or searchValue should be populated.</td>
</tr>
<tr>
<td>searchValue2</td>
<td>xsd: date</td>
<td>N</td>
<td>If the operator is between or notBetween searchValue2 must be populated.</td>
</tr>
</tbody>
</table>

### 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.
**Search Types**

SearchEnumMultiSelectField

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

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>operator</td>
<td>platformCoreTyp: SearchMultiSelectFieldOperator (attribute)</td>
<td>Y</td>
<td>Reference to a value in a system list. For more information on available values, see “Platform Enumerations” on page 66.</td>
</tr>
<tr>
<td>searchValue</td>
<td>platformCore: RecordRef</td>
<td>Y</td>
<td>An array of type SearchMultiSelectRefValue.</td>
</tr>
</tbody>
</table>

SearchCustomField XML Schema Types

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

**SearchCustomField**

This is an abstract type.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>xsd:string (attribute)</td>
<td>Y</td>
<td>References a unique instance of a custom field</td>
</tr>
</tbody>
</table>

**SearchStringCustomField**

The SearchStringCustomField type extends the SearchCustomField abstract type.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>xsd:string (attribute)</td>
<td>Y</td>
<td>Inherited from the SearchCustomField. Reference a unique instance of a custom field.</td>
</tr>
<tr>
<td>operator</td>
<td>platformCoreTyp: SearchStringFieldOperator (attribute)</td>
<td>Y</td>
<td>The type is an enumeration type that restricts the value to a predefined list. For more information on available values, see “Platform Enumerations” on page 66.</td>
</tr>
<tr>
<td>searchValue</td>
<td>xsd:string</td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>
SearchBooleanCustomField

The SearchBooleanCustomField type extends the SearchCustomField abstract type.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>xsd:string (attribute)</td>
<td>Y</td>
<td>Inherited from the SearchCustomField. Reference a unique instance of a custom field.</td>
</tr>
<tr>
<td>operator</td>
<td>platformCoreTyp: SearchBooleanFieldOperator (attribute)</td>
<td>Y</td>
<td>The type is an enumeration type that restricts the value to true or false.</td>
</tr>
</tbody>
</table>

SearchIntCustomField

The SearchIntCustomField type extends the SearchCustomField abstract type.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>xsd:string (attribute)</td>
<td>Y</td>
<td>Inherited from the SearchCustomField. Reference a unique instance of a custom field.</td>
</tr>
<tr>
<td>operator</td>
<td>platformCoreTyp: SearchIntFieldOperator (attribute)</td>
<td>Y</td>
<td>The type is an enumeration type that restricts the value to a predefined list. For more information on available values, see “Platform Enumerations” on page 66.</td>
</tr>
<tr>
<td>searchValue</td>
<td>xsd:int</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>searchValue2</td>
<td>xsd:int</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

SearchDoubleCustomField

The SearchDoubleCustomField type extends the SearchCustomField abstract type.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>xsd:string (attribute)</td>
<td>Y</td>
<td>Inherited from the SearchCustomField. Reference a unique instance of a custom field.</td>
</tr>
<tr>
<td>operator</td>
<td>platformCoreTyp: SearchDoubleFieldOperator (attribute)</td>
<td>Y</td>
<td>The type is an enumeration type that restricts the value to a predefined list. For more information on available values, see “Platform Enumerations” on page 66.</td>
</tr>
<tr>
<td>searchValue</td>
<td>xsd:double</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>searchValue2</td>
<td>xsd:double</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

SearchDateCustomField

The SearchDateCustomField type extends the SearchCustomField abstract type.
The SearchSelectCustomField type extends the SearchCustomField abstract type.

The SearchMultiSelectCustomField type extends the SearchCustomField abstract type.

The SearchEnumMultiSelectCustomField type extends the SearchCustomField abstract type.
SearchCustomFieldList

<table>
<thead>
<tr>
<th>Field Name</th>
<th>XML Schema Type</th>
<th>Req</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>customField</td>
<td>platformCore: SearchCustomField[]</td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>

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.

```xml
<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"
```
Platform Enumerations

The following record and 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.

<table>
<thead>
<tr>
<th>Enumerations</th>
<th>Record Types</th>
<th>RecordType</th>
<th>GetAllRecordTypes</th>
</tr>
</thead>
<tbody>
<tr>
<td>calendarEvent</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>task</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>customer</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>contact</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>message</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>note</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>supportCase</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>opportunity</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>customRecord</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>customerCategory</td>
<td>X X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>contactCategory</td>
<td>X X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>priceLevel</td>
<td>X X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>winLossReason</td>
<td>X X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>term</td>
<td>X X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>noteType</td>
<td>X X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>paymentMethod</td>
<td>X X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>leadSource</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Search Types (Table 1)

<table>
<thead>
<tr>
<th>Enumerations</th>
<th>searchString FieldOperator</th>
<th>searchIntField Operator</th>
<th>searchDouble FieldOperator</th>
<th>searchPeriod Field</th>
<th>searchTime FieldOperator</th>
</tr>
</thead>
<tbody>
<tr>
<td>is</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>isNot</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>startsWith</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>doesNotStartWith</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>contains</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Search Types (Table 1)

<table>
<thead>
<tr>
<th>Enumerations</th>
<th>searchString FieldOperator</th>
<th>searchIntField Operator</th>
<th>searchDouble FieldOperator</th>
<th>searchPeriod Field</th>
<th>searchTime FieldOperator</th>
</tr>
</thead>
<tbody>
<tr>
<td>doesNotContain</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>equalTo</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lessThan</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>greaterThan</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lessThanOrEqualTo</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>greaterThanOrEqualTo</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>notEqualTo</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>notLessThan</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>notGreaterThan</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>notLessThanOrEqualTo</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>notGreaterThanOrEqualTo</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>between</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>notBetween</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>empty</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>notEmpty</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Search Types (Table 2)

<table>
<thead>
<tr>
<th>Enumerations</th>
<th>searchDate FieldOperator</th>
<th>searchEnum MultiSelect FieldOperator</th>
<th>searchMulti SelectFieldOperator</th>
<th>searchDate</th>
</tr>
</thead>
<tbody>
<tr>
<td>anyOf</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>noneOf</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>on</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>before</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>after</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>onOrBefore</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>onOrAfter</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>within</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>empty</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>notOn</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>notBefore</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>notAfter</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>notOnOrBefore</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>notOnOrAfter</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>notWithin</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enumerations</td>
<td>SearchDate FieldOperator</td>
<td>SearchEnum MultiSelect FieldOperator</td>
<td>SearchMulti SelectField Operator</td>
<td>SearchDate</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>--------------------------</td>
<td>--------------------------------------</td>
<td>----------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>notEmpty</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lastBusinessWeek</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>lastFiscalQuarter</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lastFiscalQuarterToDate</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lastFiscalYear</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lastFiscalYearToDate</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lastMonth</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lastMonthToDate</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lastRollingQuarter</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lastRollingYear</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lastWeek</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lastWeekToDate</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nextBusinessWeek</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nextFiscalQuarter</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nextFiscalYear</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nextFourWeeks</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nextMonth</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nextOneMonth</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nextOneQuarter</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nextOneWeek</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nextOneYear</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nextWeek</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>previousOneDay</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>previousOneMonth</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>previousOneQuarter</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>previousOneWeek</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>previousOneYear</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>previousRollingQuarter</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>previousRollingYear</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sameMonthLastFiscalQuarter</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sameMonthLastFiscalQuarterToDate</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sameMonthLastFiscalYear</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Enumerations</td>
<td>SearchDate FieldOperator</td>
<td>SearchEnum MultiSelect FieldOperator</td>
<td>SearchMulti SelectField Operator</td>
<td>SearchDate</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>--------------------------</td>
<td>-------------------------------------</td>
<td>----------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>sameMonthLastFiscalYearToDate</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>sameQuarterLastFiscalYear</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>sameQuarterLastFiscalYearToDate</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>thisBusinessWeek</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>thisFiscalQuarter</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>thisFiscalQuarterToDate</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>thisFiscalYear</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>thisFiscalYearToDate</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>thisMonth</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>thisMonthToDate</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>thisRollingQuarter</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>thisRollingYear</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>thisWeek</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>thisWeekToDate</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>thisYear</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>today</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>tomorrow</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>yesterday</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Chapter 5  Operations

This section describes the operations that can be used in requests to retrieve and change data in your NetSuite account. For example, you can use operations to login to your NetSuite account, and then retrieve, add, update or delete a record.

The following operations are supported:

- login
- mapSso
- logout
- addList
- add
- updateList
- update
- deleteList
- delete
- search
- searchMore
- searchNext
- getList
- get
- getAll
- getSelectValue
- getCustomization
- getItemAvailability
- attachTo / attachedRecord
- changePasswordOrEmail
- getDeleted
- initialize

**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. The login operation provides a passport that includes a username, password, account and role. On success, the NetSuite server sets a cookie and establishes a session.
**Note:** All sessions have a time-out. 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. For more information on session management, see “Session Management” on page 36.

All available Web services operations require that the user first be logged in. If an operation is executed before a login is performed, it fails and the InvalidSessionFault is returned.

The login operation also verifies that the specified account has the Web Services feature turned on. If this is not the case, the SOAP fault InvalidCredentialsFault is returned with a code of WEB_SERVICES_NOT_ENABLED.

### Request

The LoginRequest type is used for this request. It contains the following fields:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>passport</td>
<td>Passport</td>
<td>Contains all the required credentials including username, password, account and role to authenticate the user and create a new session.</td>
</tr>
</tbody>
</table>

The Passport type takes the following fields:

- 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 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 relogin with a different role.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>userID</td>
<td>string</td>
<td>References an existing user record.</td>
</tr>
<tr>
<td>role</td>
<td>RecordRef</td>
<td>References an existing role record.</td>
</tr>
<tr>
<td>isDefault</td>
<td>boolean</td>
<td>If true, the role is set as the default role for this user.</td>
</tr>
<tr>
<td>isInactive</td>
<td>boolean</td>
<td>If true, this role is currently unavailable for the current user.</td>
</tr>
</tbody>
</table>
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.

```java
String userid = getHeader(port, "sessionInfo");
public static String getHeader(NetSuitePortType port, String headerName) {
    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” on page 138 for more information on faults.

- InsufficientPermissionFault
- InvalidAccountFault
- 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 internal IDs see “IDs Associated with Roles” on page 27.

```xml
<soap:Body>
<platformMgs:login>
<platformMgs:passport>
<platformCore:email>eblack@yahoo.com</platformCore:email>
<platformCore:password>Cracker1</platformCore:password>
<platformCore:account>111111</platformCore:account>
<platformCore:role internalId="3"/>
</platformMgs:passport>
</platformMgs:login>
</soap:Body>
```

SOAP Response

```xml
<soapenv:Body>
<loginResponse xmlns="urn:messages_1_2.platform.webservices.netsuite.com">
<sessionResponse xmlns="urn:messages_1_2.platform.webservices.netsuite.com">
```
private void login( bool isAuto )
{
    if ( !_isAuthenticated )
    {
        // Check whether this is a forced login as part of another operation
        if ( isAuto )
            _out.writeLn( "You 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( "Please 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();
        }
    }
}
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 ...\n");

        // 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.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\n");
        _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
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("The 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));
}
}
```

**mapSso**

Single Signon (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 streamlines the provisioning of Single Signon User Accounts by providing the ability to correlate a partner/customer specified internal ID with a Netsuite credential. This automates the mapping between external applications credentials and NetSuite's credentials for a user.

**Important:** This operation is a one-time mapping — NOT a login. The operation does NOT include the ability to login via a Single Signon Token, nor does it allow for the provisioning of a partner — it does not aid with public/private key exchange. Also, this mapping is allowed ONLY for an administrator.

This section describes the mapSso operation. For more detailed information on NetSuite’s SSO technology, contact your NetSuite Account Manager or Professional Services Manager for the **Single Signon Developer’s Guide**.

**Request**

The MapSsoRequest type is used for this request. It contains the following fields:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssoCredentials</td>
<td>SsoCredentials</td>
<td>Contains all the required credentials including username, password, account and role to authenticate the user and create a new session.</td>
</tr>
</tbody>
</table>

The SsoCredentials type requires the following fields:
- email
- password
- account
- role
- authentication Token: this is a string representing the Private Key encrypted Netsuite token.

This token, prior to any hex-encoding, is of the form:

```xml
<companyId><space><userId><space><timestamp>
```

- `<companyId>` is the integration partner's affiliate ID as provided to the partner from NetSuite (typically five numeric digits long). The `<companyId>` should be equal to `<userId>` when the sending system does not support the notion of subusers. Since spaces are used to delimit subtokens, no spaces can be used in the tokens. The timestamp string is a decimal representation of the number of milliseconds since January 1, 1970, 00:00:00 GMT. Optionally, a return URL can also be included. This is the URL the user is redirected to upon logout from NetSuite.

- partnerId

**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.

**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” on page 138 for more information on faults.

- InsufficientPermissionFault
- InvalidAccountFault
- InvalidAccountFault
- InvalidCredentialsFault
- InvalidVersionFault
- UnexpectedErrorFault

**Sample Code: SOAP**

```xml
<soap:Body>
<platformMsgs:mapSso>
<platformMsgs:ssoCredentials>
<platformCore:email>bbrown@yahoo.com</platformCore:email>
<platformCore:password>Cracker1</platformCore:password>
<platformCore:account>TSTDRV211915</platformCore:account>
<platformCore:authenticationToken>[enter token here]</platformCore:authenticationToken>
<platformCore:partnerId>[enter partner id here]</platformCore:partnerId>
</platformMsgs:ssoCredentials>
```

SuiteTalk Platform - Version 2.5.0
Sample Code: Java

```java
/* Generate a NetSuitePort */
 NetSuiteServiceLocator nss = new NetSuiteServiceLocator();
 NetSuitePortType 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().isSuccess())
     throw new Exception("Mapping Failed: " +
     sr.getStatus().getStatusDetail(0).getMessage());
 }
```

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” on page 138 for more information on faults.

- UnexpectedErrorFault

**Sample Code**

Following is an example that contains an excerpt of the SOAP body for both the request and response.
**SOAP Request**

```xml
<logout/>
```

**SOAP Response**

```xml
<logoutResponse>
  <status isSuccess="true"/>
</logoutResponse>
```

**C#**

```csharp
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**

```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" );
    }
}
```
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

Request

The AddListRequest type is used for the request.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>record</td>
<td>Record</td>
<td>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.</td>
</tr>
</tbody>
</table>

Response

The AddListResponse type is used for the response.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>response</td>
<td>WriteResponse</td>
<td>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.</td>
</tr>
</tbody>
</table>

Faults

This operation can throw one of the following faults. See “Soap Fault Status Codes” on page 138 for more information on faults.

- InvalidSessionFault
- ExceededRateLimitFault
- ExceededMaxRecordsFault
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.

```xml
<soap:Body>
  <platformMsgs:addList>
    <platformMsgs:record xsi:type="listRel:Customer">
      <listRel:entityId>Shutter Fly</listRel:entityId>
      <listRel:companyName>Shutter Fly, Inc</listRel:companyName>
    </platformMsgs:record>
    <platformMsgs:record xsi:type="listRel:Customer">
      <listRel:entityId>GNC</listRel:entityId>
      <listRel:companyName>GNC Corp</listRel:companyName>
    </platformMsgs:record>
  </platformMsgs:addList>
</soap:Body>
```

SOAP Response

In the response, notice that the internalID for each record added is returned.

```xml
<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>
```

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.

**Request**

The AddRequest type is used for the request. It contains the following fields.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>record</td>
<td>Record</td>
<td>The record type is an abstract type so an instance of a type that extends record must be used — such as Customer or Event.</td>
</tr>
</tbody>
</table>
Response

The AddResponse type is used for the response. It contains the following fields.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>response</td>
<td>WriteResponse</td>
<td>Contains details on the status of the operation and a reference to the created record.</td>
</tr>
</tbody>
</table>

Faults

This operation can throw one of the following faults. See “Soap Fault Status Codes” on page 138 for more information on faults.

- InvalidSessionFault
- ExceededRateLimitFault
- ExceededMaxRecordsFault
- UnexpectedErrorFault

Sample Code

SOAP Request

In this example, a single customer record is added.

```xml
<soap:Body>
<platformMsgs:add>
  <platformMsgs:record xsi:type="listRel:Customer">
    <listRel:entityId>Shutter Fly</listRel:entityId>
    <listRel:companyName>Shutter Fly, Inc</listRel:companyName>
  </platformMsgs:record>
</platformMsgs:add>
</soap:Body>
```

SOAP Response

In the response, notice that the internalID for the record added is returned.

```xml
<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#

```csharp
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("Please 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
// addresses 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 = State._california;
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(
//"The add() operation with document id " + _service.documentInfo.nsID + " was processed ");

// Process the response
if ( response.status.isSuccess )
{
    _out.info(
        "The following customer was added successfully:" +
        "key=" + ((RecordRef) response.baseRef).internalId +
        "entityId=" + customer.entityId +
        "companyName=" + customer.companyName +
        "email=" + customer.email +
        "statusKey=" + customer.entityStatus.internalId +
        "addressbookList[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 = "XY2 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
    
    if ( !response.status.isSuccess )
    {
        _out.info("The customer was not added:" );
        _out.error( getStatusDetails( response.status ) );
    }
    else
    {
        _out.info(
            "The following customer was added successfully:" +
            "key=" + ((RecordRef) response.baseRef).internalId +
            "entityId=" + customer.entityId +
            "companyName=" + customer.companyName +
            "email=" + customer.email +
            "statusKey=" + customer.entityStatus.internalId +
            "addressbookList[0].label=" + customer.addressbookList.addressbook[0].label
        );
    }
}
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("Please 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
    // addresses 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(State._california);
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()));
}

**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.
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 “Working with Hidden Fields” on page 31.

Request

The UpdateListRequest type is used for the request. It contains the following fields.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>record</td>
<td>Record</td>
<td>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.</td>
</tr>
</tbody>
</table>

Response

The UpdateListResponse type is used for the response. It contains the following fields.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>response</td>
<td>WriteResponse</td>
<td>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.</td>
</tr>
</tbody>
</table>

Faults

This operation can throw one of the following faults. See “Soap Fault Status Codes” on page 138 for more information on faults.

- InvalidSessionFault
- ExceededRateLimitFault
- ExceededMaxRecordsFault
- 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.

```xml
<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>
```
<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#**

```csharp
private void updateCustomerList()
{
    // This operation requires a valid session
    this.login( true );

    // Prompt for list of nsKeys and put in an array
    _out.write( "Enter 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( "The 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( "Customer[" + i + "]:");
        }
    }
```
Java

```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("Enter 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("The following customers were updated successfully:");
    for (int i = 0; i < responses.length; i++) {
```
The update operation is used to update an instance of a record in NetSuite. It is similar to the updateList operation but only allows one record to be updated 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, refer to “Working with Hidden Fields” on page 31. Also, custom fields can only be set to NULL by submitting the field in nullFieldList. For more information, refer to “CustomFieldList” on page 14 of the Records Guide.

In order 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.
Request

The UpdateRequest type is used for the request. It contains the following fields.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>record</td>
<td>Record</td>
<td>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.</td>
</tr>
</tbody>
</table>

Response

The UpdateResponse type is used for the response. It contains the following fields.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>response</td>
<td>WriteResponse</td>
<td>Contains details on the status of the operation and a reference to the updated record.</td>
</tr>
</tbody>
</table>

Faults

This operation can throw one of the following faults. See “Soap Fault Status Codes” on page 138 for more information on faults.

- InvalidSessionFault
- ExceededRateLimitFault
- ExceededMaxRecordsFault
- 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.

```xml
<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

```xml
<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>
```
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.readLine().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.defaultBillingSpecified = false;
    address.defaultBillingSpecified = true;
    address.label = "Billing Address";
    address.addr1 = "4765 Sunset Blvd";
    address.city = "San Mateo";
    address.state = State._california;
    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(State._california);
    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:
            | key=
            | entityId=
            | companyName=
            | email=
            | addressbookList[0].label=
        ");
    } else {
        _console.error(getStatusDetails(response.getStatus()));
    }
}

SuiteTalk Platform - Version 2.5.0
Updating Record Lists

When updating a list of records (or a list machine) within a business record you can NOT update a specific item in the list. Instead you must interact with the list as a whole. For details, refer to “Working with Lists” on page 29.

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.

Request

The DeleteListRequest type is used for the request. It contains the following fields.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>record</td>
<td>RecordRef</td>
<td>Must contain a reference to an existing instance of a record.</td>
</tr>
</tbody>
</table>

Response

The DeleteListResponse type is used for the response. It contains the following fields.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>response</td>
<td>WriteResponse</td>
<td>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.</td>
</tr>
</tbody>
</table>

Faults

This operation can throw one of the following faults. See “Soap Fault Status Codes” on page 138 for more information on faults.

- InvalidSessionFault
- ExceededRateLimitFault
- ExceededMaxRecordsFault
- 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>
<platformMgs:deleteList>
  <platformMgs:baseRef internalId="981" type="customer" xsi:type="platformCore:RecordRef"/>
  <platformMgs:baseRef internalId="982" type="customer" xsi:type="platformCore:RecordRef"/>
</platformMgs: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#

```csharp
private void deleteCustomerList()
{
    // This operation requires a valid session
    this.login( true );

    // Prompt for list of nsKeys and put in an array
    _out.write( "Enter 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( "Checking 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;
            ... (remaining steps of the code)
        }
```
System.Console.Out.WriteLine("\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++)
        {
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.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>record</td>
<td>RecordRef</td>
<td>Must contain a reference to an existing instance of a record.</td>
</tr>
</tbody>
</table>

**Response**

The DeleteResponse type is used for the response. It contains the following fields.
Faults

This operation can throw one of the following faults. See “Soap Fault Status Codes” on page 138 for more information on faults.

- InvalidSessionFault
- ExceededRateLimitFault
- ExceededMaxRecordsFault
- UnexpectedErrorFault

Sample Code

Sample Code

**SOAP Request**

In this example, a single customer record is deleted.

```xml
<soap:Body>
<platformMsgs:delete>
<platformMsgs:baseRef internalId="980" type="customer" xsi:type="platformCore:RecordRef"/>
</platformMsgs:delete>
</soap:Body>
```

**SOAP Response**

```xml
<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:ns2="urn:core_2_5.platform.webservices.netsuite.com"/>
</writeResponse>
</deleteResponse>
</soapenv:Body>
```

**C#**

```csharp
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.WriteLine( "Delete the record above? [Y/N]: " );
    string userResponse = _out.readLn().ToUpper();
    // Delete records
    if ( userResponse.Equals( "Y" ) )
    {
    }
}
```

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>response</td>
<td>WriteResponse</td>
<td>Contains details on the status of the delete operation and a reference to the deleted record.</td>
</tr>
</tbody>
</table>
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."");
   }
}

search

The search operation is used to execute a search on a specific record type based on a set of
search criteria that is specified through a custom search object type SearchRecord. You can
perform a search on a record type for fields specific for that record type only or you can
perform a joined search based on fields specific for that record and filtered by any search fields on an associated record.

**Working with Searches and Joined Searches**

The SearchRecord type consists of a Basic element which references all of the search fields specific to that record type and a list of join elements that reference fields available from other exposed records.

For example, the CustomerSearch record consists of one CustomerSearchBasic element and seven join search elements: parentCustomerJoin, subCustomerJoin, contactJoin, salesRepJoin, caseJoin, contactPrimaryJoin, and webSiteItemJoin. Note that all Basic search types physically reside in the platformCommon XSD.

```xml
<complexType name="CustomerSearch">
  <complexContent>
    <extension base="platformCore:SearchRecord">
      <sequence>
        <element name="basic" type="platformCommon:CustomerSearchBasic" minOccurs="0"/>
        <element name="parentCustomerJoin" type="platformCommon:CustomerSearchBasic" minOccurs="0"/>
        <element name="subCustomerJoin" type="platformCommon:CustomerSearchBasic" minOccurs="0"/>
        <element name="contactJoin" type="platformCommon:ContactSearchBasic" minOccurs="0"/>
        <element name="salesRepJoin" type="platformCommon:EmployeeSearchBasic" minOccurs="0"/>
        <element name="caseJoin" type="platformCommon:SupportCaseSearchBasic" minOccurs="0"/>
        <element name="contactPrimaryJoin" type="platformCommon:ContactSearchBasic" minOccurs="0"/>
        <element name="webSiteItemJoin" type="platformCommon:ItemSearchBasic" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

**Note:** Only fields from currently exposed records can be specified as filter criteria for a joined search request. You can identify which records can be used as filter criteria in the NetSuite UI by navigating to the Search Record for the desired record type, enabling **Use Advanced Search** and then scrolling through the Filter drop-down list. Joined search records are indicated by the Record Name followed by an ellipsis (...). Search fields from any of the records listed here that are also currently exposed can be included in the search criteria.
Available Joined Searches

The following table outlines the currently available joined searches for each Search Record type.

<table>
<thead>
<tr>
<th>Search Type</th>
<th>Available Joins</th>
</tr>
</thead>
<tbody>
<tr>
<td>SupportCaseSearch</td>
<td>contactJoin</td>
</tr>
<tr>
<td></td>
<td>customerJoin</td>
</tr>
<tr>
<td></td>
<td>employeeJoin</td>
</tr>
<tr>
<td></td>
<td>itemJoin</td>
</tr>
<tr>
<td>ContactSearch</td>
<td>caseJoin</td>
</tr>
<tr>
<td></td>
<td>customerJoin</td>
</tr>
<tr>
<td></td>
<td>customerPrimaryJoin</td>
</tr>
<tr>
<td></td>
<td>opportunityJoin</td>
</tr>
<tr>
<td>CustomerSearch</td>
<td>supportCaseJoin</td>
</tr>
<tr>
<td></td>
<td>contactJoin</td>
</tr>
<tr>
<td></td>
<td>contactPrimaryJoin</td>
</tr>
<tr>
<td></td>
<td>parentCustomerJoin</td>
</tr>
<tr>
<td></td>
<td>salesRepJoin</td>
</tr>
<tr>
<td></td>
<td>subCustomerJoin</td>
</tr>
<tr>
<td></td>
<td>webSitelItemJoin</td>
</tr>
<tr>
<td>EmployeeSearch</td>
<td>transactionJoin</td>
</tr>
<tr>
<td>CalendarEventSearch</td>
<td>opportunityJoin</td>
</tr>
<tr>
<td>ItemSearch</td>
<td>shopperJoin</td>
</tr>
</tbody>
</table>
Note: In order to simplify code, the extension base is platformCore:SearchRecord for both the basic search element types and the regular search element types. For example, both search(CustomerSearch object) and search(CustomerSearchBasic object) are valid search operations.

Maintaining Code

The current version of the WSDL has updated the naming of search objects in order to accommodate the joined search functionality. In order to maintain backwards compatibility the following options are available:

- NO code change is required if you do NOT want to use the latest endpoint
- If you DO want to use the latest endpoint but do not need to take the advantage of joined searches yet you can just modify the search record you used to instantiating from <Record>Search to <Record>SearchBasic. This should be sufficient.
- If you want to use the latest endpoint and use the joined searches you can instantiate the search record as the previous bullet describes and add any joined searches

Returning Associated Lists

Using the 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.
In the following example, all contacts associated with customers of internalId 1, 2 and 3 are returned.

**Java**
```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 Request**
```
<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>
```

**Search Preferences**

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

**Note:** For more information and code samples of how to set preferences at the request level, refer to “Setting Preferences at the Request Level” on page 24.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>bodyFieldsOnly</td>
<td>boolean</td>
<td>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 associated with the record are returned.</td>
</tr>
<tr>
<td>pageSize</td>
<td>int</td>
<td>See Pagination below.</td>
</tr>
</tbody>
</table>

**Important:** When performing searches on records, by default only the body fields are returned. To return field values for all lists on a record, you must set the bodyFieldsOnly element of the searchPreferences type to False. It is recommended that the default settings are used wherever list values are not necessary since this significantly improves performance.
You can perform quick and full searches as the same time. For example, to show a list of email folders, but only query for messages in the default folder, do a quick search to get folder names/ids, and a full search on all messages.

**Pagination**

Pagination is the concept of breaking up a large number of records that are part of a search result set into smaller chunks called pages. You then have the ability to retrieve these pages of records one at a time.

Pagination is mainly used to make processing such large result sets in a more manageable manner.

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` field in the `searchPreferences` 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. See page 109 for more information on `searchMore`.

**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 in essence means “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:

```xml
<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:

```java
// 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);

.Net code to generate this SOAP would be:

```csharp
// 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;
```

**Request**

The SearchRequest type is used for the request. It contains the following field.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>searchRecord</td>
<td>SearchRecord</td>
<td>The SearchRecord type is an abstract type. An instance of a type that extends SearchRecord must be used—such as CustomerSearchBasic or EventSearchBasic.</td>
</tr>
</tbody>
</table>

**Response**

The SearchResponse type is used for the response. It contains the following fields.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>Status</td>
<td>The status for this search. All applicable errors or warnings will be listed within this type.</td>
</tr>
<tr>
<td>totalRecords</td>
<td>xsd:int</td>
<td>The total number of records for this search. Depending on the pageSize value, some or all the records may be returned in this response</td>
</tr>
<tr>
<td>pageSize</td>
<td>xsd:int</td>
<td>The page size for this search.</td>
</tr>
<tr>
<td>totalPages</td>
<td>xsd:int</td>
<td>The total number of pages that are part of this search.</td>
</tr>
<tr>
<td>pageIndex</td>
<td>xsd:int</td>
<td>The page index for the current set of results.</td>
</tr>
<tr>
<td>recordList</td>
<td>Record[]</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Faults**

This operation can throw one of the following faults. See “Soap Fault Status Codes” on page 138 for more information on faults.

- InvalidSessionFault
- ExceededRateLimitFault
- ExceededMaxRecordsFault
- UnexpectedErrorFault
Sample Code

**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" on page 103).

```xml
<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.

```xml
<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:entityStatus>
                    <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#

```csharp
private void searchCustomer()
{
    // This operation requires a valid session
```
this.login( true );

_out.writeln( "Enter 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.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("No search criteria was specified. Searching for all records.");
} 
else 
{
    _out.info("\nSearching for customers with the following criteria: " +
    (entityId==null ? "" : ("entityID=" + custSearch.entityId.searchValue) + ",
    Operator=" + entityId.@operator.ToString()) +
    (stage==null ? "" : ("stage 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

class
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 = new SearchStringField();
    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("No 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()));
}

Searching for a Multi-Select Custom Field

In the following code sample, the results for the custom transaction field 'custcolcolumnname' are returned.
// transaction search by custom column field
TransactionSearch transactionSearch = new TransactionSearch();
SearchCustomFieldList searchCustomFieldList = new SearchCustomFieldList();
transactionSearch.setCustomFieldList(searchCustomFieldList);

// make the multiselect search
SearchMultiSelectCustomField searchMultiSelectCustomField = new SearchMultiSelectCustomField();
ListOrRecordRef listOrRecordRef = new ListOrRecordRef();
listOrRecordRef.setInternalId("1"); // the internal id of the custom list entry to match
listOrRecordRef.setName("yourcustomlistname"); // the name of the list
searchCustomFieldList.setCustomField(new SearchCustomField[]{searchMultiSelectCustomField});

// make the search expression
searchMultiSelectCustomField.setInternalId("custcolcolumnname"); // the name of the tx custom column
searchMultiSelectCustomField.setOperator(SearchMultiSelectFieldOperator.anyOf);
searchMultiSelectCustomField.setSearchValue(new ListOrRecordRef[]{listOrRecordRef});
SearchResult sr = _port.search(transactionSearch);

searchMore

The searchMore operation is used to retrieve more records after an initial search operation. 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:

```xml
<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">
        Susan</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.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>pageIndex</td>
<td>SearchRecord</td>
<td>An index that specifies which page in the search to return. If it is not provided, the next page is returned.</td>
</tr>
</tbody>
</table>

### Response

The SearchMoreResponse type is used for the response. It contains the following fields.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>Status</td>
<td>The status for this search. All applicable errors or warnings will be listed within this type.</td>
</tr>
<tr>
<td>searchPreferences</td>
<td>searchPreferences</td>
<td>Sets the Preferences for this search. The bodyFieldsOnly field of the searchPreference type 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 associated with the record are returned.</td>
</tr>
<tr>
<td>totalRecords</td>
<td>xsd:int</td>
<td>The total number of records for this search. Depending on the pageSize value, some or all the records may be returned in this response</td>
</tr>
</tbody>
</table>

---

**Employee Records in Database before Search**

- Susan Presley
- Susan Moore
- Susan Khan
- Susan Thompson
- Susan Kennedy
- Susan Carson
- Susan Green
- Susan Wong
- Susan Tracy
- Susan Black
- Susan Smith
- Susan Johnson

**Employee Records Returned in Search**

- Susan Presley
- Susan Moore
- Susan Khan
- Susan Thompson
- Susan Kennedy
- Susan Carson
- Susan Green
- Susan Wong
- Susan Tracy
- Susan Black

**Employee Records Returned in SearchMore**

- Susan Smith
- Susan Johnson

Susan Cook Employee Record Added
Faults

This operation can throw one of the following faults. See “Soap Fault Status Codes” on page 138 for more information on faults.

- InvalidSessionFault
- ExceededRateLimitFault
- ExceededMaxRecordsFault
- UnexpectedErrorFault

Sample Code

SOAP Request

In this example, page 2 of the search result set is requested.

```xml
<soap:Body>
  <platformMsgs:searchMore>
    <platformMsgs:pageIndex>2</platformMsgs:pageIndex>
  </platformMsgs:searchMore>
</soap:Body>
```

SOAP Response

```xml
<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>
          ...
          <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">
          ...
          <ns1:customFieldList>
            <customField internalId="custentity_map" xsi:type="StringCustomFieldRef">
                <value>http://maps.google.com</value>
            </customField>
          </ns1:customFieldList>
        </record>
      </recordList>
  </searchResult>
</searchMoreResponse>
```

### Element Name | XSD Type | Notes

| 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. |
| 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. |
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.readLine().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

```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):\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()));
        }
    }
}
```

**searchNext**

The `searchNext` operation is used to retrieve the next set of records after an initial search operation.

**Request**

The `searchNextRequest` type is used for the request. It does not contain any fields.

**Response**

The `searchMoreResponse` type is used for the response. It contains the following fields.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>Status</td>
<td>The status for this search. All applicable errors or warnings will be listed within this type.</td>
</tr>
<tr>
<td>totalRecords</td>
<td>xsd:int</td>
<td>The total number of records for this search. Depending on the <code>pageSize</code> value, some or all the records may be returned in this response</td>
</tr>
</tbody>
</table>
Faults

This operation can throw one of the following faults. See “Soap Fault Status Codes” on page 138 for more information on faults.

- InvalidSessionFault
- ExceededRateLimitFault
- ExceededMaxRecordsFault
- UnexpectedErrorFault

Sample Code

**SOAP Request**

```xml
<soap:Body>
  <platformMsgs:searchNext/>
</soap:Body>
```

**SOAP Response**

```xml
<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>
          ...
          ...
        </record>
        ...
        ...
      </recordList>
    </searchResult>
  </searchNextResponse>
</soapenv:Body>
```

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>pageSize</td>
<td>xsd:int</td>
<td>The page size for this search.</td>
</tr>
<tr>
<td>totalPages</td>
<td>xsd:int</td>
<td>The total number of pages that are part of this search.</td>
</tr>
<tr>
<td>pageIndex</td>
<td>xsd:int</td>
<td>The page index for the current set of results.</td>
</tr>
<tr>
<td>recordList</td>
<td>Record[]</td>
<td>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.</td>
</tr>
</tbody>
</table>
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.

**Request**

The getListRequest type is used for the request. It contains the following fields.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>recordRef</td>
<td>RecordRef</td>
<td>An array of recordRef objects that specify the ids of the records to be retrieved.</td>
</tr>
</tbody>
</table>

**Response**

The getListResponse type is used for the response. It contains the following fields.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>Status</td>
<td>The status for this operation. All applicable errors or warnings are listed within this type.</td>
</tr>
<tr>
<td>recordList</td>
<td>Record[]</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Faults**

This operation can throw one of the following faults. See “Soap Fault Status Codes” on page 138 for more information on faults.

- InvalidSessionFault
- ExceededRateLimitFault
- ExceededMaxRecordsFault
- UnexpectedErrorFault
Sample Code

SOAP Request

In the following example, 2 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#

```csharp
private int getCustomerList()
{
    // This operation requires a valid session
}
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.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>recordRef</td>
<td>RecordRef</td>
<td>A recordRef object that specifies the id of the record to be retrieved.</td>
</tr>
</tbody>
</table>

**Response**

The getListResponse type is used for the response. It contains the following fields.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>Status</td>
<td>The status for this operation. All applicable errors or warnings are listed within this type.</td>
</tr>
<tr>
<td>record</td>
<td>Record</td>
<td>A record that represents the specified id. The actual record returned needs to be a type that extends the abstract type Record.</td>
</tr>
</tbody>
</table>

**Faults**

This operation can throw one of the following faults. See “Soap Fault Status Codes” on page 138 for more information on faults.
• InvalidSessionFault
• ExceededRateLimitFault
• ExceededMaxRecordsFault
• 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.

```xml
<soap:Body>
<platformMsgs:get>
  <platformMsgs:baseRef internalId="983" type="customer" xsi:type="platformCore:RecordRef">
    <platformCore:name/>
  </platformMsgs:baseRef>
</platformMsgs:get>
</soap:Body>
```

SOAP Response

```xml
<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:entityStatus>
    <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#

```csharp
private void getCustomer()
{
    // This operation requires a valid session
    this.login( true );

    // Prompt for the nsKey
    _out.write("nsKey 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("nsKey=" + customer.internalId + ", ", 
    "entityId=" + 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)) + 
    (customer.isInactive) + 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("\nnnsKey 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("\nnnsKey="
        + customer.getInternalId()
        + ", ",
        "entityId="
        + customer.getEntityId()
        + (customer.getCompanyName() == null ? "" : 
        ("\ncompanyName=" + customer.getCompanyName()));
    }
}
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:

```xml
<!-- getAll Record Types -->
<simpleType name="GetAllRecordType">
    <restriction base="xsd:string">
        <enumeration value="customerCategory" />
        <enumeration value="contactCategory" />
        <enumeration value="priceLevel" />
        <enumeration value="winLossReason" />
        <enumeration value="term" />
        <enumeration value="noteType" />
        <enumeration value="paymentMethod" />
        <enumeration value="leadSource" />
    </restriction>
</simpleType>
```

**Note:** These are all records of the Other List type.

**Request**

The getAllRequest type is used for the request. It contains the following fields.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>recordType</td>
<td>GetAllRecordType</td>
<td>Specify the record type.</td>
</tr>
</tbody>
</table>

**Response**

The getList response type is used for the response. It contains the following fields.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>Status</td>
<td>The status for this operation. All applicable errors or warnings are listed within this type.</td>
</tr>
<tr>
<td>recordList</td>
<td>Record[]</td>
<td>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.</td>
</tr>
</tbody>
</table>
Faults

This operation can throw one of the following faults. See “Soap Fault Status Codes” on page 138 for more information on faults.

- InvalidSessionFault
- ExceededRateLimitFault
- ExceededMaxRecordsFault
- UnexpectedErrorFault

Sample Code

SOAP Request

In this example, all customer category records for the currently logged in account are retrieved.

```xml
<soap:Body>
<platformMsgs:getAll>
  <platformMsgs:record recordType="customerCategory"/>
</platformMsgs:getAll>
</soap:Body>
```

SOAP Response

```xml
<soapenv:Body>
<getAllResponse xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
  <getAllResult xmlns="urn:core_2_5.platform.webservices.netsuite.com">
    <status isSuccess="true"/>
    <totalRecords>2</totalRecords>
    <recordList>
      <record internalId="1" xsi:type="ns1:CustomerCategory"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:ns1="urn:accounting_2_5.lists.webservices.netsuite.com">
        <ns1:name>Corporate</ns1:name>
        <ns1:inactive>false</ns1:inactive>
      </record>
      <record internalId="2" xsi:type="ns2:CustomerCategory"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:ns2="urn:accounting_2_5.lists.webservices.netsuite.com">
        <ns2:name>Individual</ns2:name>
        <ns2:inactive>false</ns2:inactive>
      </record>
    </recordList>
  </getAllResult>
</getAllResponse>
</soapenv:Body>
```

C#

```csharp
private void getAll()
{
    // This operation requires a valid session
    this.login( true );

    // Instantiate GetAllRecord and set the record type
    GetAllRecord record = new GetAllRecord();
    record.recordTypeSpecified = true;

    _out.writeLn( "\nSelect the list type to be retrieved" );
    _out.writeLn("1) Customer Categories");
}
_out.writeLn("2) Contact Categories");
_out.writeLn("3) Price Levels");
_out.writeLn("4) Win Loss Reasons");
_out.writeLn("5) Terms");
_out.writeLn("6) Note Types");
_out.writeLn("7) Payment Methods");
_out.writeLn("8) Lead Sources");
_out.write("\nSelection: ");
String strMyChoice = _out.readLn().ToUpper();
_out.write("\n");
int choice = Convert.ToInt32(strMyChoice);

switch (choice)
{
case 1:
    record.recordType = GetAllRecordType.customerCategory;
    break;
case 2:
    record.recordType = GetAllRecordType.contactCategory;
    break;
case 3:
    record.recordType = GetAllRecordType.priceLevel;
    break;
case 4:
    record.recordType = GetAllRecordType.winLossReason;
    break;
case 5:
    record.recordType = GetAllRecordType.term;
    break;
case 6:
    record.recordType = GetAllRecordType.noteType;
    break;
case 7:
    record.recordType = GetAllRecordType.paymentMethod;
    break;
case 8:
    record.recordType = GetAllRecordType.leadSource;
    break;
}

// Invoke getAll() operation
GetAllResult result = _service.getAll(record);
Record[] records = result.recordList;

if (result.status.isSuccess)
{
    _out.info("\nThe requested list for " + record.recordType.ToString() + 
    " returned " + result.totalRecords + " records:" );
    int numRecords = 0;
    for (int i=0; i<records.Length; i++)
    {
        numRecords++;
        switch (choice)
        {
        case 1:
            CustomerCategory custCategory = (CustomerCategory)records[i];
            _out.info("key=" + custCategory.internalId + ", name=" + 
custCategory.name);
            break;
        case 2:
            ContactCategory contCategory = (ContactCategory)records[i];
            _out.info("key=" + contCategory.internalId + ", name=" + 
contCategory.name);
            break;
        // Add cases for other record types as needed...
        }
public void getAll() throws RemoteException, ExceededUsageLimitFault, UnexpectedErrorFault, InvalidSessionFault, ExceededRecordCountFault {
    // This operation requires a valid session
    this.login(true);

    // Instantiate GetAllRecord and set the record type
    GetAllRecord record = new GetAllRecord();

    _console.writeLn("Select the list type to be retrieved");
    _console.writeLn("1) Customer Categories");
    _console.writeLn("2) Contact Categories");
    _console.writeLn("3) Price Levels");
    _console.writeLn("4) Win Loss Reasons");
    _console.writeLn("5) Terms");
    _console.writeLn("6) Note Types");
    _console.writeLn("7) Payment Methods");
    _console.writeLn("8) Lead Sources");
    _console.write("\nSelection: ");
    String strMyChoice = _console.readLine().toUpperCase();
    _console.writeLn();

    int choice = Integer.parseInt(strMyChoice);

    switch (choice) {
    case 1:
        record.setRecordType(GetAllRecordType.customerCategory);
        break;
    case 2:
        record.setRecordType(GetAllRecordType.contactCategory);
        break;
    case 3:
        PriceLevel priceLevel = (PriceLevel) records[i];
        _out.info("key="+priceLevel.internalId+"., name="+priceLevel.name);
        break;
    case 4:
        WinLossReason winLossReason = (WinLossReason) records[i];
        _out.info("key="+winLossReason.internalId+"., name="+winLossReason.name);
        break;
    case 5:
        Term term = (Term) records[i];
        _out.info("key="+term.internalId+"., name="+term.name);
        break;
    case 6:
        NoteType noteType = (NoteType) records[i];
        _out.info("key="+noteType.internalId+"., name="+noteType.name);
        break;
    case 7:
        PaymentMethod paymentMethod = (PaymentMethod) records[i];
        _out.info("key="+paymentMethod.internalId+"., name="+paymentMethod.name);
        break;
    case 8:
        LeadSource leadSource = (LeadSource) records[i];
        _out.info("key="+leadSource.internalId+"., name="+leadSource.name);
        break;
    }
}
break;
case 3:
    record.setRecordType(GetAllRecordType.priceLevel);
    break;
case 4:
    record.setRecordType(GetAllRecordType.winLossReason);
    break;
case 5:
    record.setRecordType(GetAllRecordType.term);
    break;
case 6:
    record.setRecordType(GetAllRecordType.noteType);
    break;
case 7:
    record.setRecordType(GetAllRecordType.paymentMethod);
    break;
case 8:
    record.setRecordType(GetAllRecordType.leadSource);
    break;
}

// Invoke getAll() operation
GetAllResult result = _port.getAll(record);

// Process results
RecordList recordList = result.getRecordList();
Record[] records = recordList.getRecord();
if (result.getStatus().isIsSuccess()) {
    _console.info("The requested list for ");
    _console.info(record.getRecordType().toString() + " returned ");
    _console.info(result.getTotalRecords() + " records:");
    int numRecords = 0;
    for (int i = 0; i < records.length; i++) {
        numRecords++;
        switch (choice) {
            case 1:
                CustomerCategory custCategory = (CustomerCategory) records[i];
                _console.info("key=" + custCategory.getInternalId() + ", name=" + custCategory.getName());
                break;
            case 2:
                ContactCategory contCategory = (ContactCategory) records[i];
                _console.info("key=" + contCategory.getInternalId() + ", name=" + contCategory.getName());
                break;
            case 3:
                PriceLevel priceLevel = (PriceLevel) records[i];
                _console.info("key=" + priceLevel.getInternalId() + ", name=" + priceLevel.getName());
                break;
            case 4:
                WinLossReason winLossReason = (WinLossReason) records[i];
                _console.info("key=" + winLossReason.getInternalId() + ", name=" + winLossReason.getName());
                break;
            case 5:
                Term term = (Term) records[i];
                _console.info("key=" + term.getInternalId() + ", name=" + term.getName());
                break;
            case 6:
                NoteType noteType = (NoteType) records[i];
                _console.info("key=" + noteType.getInternalId() + ", name=" + noteType.getName());
                break;
        }
    }
}
getSelectValue

The getSelectValue operation is used 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.

Important: Slaving that would normally occur in the UI is NOT performed when accessing this list via Web services. The list of values that returned reflects the state of the list as it would be on initial load of a given form. Also, the type is null if the record type is not yet exposed in the platform, or if the backend is unable to determine the type.

Request

The getSelectValueRequest type is used for the request. It contains the following field.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>fieldName</td>
<td>Enum</td>
<td>This enum contains a fully qualified reference to exposed multiselect fields that can be called using this operation. For example, the value that corresponds to attendees on an event record is <code>calendarEvent.attendeeList.attendee</code> — where calendarEvent is the record type, attendeeList is the machine on the record, and attendee is the desired multi-select field.</td>
</tr>
</tbody>
</table>

You can also set filter criteria for a given request to specify that a subset of choices be returned. The filter takes a string value based on the choices available in the specified drop-down list.

For example, in the following SOAP request, all values from the `employee_rolesList_selectedRole` drop-down list that also contain 'Administrator' will be returned.

```xml
<soapenv:Body>
<getSelectValue xmlns="urn:messages_1_3.platform.webservices.netsuite.com">
  <fieldName fieldType="employee_rolesList_selectedRole">
    <ns1:searchCriteria xmlns:ns1="urn:core_1_3.platform.webservices.netsuite.com">Administrator</ns1:searchCriteria>
  </fieldName>
</getSelectValue>
</soapenv:Body>
```
Response

The getSelectValueResponse type is used for the response. It contains the following fields.

**Note:** A maximum of 10,000 values are returned. If the actual number of values available for the requested type exceeds 10,000 a warning is returned.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>Status</td>
<td>The status for this operation. All applicable errors or warnings are listed within this type.</td>
</tr>
<tr>
<td>recordList</td>
<td>Record[]</td>
<td>A list of recordRefs that represent valid values available for the current field. (includes the type and name)</td>
</tr>
<tr>
<td>totalRecords</td>
<td>int</td>
<td>number of choices</td>
</tr>
</tbody>
</table>

Faults

This operation can throw one of the following faults. See “Soap Fault Status Codes” on page 138 for more information on faults.

- InvalidSessionFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- UnexpectedErrorFault

Sample Code

**SOAP Request**

```xml
<soapenv:Body>
  <getSelectValue xmlns="urn:messages_2_5.platform.webservices.netsuite.com">
    <fieldName fieldType="customer_customForm"/>
  </getSelectValue>
</soapenv:Body>
```

**SOAP Response**

```xml
<soapenv:Body>
  <getSelectValueResponse xmlns="urn:messages_1_2.platform.webservices.netsuite.com">
    <getSelectValueResult xmlns="urn:core_1_2.platform.webservices.netsuite.com">
      <status isSuccess="true"/>
      <totalRecords>4</totalRecords>
      <recordRefList>
        <recordRef internalId="5">
          <name>Customer Form - Automation</name>
        </recordRef>
        <recordRef internalId="6">
          <name>Customer Form - Custom Code</name>
        </recordRef>
        <recordRef internalId="-2">
          <name>Standard Customer Form</name>
        </recordRef>
        <recordRef internalId="-5">
          <name>Standard Pop-up Customer Form</name>
        </recordRef>
      </recordRefList>
    </getSelectValueResult>
  </getSelectValueResponse>
</soapenv:Body>
```
Java

GetSelectValueResult gr = port.getSelectValue(new GetSelectValueField(GetSelectValueType.fromValue("customer_customForm")));

Or

GetSelectValueResult gr = port.getSelectValue(new GetSelectValueField(GetSelectValueType.customer_category));

**GetSelectValue Types**

For a complete list of all of the values currently supported by the getSelectValue operation, refer to the GetSelectValueType enumerations in the coreTypes XSD.

**getCustomization**

Because any record in NetSuite can be fully customized to suit the needs of a given business, it is critical to consider these customizations when developing generic web services applications. Use the getCustomization operation to dynamically retrieve and manage the metadata for Custom Fields, Lists, and Record Types in order to handle some of the custom business logic included at the record level.

Following are the custom objects currently supported by the getCustomization operation. These are enumerated in the coreTypes XSD.

- crmCustomField
- customList
- customRecordType
- entityCustomField
- itemCustomField
- itemOptionCustomField
- otherCustomField
- transactionBodyCustomField
- transactionColumnCustomField

**Note:** Normally, you can NOT add or update the internalId of an object. Customization is an exception to this rule. You can specify the ID on add, but can NOT update the ID thereafter. InternalIds for custom objects can be set to any unique alphanumeric string up to 30 characters long. This string must not include any spaces but can include underscores ("_").

**Request**

The getCustomizationRequest type is used for this request. It contains the following fields:
Response

The getCustomizationResult type is used for the Response. It contains the following fields:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>Status</td>
<td>The status for this operation. All applicable errors or warnings are listed within this type.</td>
</tr>
<tr>
<td>totalRecords</td>
<td>int</td>
<td></td>
</tr>
<tr>
<td>recordList</td>
<td>Record[]</td>
<td>A list of records that correspond to the specified customization type.</td>
</tr>
</tbody>
</table>

Sample Code

**SOAP Request**

```xml
<soap:Body>
  <platformMsgs:getCustomization>
    <platformMsgs:customizationType getCustomizationType="crmCustomField"/>
  </platformMsgs:getCustomization>
</soap:Body>
</soap:Envelope>
```

**SOAP Response**

```xml
<soapenv:Body>
  <getCustomizationResponse xmlns="urn:messages_2_0.platform.webservices.netsuite.com">
    <getCustomizationResult xmlns="urn:core_2_5.platform.webservices.netsuite.com">
      <status isSuccess="true"/>
      <totalRecords>1</totalRecords>
      <recordList>
        <record internalId="CUSTEVENT1" xsi:type="ns1:CrmCustomField"
            xmlns:ns1="urn:customization_2_5.setup.webservices.netsuite.com">
          <ns1:label>test</ns1:label>
          <ns1:fieldType>_freeFormText</ns1:fieldType>
          <ns1:storeId>true</ns1:storeId>
          <ns1:showInList>false</ns1:showInList>
          <ns1:isParent>false</ns1:isParent>
          <ns1:displayType>_normal</ns1:displayType>
          <ns1:isMandatory>false</ns1:isMandatory>
          <ns1:defaultChecked>false</ns1:defaultChecked>
          <ns1:isFormula>false</ns1:isFormula>
          <ns1:appliesToTask>false</ns1:appliesToTask>
          <ns1:appliesToPhoneCall>false</ns1:appliesToPhoneCall>
          <ns1:appliesToEvent>false</ns1:appliesToEvent>
          <ns1:appliesToCase>false</ns1:appliesToCase>
          <ns1:appliesToCampaign>false</ns1:appliesToCampaign>
          <ns1:appliesPerKeyword>false</ns1:appliesPerKeyword>
          <ns1:availableExternally>false</ns1:availableExternally>
        </record>
      </recordList>
    </getCustomizationResult>
  </getCustomizationResponse>
</soapenv:Body>
```
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.

### Request

The `GetItemAvailabilityRequest` type is used for the request.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>itemAvailabilityFilter</td>
<td>ItemAvailability</td>
<td>You can filter the returned <code>itemAvailability</code> using this filter.</td>
</tr>
</tbody>
</table>

### ItemAvailabilityFilter

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>item</td>
<td>RecordRef</td>
<td>References an exiting item record in NetSuite.</td>
</tr>
<tr>
<td>lastQtyAvailableChange</td>
<td>dateTime</td>
<td>If set, only items with quantity available changes recorded as of the specified date are returned.</td>
</tr>
</tbody>
</table>

### Response

The `GetItemAvailabilityResult` type is used for the response.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>Status</td>
<td>The status for this operation. All applicable errors or warnings are listed within this type.</td>
</tr>
<tr>
<td>itemAvailabilityList</td>
<td>List</td>
<td>Returns a list of available items.</td>
</tr>
</tbody>
</table>

### ItemAvailabilityList

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>item</td>
<td>RecordRef</td>
<td>References an existing item record.</td>
</tr>
<tr>
<td>lastQtyAvailableChange</td>
<td>dateTime</td>
<td>If set, only items with quantity available changes recorded as of this date are returned.</td>
</tr>
<tr>
<td>locationId</td>
<td>RecordRef</td>
<td>References a location in a user defined list at Lists &gt; Accounting &gt; Locations.</td>
</tr>
<tr>
<td>quantityOnHand</td>
<td>double</td>
<td>The number of units of an item in stock.</td>
</tr>
</tbody>
</table>
### ItemAvailabilityFilter

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>item</td>
<td>RecordRefList</td>
<td></td>
</tr>
<tr>
<td>lastQtyAvailableChange</td>
<td>dateTime</td>
<td></td>
</tr>
</tbody>
</table>

### Faults

This operation can throw one of the following faults. See “Soap Fault Status Codes” on page 138 for more information on faults.

- InvalidSessionFault
- ExceededRequestLimitFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- UnexpectedErrorFault

### Sample Code

**SOAP Request**

```xml
<soap:Body>
  <platformMsgs:getItemAvailability>
    <platformMsgs:itemAvailabilityFilter>
      <platformCore:item>
        <platformCore:recordRef internalId="390" type="inventoryItem">
          <platformCore:name/>
        </platformCore:recordRef>
      </platformCore:item>
    </platformMsgs:itemAvailabilityFilter>
  </platformMsgs:getItemAvailability>
</soap:Body>
```
**SOAP Response**

```xml
<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>
  <quantityOnHand>20.0</quantityOnHand>
</itemAvailability>
</itemAvailabilityList>
</getItemAvailabilityResult>
</getItemAvailabilityResponse>
</soapenv:Body>
```

**attachTo / attachedRecord**

The attachTo and attachedRecord operations can be used to attach a contact to a customer record. This operation should also be used to update the read only customerContact list on the Customer record.

**Important:** The attach operations can be used to attach Contacts to Customer Records only. Attempting to attach contacts to any other record type will result in a user error.

The contacts' role can also be specified during the request. Contact Roles are roles available in a user defined list at Accounting > Lists > Other Lists. This list has been exposed as ContactRole in accounting.xsd.

**changePasswordOrEmail**

You can now change a users email or password via the changePasswordOrEmail operation.

**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 deleted via the UI. This operation is useful in order to easily synchronize information in a client application to the data currently in NetSuite.
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 internalID and record type. In cases where internalIDs are NOT unique within a record type, the deletions are NOT tracked and therefore cannot be retrieved using the getDeleted operation. Examples include items, bins, and serial numbers. For a complete list of record types, refer to the DeletedRecordType enumeration in the coreTypes xsd.

**Important:** Entity records are unique in that you may have an entity that belongs to several different subtypes. For example, a partner that is also a contact. If the partner record is deleted, the contact record remains intact.

**Request**

The GetDeletedRequest type is used for the request.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>getDeletedFilter</td>
<td>GetDeletedFilter</td>
<td>See GetDeletedFilter.</td>
</tr>
</tbody>
</table>

**GetDeletedFilter**

The GetDeletedFilter allows you to retrieve data based on the date or record type of the deleted record.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>deletedDate</td>
<td>SearchDateField</td>
<td>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.</td>
</tr>
<tr>
<td>type</td>
<td>SearchEnumMultiSelectField</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Response**

The DeletedRecord type is used for the response.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>XSD Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>deletedDate</td>
<td>dateTime</td>
<td>The status for this operation. All applicable errors or warnings are listed within this type.</td>
</tr>
<tr>
<td>record</td>
<td>BaseRef</td>
<td>Returns a list of available items.</td>
</tr>
</tbody>
</table>
**Faults**

This operation can throw one of the following faults. See “Soap Fault Status Codes” on page 138 for more information on faults.

- InvalidSessionFault
- ExceededUsageLimitFault
- ExceededRecordCountFault
- UnexpectedErrorFault

**Sample Code**

**SOAP Request**

```xml
<soapenv:Body>
<platformMsgs:getDeleted xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
 xmlns:platformCoreTyp="urn:types.core_2_5.platform.webservices.netsuite.com"
 xmlns:platformCore="urn:core_2_5.platform.webservices.netsuite.com"
 xmlns:platformMsgs="urn:messages_2_5.platform.webservices.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>
```

**initialize**

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.

The following table outlines all the transaction types that can be used with the initialize operation and the valid reference types they can use.

<table>
<thead>
<tr>
<th>Transaction Type</th>
<th>Initialize Reference</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cash Refund</td>
<td>Cash Sale</td>
<td>Sets createdFrom and populates all the lines</td>
</tr>
<tr>
<td>Cash Sale</td>
<td>Customer</td>
<td>Populates Billable tabs</td>
</tr>
<tr>
<td>Cash Sale</td>
<td>Estimate</td>
<td></td>
</tr>
<tr>
<td>Cash Sale</td>
<td>Opportunity</td>
<td></td>
</tr>
</tbody>
</table>
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 can NOT be set.
Sample Code

SOAP Request

```xml
<soapenv:Body>
xmlns:platformCoreTyp="urn:types.core_2_5.platform.webservices.netsuite.com"
xmlns:platformCore="urn:core_2_5.platform.webservices.netsuite.com"
xmlns:platformMsgs="urn:messages_2_5.platform.webservices.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>
```
### Handling Errors

#### Soap Faults for each Operation

The following table lists the SOAP faults that can be thrown for each supported operation.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Faults</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login</td>
<td>ACCT_TEMP_UNAVAILABLE, EMAIL_ADDRS_REQD, INVALID_ACCT, INVALID_LOGIN_CREDENTIALS, INVALID_VERSION, PSWD_REQD, UNEXPECTED_ERROR, WS_CONCUR_SESSION_DISALLWD, WS_FEATUREREQD, WS_PERMISSION_REQD</td>
</tr>
<tr>
<td>MapSso</td>
<td>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</td>
</tr>
<tr>
<td>Logout</td>
<td>INVALID_VERSION, UNEXPECTED_ERROR, WS_CONCUR_SESSION_DISALLWD, SESSION_TIMED_OUT</td>
</tr>
</tbody>
</table>
### 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
- 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
Soap Fault Status Codes

The following table lists and defines the possible SOAP fault types and corresponding codes. For a complete description of faults and how they differ from Errors and Warnings, refer to “Handling Errors” on page 48.

<table>
<thead>
<tr>
<th>Fault Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InsufficientPermissionFault</td>
<td>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 the client (user) has more than one role, they may need to login again supplying a different role with more permissions.</td>
</tr>
<tr>
<td>InvalidAccountFault</td>
<td>This fault is thrown when the client attempts to login with an invalid account id.</td>
</tr>
<tr>
<td>InvalidPartnerCredentials</td>
<td>Partner ID or password submitted with this request is invalid.</td>
</tr>
<tr>
<td>InvalidRequestIP</td>
<td>Originating IP is not one of the registered IPs from which this Partner request may come.</td>
</tr>
<tr>
<td>InvalidSessionFault</td>
<td>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, refer to “Session Management” on page 36.</td>
</tr>
<tr>
<td>InvalidVersionFault</td>
<td>This fault is thrown in the event that the request message contains an unsupported version of the schema.</td>
</tr>
<tr>
<td>ExceededRecordCountFault</td>
<td>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” on page 10.</td>
</tr>
<tr>
<td>ExceededRequestLimitFault</td>
<td>This fault is thrown if the allowed number of concurrent requests is exceeded. For more information, see “Understanding Web Services Governance” on page 10.</td>
</tr>
<tr>
<td>UnexpectedErrorFault</td>
<td>This fault is thrown in the event of an occurrence of an unexpected exception.</td>
</tr>
</tbody>
</table>
### Error Status Codes

The following table lists possible 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.

<table>
<thead>
<tr>
<th>Error Code Returned</th>
<th>Long Description or Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCT_DISABLED</td>
<td>YOUR_ACCOUNT_HAS_BEEN_INACTIVATED_BY_AN_ADMINISTRATOR;</td>
</tr>
<tr>
<td>ACCT_DISABLED</td>
<td>ACCT_DISABLED;</td>
</tr>
<tr>
<td>ACCT_DISABLED</td>
<td>THIS_ACCOUNT_HAS_BEEN_DISABLED;</td>
</tr>
<tr>
<td>ACCT_DISABLED</td>
<td>THIS_COMPANY_HAS_BEEN_DISABLED_PLEASE_CONTACT_A_HREFMAIL_TO1_SUPPORTA_TOREENABLE_THIS_COMPANY;</td>
</tr>
<tr>
<td>ACCT_NUMS_REQD_OR_DONT_MATCH</td>
<td>MISSING_ACCT__OR_ACCT_NUMBERS_DONT.Match;</td>
</tr>
<tr>
<td>ACCT_TEMP_DISABLED</td>
<td>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_PASSWORDPLEASE_CONTACT_CUSTOMER_SUPPORT;</td>
</tr>
<tr>
<td>ACCT_TEMP_UNAVAILABLE</td>
<td>TEMPORARILY_UNAVAILABLE;</td>
</tr>
<tr>
<td>ACCT_TEMP_UNAVAILABLE</td>
<td>WS_ACCOUNT_IS_CURRENTLY_UNAVAILABLE;</td>
</tr>
<tr>
<td>ACCT_TEMP_UNAVAILABLE</td>
<td>WE_ARE_CURRENTLY_PERFORMING_MAINTENANCE_ON_OUR_SYSTEMPLEASE_TRY_AGAIN_SOON;</td>
</tr>
<tr>
<td>ACCT_TEMP_UNAVAILABLE</td>
<td>YOUR_ACCOUNT_IS_DISABLED_FOR_1_MORE_MINUTES_DUE_TO_2_CONSECUTIVE_FAILED_LOGIN_ATTEMPTS;</td>
</tr>
<tr>
<td>ACCT_TEMP_UNAVAILABLE</td>
<td>YOUR_ACCOUNT_IS_NOT_YET_READY_FOR_YOU_TO_LOG_INPLEASE_WAIT_AND_TRY_AGAIN;</td>
</tr>
<tr>
<td>ACCT_TEMP_UNAVAILABLE</td>
<td>YOUR_COMPANY_DATABASE_IS_OFFLINE;</td>
</tr>
<tr>
<td>ACCT_TEMP_UNAVAILABLE</td>
<td>YOUR_DATA_IS_STILL BEING_LOADEDPLEASE.Try_AGAIN_LATER_CONTACT_A_HREFAPPCRM.SUPPORTNLTBUGFORMNLTYPEBUGSPF31PROFESSIONALSERVICESA_IF_YOU_HAVE_QUESTIONS;</td>
</tr>
<tr>
<td>ACCT_REQD</td>
<td>ATTEMPTING_TO_ADJUST_PROVISIONING_FOR_A_CUSTOMER WITHOUT_AN_EXISTING_ACCOUNT;</td>
</tr>
<tr>
<td>ACTIVE_TRANS_EXIST</td>
<td>THERE_ARE_ACTIVE_DIRECT_DEPOSIT_TRANSACTIONS_FOR_THIS_PAYCHECK;</td>
</tr>
<tr>
<td>ADMIN_ACCESS_REQ</td>
<td>AT_LEAST_ONE_ACTIVE_ADMINISTRATOR_FOR_EACH_ACCOUNT_MUST_HAVE_ACCESS;</td>
</tr>
<tr>
<td>ADMIN_ACCESS_REQ</td>
<td>AT_LEAST_ONE_ACTIVE_ADMINISTRATOR_FOR_THIS_ACCOUNT_MUST_HAVE_ACCESS;</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ADMIN_ACCESS_REQD</td>
<td>ONLY_ADMINISTRATORS_MAY_ENTER_A_MEMORIZED_TRANSACTION_IN_A_CLOSED_PERIOD;</td>
</tr>
<tr>
<td>ALL_DATA_DELETE_REQD</td>
<td>YOU_MUST_FIRST_DELETE_ALL_THE_DATA_IN_YOUR_ACCOUNT BEFORE_PERFORMING_THIS_ACTION;</td>
</tr>
<tr>
<td>ALL_MTRX_SUBITMES_OPTNS_REQD</td>
<td>SOME_MATRIX_SUBITEMS_EXIST_THAT_ARENT_INCLUDED_IN_THE_OPTIONS_YOU_JUST_SPECIFIED_ON_THE_MATRIX_TAB PLEASE_MAKE_SURE_THE_OPTIONS_YOU_SELECT.Include_ALL_EXISTING_SUBITEMS;</td>
</tr>
<tr>
<td>ALREADY_IN_INVNT</td>
<td>THE_FOLLOWING_1_NUMBER_IS_ALREADY_IN_INVENTORY_2;</td>
</tr>
<tr>
<td>ALREADY_IN_INVNT</td>
<td>THE_FOLLOWING_1_NUMBERS_ARE_ALREADY_IN_INVENTORY_2;</td>
</tr>
<tr>
<td>APP_DEPRECATION_WARN</td>
<td><strong>The record</strong> <code>&lt;record name&gt;</code> <strong>is deprecated.</strong> A new required field, <code>&lt;field name&gt;</code>, has been added to this record. We are setting this field to the default value of <code>&lt;default value&gt;</code>. The deprecation period will end on <code>&lt;end date&gt;</code>. The current version of your client will be obsolete at the end of the deprecation period, please upgrade to the latest version.</td>
</tr>
<tr>
<td>APP_DEPRECATION_WARN</td>
<td>The record <code>&lt;record name&gt;</code> is deprecated. The <code>&lt;field name&gt;</code> field is now required. We are setting this field to the default value of <code>&lt;default value&gt;</code>. The deprecation period will end on <code>&lt;end date&gt;</code>. The current version of your client will be obsolete at the end of the deprecation period, please upgrade to the latest version.</td>
</tr>
<tr>
<td>APP_DEPRECATION_WARN</td>
<td>The record <code>&lt;record name&gt;</code> is deprecated. The type of the <code>&lt;field name&gt;</code> field has changed to <code>&lt;new field type&gt;</code>. We are mapping the value of this field to <code>&lt;mapped value&gt;</code>. The deprecation period will end on <code>&lt;end date&gt;</code>. The current version of your client will be obsolete at the end of the deprecation period, please upgrade to the latest version.</td>
</tr>
<tr>
<td>APP_DEPRECATION_WARN</td>
<td>The record <code>&lt;record name&gt;</code> is deprecated. The <code>&lt;field name&gt;</code> field has been renamed to <code>&lt;new field name&gt;</code>. We are mapping the old name to the new name. The deprecation period will end on <code>&lt;end date&gt;</code>. The current version of your client will be obsolete at the end of the deprecation period, please upgrade to the latest version.</td>
</tr>
<tr>
<td>APP_DEPRECATION_WARN</td>
<td>The record <code>&lt;record name&gt;</code> is deprecated. The <code>&lt;field name&gt;</code> field has been removed. The deprecation period will end on <code>&lt;end date&gt;</code>. The current version of your client will be obsolete at the end of the deprecation period, please upgrade to the latest version.</td>
</tr>
<tr>
<td>APP_DEPRECATION_WARN</td>
<td>The <code>&lt;record name&gt;</code> record is deprecated. It has been renamed to <code>&lt;new record name&gt;</code>. The deprecation period will end on <code>&lt;end date&gt;</code>. The current version of your client will be obsolete at the end of the deprecation period, please upgrade to the latest version.</td>
</tr>
<tr>
<td>APP_DEPRECATION_WARN</td>
<td>The <code>&lt;record name&gt;</code> record is deprecated. It has been removed. The deprecation period will end on <code>&lt;end date&gt;</code>. The current version of your client will be obsolete at the end of the deprecation period, please upgrade to the latest version.</td>
</tr>
<tr>
<td>AREA_CODE_REQD</td>
<td>PLEASE_INCLUDE_AN_AREA_CODE_WITH_THE_PHONE_NUMBER;</td>
</tr>
<tr>
<td>ATTACH_SIZE_EXCEEDED</td>
<td>YOU_HAVE_EXCEEDED_THE_MAXIMUM_ATTACHMENTS_SIZE_OF_10_MB PLEASE_REMOVE_ONE_OR_MORE_ATTACHMENTS_AND_TRY_AGAIN;</td>
</tr>
<tr>
<td>AUTO_NUM_UPDATE_DISALLOWED</td>
<td>WE_CURRENTLY_DO_NOT_SUPPORT_AN_AUTOMATIC_NUMBERING_UPDATE_DATE_OF_MORE_THAN_1_2_RECORDS;</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BASE_CRNCY_REQD</td>
<td>YOU_MAY_NOT_DELETE_YOU_BASE_CURRENCY;</td>
</tr>
<tr>
<td>BILL_PAY_STATUS_UNAVAILABLE</td>
<td>VIEW_ONLINE_BILL_PAY_STATUS_INFORMATION_IS_CURRENTLY_NOT_AVAILABLE_PLEASE_TRY_AGAIN_IN_A_FEW_MINUTES;</td>
</tr>
<tr>
<td>BILL_PAY_STATUS_UNAVAILABLE</td>
<td>VIEW_ONLINE_BILL_PAY_STATUS_IS_NOTAVAILABLE_UNTIL_YOUR_BILL_PAY_REGISTRATION_IS_COMPLETE;</td>
</tr>
<tr>
<td>BILL_PMTS_MADE_FROM_ACCT_ONLY</td>
<td>YOUR_PAYMENT_HAS_BEEN.Recorded_BUT_ONLINE_BILL_PAYMENTS_CAN_ONLY_BE_MADE_FROM_THE_ACCOUNT;</td>
</tr>
<tr>
<td>BILLING_ISSUES</td>
<td>YOUR_ACCOUNT_HAS_BEEN_LOCKED_DUE_TO_BILLING_ISSUES__YOU_MUST_CALL_YOUR_SALES_REPRESENTATIVE_AT_1800SMALLBIZ_FOR_FURTHER_ASSISTANCE;</td>
</tr>
<tr>
<td>BILLING_ISSUES</td>
<td>YOUR_ACCOUNT_HAS_NOT_BEEN_FULLY_PAID_FOR_PLEASE_LOG_IN_TO_YOUR_ACCOUNT_AND_FOLLOW_THE_BILLING_PROCESS_OR_CONTACT_YOUR_ACCOUNT_MANAGER;</td>
</tr>
<tr>
<td>CALENDAR_PREFS_REQD</td>
<td>SET_UP_1 CALENDAR_PREFERENCES_FIRST;</td>
</tr>
<tr>
<td>CALENDAR_PREFS_REQD</td>
<td>SET_UPCALENDAR_PREFERENCES_FIRST;</td>
</tr>
<tr>
<td>CAMPAIGN_ALREADY_EXECUTED</td>
<td>YOU_CANNOT_DELETE_EMAIL_CAMPAIGNS_THAT_HAVE_ALREADY_BEEN_EXECUTED;</td>
</tr>
<tr>
<td>CAMPAIGN_IN_USE</td>
<td>YOU_CANNOT_DELETE_A_CAMPAIGN_EVENT_THAT_ALREADY_HAS_ACTIVITY;</td>
</tr>
<tr>
<td>CAMPAIGN_SET_UP_REQD</td>
<td>THE_FOLLOWING_STEPS_NEED_TO_BE_PERFORMED_BEFORE_A_CAMPAIGN_CAN_BE_CREATED;</td>
</tr>
<tr>
<td>CANT_CREATE_NON_UNIQUE_RCRD</td>
<td>WS_ERROR_CREATING_NON_UNIQUE_RECORD;</td>
</tr>
<tr>
<td>CANT_DELETE_CHILD_RCRD_FOUND</td>
<td>WS_CHILD_RECORD_FOUND_IN_DELETE;</td>
</tr>
<tr>
<td>CANT_DELETE_CHILD_RCRDS_EXIST</td>
<td>THIS_RECORD_CAN_NOT_BE_DELETED_Because_IT_HAS_CHILD_RECORDS;</td>
</tr>
<tr>
<td>CANT_DELETE_CUST_PLACEHOLDER</td>
<td>YOU_CANNOT_DELETE_THE_ANONYMOUS_CUSTOMER_PLACEHOLDER_IF_YOU MUST_DELETE_THIS_RECORD_FIRST_GO_TO_THE_A_HREFAPPSITESETUPSITEADMINNLSET_UP_WEB_SITEA_PAGE_AND_DESELECT_THE_ANONYMOUS_CUSTOMER_PLACEHOLDER;</td>
</tr>
<tr>
<td>CANT_DELETE_LEGACY_CATEGORY</td>
<td>LEGACY_CATEGORY_CANNOT_BE_REMOVED;</td>
</tr>
<tr>
<td>CANT_DELETE_STATUS_TYPE</td>
<td>YOU_CANNOT_DELETE_THE_ONLY_STATUS_OF_TYPE_1;</td>
</tr>
<tr>
<td>CANT_ESTABLISH_LINK</td>
<td>UNABLE_TO_ESTABLISH_LINK_WITH_1;</td>
</tr>
<tr>
<td>CANT_SEND_EMAIL</td>
<td>UNABLE_TO_SEND_NOTIFICATION_EMAIL;</td>
</tr>
<tr>
<td>CANT_SEND_EMAIL</td>
<td>UNABLE_TO_SEND_NOTIFICATION_EMAIL_TO_SUPPORT_REP;</td>
</tr>
<tr>
<td>CANT_SET_CLOSE_DATE</td>
<td>UNABLE_TO_SET_EXPECTED_CLOSE_DATE_OF_PROSPECTLEAD_BASED_ON_CURRENT_ESTIMATESOPPORTUNITIES;</td>
</tr>
<tr>
<td>CANT_SET_STATUS</td>
<td>UNABLE_TO_SET_STATUS_OF_PROSPECTLEAD_BASED_ON_CURRENT_ESTIMATES;</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CANT_SWITCH.Roles_FROMLOGIN</td>
<td>ROLE_SWITCHING_IS_NOT_ALLOWED_FROM_THIS LOGIN;</td>
</tr>
<tr>
<td>CANT_UPDATE_ATTACHMENT</td>
<td>Updating an attachment in this context is not supported. You can explicitly update the file through an update operation on the File record.</td>
</tr>
<tr>
<td>CANT_UPDATE_STATUS_TYPE</td>
<td>YOU_CANNOT_UPDATE_THE_ONLY_STATUS_OF_TYPE_1;</td>
</tr>
<tr>
<td>CASE_ALREADY_ASSIGNED</td>
<td>THIS_CASE_CANNOT_BE_GRABBED_BECauses IT IS_ALREADY_ASSIGNED TO_ANOTHER_REP_TO_VIEW_THE_CASE_GO_BACK_AND_CLICK_ON_ THE_CASE_NUMBER;</td>
</tr>
<tr>
<td>CASE_NOT_GROUP_MEMBER</td>
<td>_1 THIS_CASE_RECORD_DOES_NOT_BELONG_TO_YOUR_GROUP;</td>
</tr>
<tr>
<td>CASH_SALE_EDIT_DISALLOWED</td>
<td>THIS_CASH_SALE_CANNOT_BE_EDITED WHILE IT HAS AN AUTOMATED CLEARING HOUSE TRANSMISSION IN PROCESSTDTRTRTD_CLASSTEXTNBSPTO_VIEW_THE_STATUS_OF_CASH_SALES_WITH_ACH_TRANSMISSIONS_GO_TO_TRANSACTIONS_VIEW_ELECTRONIC_FUNDS_TRANSFER_STATUS;</td>
</tr>
<tr>
<td>CC_ALREADY_SAVED</td>
<td>THAT_CREDIT_CARD_IS_ALREADY_SAVED PLEASE USE THE_SAVED_CREDIT_CARD;</td>
</tr>
<tr>
<td>CC_EMAIL_ADDRESS_REQD</td>
<td>PLEASE_GO_BACK_AND_PROVIDE_AN_EMAIL_ADDRESS_TO_CC_STORE_ORDERS_TO;</td>
</tr>
<tr>
<td>CC_NUM_REQD</td>
<td>PLEASE_PROVIDE_A_CREDIT_CARD_NUMBER;</td>
</tr>
<tr>
<td>COMMSSN_ALREADY_CALCULATED</td>
<td>A_PLANS_SCHEDULES_OR_PAST_SALES_REPO_ASSIGNMENTS_CANNOT_BE_MODIFIED_ONCE_COMMISSIONS_AGAINST_THE_PLAN_HAVE_Been_CALCULATED;</td>
</tr>
<tr>
<td>COMMSSN_FEATURE_DISABLED</td>
<td>YOU_HAVE_NOT_ENABLED_THE_COMMISSIONS_FEATURE;</td>
</tr>
<tr>
<td>COMMSSN_PAYROLL_ITEM_REQD</td>
<td>A_COMMISSION_PAYROLL_ITEM_MUST_BE_ADDED_FOR_EACH_EMPLOYEE_TO_BE_PROCESSED_THROUGH_PAYROLL;</td>
</tr>
<tr>
<td>COMPANION_PROP_REQD</td>
<td>ERROR ITEMS_DO_NOT_HAVE_COMPANION_PROPERTY_COLUMN_0;</td>
</tr>
<tr>
<td>COMPANY_RCRD_DELETED</td>
<td>THE_COMPANY_YOU_TRY_TO_ATTACH_THE_CONTACT_TO_HAS_BEEN_DELETED_OR_MERGED;</td>
</tr>
<tr>
<td>CONSLD_PRNT_AND_CHILD_DISALLOWED</td>
<td>A_COMPANY_CAN_BE_A_CONSOLIDATED_CHILD_OR_A_CONSOLIDATED_PARENT_BUT_NOT_BOTH;</td>
</tr>
<tr>
<td>CONTACT_NOT_GROUP_MEMBER</td>
<td>_1 THIS_CONTACT DOES NOT_BELONG_TO_YOUR_GROUP;</td>
</tr>
<tr>
<td>COOKIES_DISABLED</td>
<td>YOU_HAVE_DISABLED.Cookies_FROM_BEING_STORED_ON_YOUR_COMPUTER_OR_TURNED_OFF_PERSSESSION.Cookies PLEASE_ENABLE_THIS_FEATURE_AND_TRY_AGAIN;</td>
</tr>
<tr>
<td>CRNCY_NOT_UPDATED</td>
<td>THE_FOLLOWING_CURRENCIES_WERE_NOT_UPDATED;</td>
</tr>
<tr>
<td>CRNCY_RCRD_DELETED</td>
<td>THIS_CURRENCY_RECORD_HAS_BEEN_DELETED YOU_CAN_CREATE_A_NEW_CURRENCY_RECORD_AT_LISTS_CURRENCIES;</td>
</tr>
<tr>
<td>CSTM_FIELD_KEY_REQD</td>
<td>WS_CUSTOM_FIELD_KEY_IS_MISSING;</td>
</tr>
<tr>
<td>CSTM_FIELD_VALUE_REQD</td>
<td>WS_CUSTOM_FIELD_VALUE_IS_MISSING;</td>
</tr>
<tr>
<td>CUST_ACCESS_FEATURE_DISABLED</td>
<td>CANT_OPEN_STORE_FOR_1 THIS_COMPANY DOES NOT HAVE THE CUSTOMER_ACCESS_FEATURE ENABLED THIS_FEATURE IS_REQUIRED FOR CUSTOMERS TO BE_ABLE_TO_REGISTER_CHECK_OUT_AND_LOG_IN_TO_THE_SCORE;</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CUST_LEAD_NOT_GROUP_MEM BR</td>
<td><em>1</em> THIS_CUSTOMER_OR_LEAD_DOES_NOT_BELONG_TO_YOUR_GROUP;</td>
</tr>
<tr>
<td>DATA_MUST_BE_UNIQUE</td>
<td>THE_UPDATE_FAILED_Because_EVERY_ENTRY_IN_THIS_COLUMN_MUST_BE_UNIQUE;</td>
</tr>
<tr>
<td>DATA_REQD</td>
<td>WS_PROVIDE_A PROPER_VALUE_FOR_THE_REQUIRED_FIELD;</td>
</tr>
<tr>
<td>DATE_EXPECTED</td>
<td>YOU_ENTERED_1 INTO A_FIELD WHERE A CALENDAR_DATE WAS EXPECTED; PLEASE_GO_BACK_AND_CHANGE_1_TO_THE_CORRECT_DATE;</td>
</tr>
<tr>
<td>DATE_PARAM_REQD</td>
<td>MISSING_DATE_PARAMETER;</td>
</tr>
<tr>
<td>DEFAULT_TYPE_DELETE_DISALLOW</td>
<td>YOU_CANNOT_DELETE_DEFAULT_TYPES;</td>
</tr>
<tr>
<td>DEPT_IN_USE</td>
<td>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;</td>
</tr>
<tr>
<td>DISALLOWED_IP_ADDRESS</td>
<td>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;</td>
</tr>
<tr>
<td>DISTRIB_REQD_ONE_DAY_BFOR</td>
<td>ALL_ITEMS_MUST_BE_DISTRIBUTED_AT_LEAST_ONE_DAY_Before_THE_Y_MAY_BE_TRANSFERRED;</td>
</tr>
<tr>
<td>DFRNT_SWAP_PRICE_LEVELS_REQD</td>
<td>PLEASE_SELECT_DIFFERENT_PRICE_LEVELS_TO_SWAP_PRICES;</td>
</tr>
<tr>
<td>DUE_DATE_BFORE_START_DATE</td>
<td>DUE_DATE_OCCURS_Before_START_DATE;</td>
</tr>
<tr>
<td>DUP_PAYROLL_ITEM</td>
<td>THERE_IS_ALREADY_A_PAYROLL_ITEM_Named_1;</td>
</tr>
<tr>
<td>DUP_TRACKING_NUM</td>
<td>YOU_ENTERED_THE_FOLLOWING_TRACKING_NUMBER_TWICE;</td>
</tr>
<tr>
<td>DUPLICATE_INVENTORY_NUM</td>
<td>DUPLICATE_INVENTORY_NUMBER_FOUND_IN_ENTRY_1;</td>
</tr>
<tr>
<td>DUPLICATE_INVENTORY_NUM</td>
<td>DUPLICATE_INVENTORY NUMBER_FOUND_ON_DIFFERENT_LINES_OF_TRANSACTION;</td>
</tr>
<tr>
<td>DUPLICATE_KEYS</td>
<td>WS_DUPLICATED_KEYS;</td>
</tr>
<tr>
<td>DUPLICATE_NAME_FOR_PRD</td>
<td>PLEASE_CHOOSE_A_DIFFERENT_PERIOD_NAME_1_IS_ALREADY_TAKEN;</td>
</tr>
<tr>
<td>DUPLICATE_NAME_FOR_ROLE</td>
<td>PLEASE_CHOOSE_A_DIFFERENT_ROLE_NAME_1_IS_ALREADY_TAKEN;</td>
</tr>
<tr>
<td>DUPLICATE_USER_NAME</td>
<td>A_USER_WITH_THIS_NAME_ALREADY_EXISTS;</td>
</tr>
<tr>
<td>EMAIL_ADDRS_REQD</td>
<td>PLEASE_ENTER_YOUR_EMAIL_ADDRESS;</td>
</tr>
<tr>
<td>EMAIL_ADDRS_REQD_TO_NOTIFY</td>
<td>PLEASE_ENTER_AN_EMAIL_ADDRESS_FOR_THIS_COMPANY_A_NOTIFICATION_EMAIL_WILL_BE_SENT_WHEN_THIS_CASE_RECORD_IS_SAVED;</td>
</tr>
<tr>
<td>EMAIL_ADDRS_REQD_TO_NOTIFY</td>
<td>THE_RECIPIENT_YOU_ARE_SENDING_THIS_EMAIL_TO DOES NOT HAVE_AN_EMAIL_ADDRESS_PLEASE_ENTER.ONE_AND.Try.AGAIN;</td>
</tr>
<tr>
<td>EMAIL_REQD</td>
<td>YOU_MUST_ENTER_A_VALID_EMAIL_ADDRESS_IN_ORDER_TO_EMAIL_THE_TRANSACTION;</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EMPL_IN_USE</td>
<td>YOU_CANT_DELETE_THIS_EMPLOYEE_AS_COMMISSIONS_HAVE_BEEN_CALCULATED_FOR_THIS_EMPLOYEE;</td>
</tr>
<tr>
<td>ERROR_PROCESSING_TRANSPORT</td>
<td>THERE_WERE_ERRORS_PROCESSING_THE_SELECTED_TRANSACTIONS_PLEASE_PROCESS_THEM_INDIVIDUALLY_FOR_MORE_INFORMATION;</td>
</tr>
<tr>
<td>EVENT_ID_NOT_FOUND</td>
<td>EVENT_ID_NOT_FOUND;</td>
</tr>
<tr>
<td>EXPIRED_SEARCH_CRITERIA</td>
<td>YOUR_SEARCH_CRITERIA_EXPIRED_THE_CRITERIA_FOR_A_GIVEN_SEARCH_GENERALLY_EXPIRE_AFTER_15_MINUTES_OF_INACTIVITY_CLICK_A_HREF_ONCLICKHISTORYGO1HEREA_TO_GO_BACK_TO_THE_SEARCH_DEFINITION_PAGE_AND_RESUBMIT_YOUR_SEARCH;</td>
</tr>
<tr>
<td>FAX_NUM_REQD</td>
<td>YOU_MUST_ENTER_A_FAX_NUMBER;</td>
</tr>
<tr>
<td>FAX_NUM_REQD</td>
<td>YOU_MUST_ENTER_A_FAX_NUMBER_FOR_THIS_RECIPIENT_BEFORE_PERFORMING_A_FAX_MERGE_OPERATION;</td>
</tr>
<tr>
<td>FAX_NUM_REQD</td>
<td>YOU_MUST_ENTER_A_VALID_FAX_NUMBER_IN_ORDER_TO_FAX_THE_TRANSACTION;</td>
</tr>
<tr>
<td>FEATURE_DISABLED</td>
<td>THE_FEATURE_REQUIRED_TO_ACCESS_THIS_PAGE_IS_NOT_ENABLED_IN_THIS_ACCOUNT;</td>
</tr>
<tr>
<td>FEATURE_UNAVAILABLE</td>
<td>THE_FEATURE_IS_NOT_AVAILABLE_TO_YOUR_COMPANY;</td>
</tr>
<tr>
<td>FED_WITHHOLDING_REQD</td>
<td>YOUR_EMPLOYEE_RECORD_DOES_NOT_HAVE_CURRENT_FEDERAL_WITHHOLDING_INFORMATION_PLEASE_CONTACT_YOUR_SUPERVISOR_TO_SET_UP_YOUR_RECORD_WITH_THE_APPROPRIATE_INFORMATION;</td>
</tr>
<tr>
<td>FIELD_CALL_DATE_REQD</td>
<td>MISSING_REQUIRED_FIELD_CALL_DATE;</td>
</tr>
<tr>
<td>FIELD_DEFN_REQD</td>
<td>FIELD_DEFINITION_NOT_FOUND;</td>
</tr>
<tr>
<td>FILE_NOT_FOUND</td>
<td>FILEMEDIA_ITEM_1_NOT_FOUND;</td>
</tr>
<tr>
<td>FILE_REQD</td>
<td>YOU_MUST_UPLOAD_A_FILE_BEFORE_CREATING_THIS_MEDIA_ITEM;</td>
</tr>
<tr>
<td>FILTER_BY_AMT_REQD</td>
<td>PLEASE_ENTER_AN_AMOUNT_TO_FILTER_BY;</td>
</tr>
<tr>
<td>FIRST_LAST_NAMES_REQD</td>
<td>PLEASE_ENTER_BOTH_YOUR_FIRST_AND_LAST_NAME;</td>
</tr>
<tr>
<td>FORM_RESUBMISSION_REQD</td>
<td>YOU_HAVE_LOGGED_IN_TO_A_DIFFERENT_USER_SINCE_YOU_NAVIGATED_TO_THIS_FORM__YOU_MUST_RESUBMIT_THIS_FORM_AS_THE_NEW_USER;</td>
</tr>
<tr>
<td>FULL_DISTRIB_REQD</td>
<td>YOU_MUST_FULLY_DISTRIBUTE_ALL_1_NUMBERS_FOR_1_NUMBERED_ITEMS;</td>
</tr>
<tr>
<td>FULL_USERS_REQD_TO_INTEGRATE</td>
<td>ONLY_FULL_1_USERS_CAN_INTEGRATE_WITH_PARTNERS;</td>
</tr>
<tr>
<td>GETALL_RCRD_TYPE_REQD</td>
<td>WS_MISSING_GETALL_RECORD_TYPE;</td>
</tr>
<tr>
<td>GROUP_DSNT_EXIST</td>
<td>THAT_GROUP_DOES_NOT_EXIST;</td>
</tr>
<tr>
<td>GROUP_REQD</td>
<td>YOU_CANNOT_PERFORM_A_BULK_MERGE_OPERATION_WITH_AN_EMPTY_GROUP;</td>
</tr>
<tr>
<td>ILLEGAL_PERIOD_STRUCTURE</td>
<td>ILLEGAL_PERIOD_STRUCTURE_DATE_1_IS_IN_MULTIPLE_PERIODS;</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>INACTIVE_RCRD_FOR_ROLE</td>
<td>THE_RECORD_FOR_THIS_ROLE_HAS_BEEN_MADE_INACTIVE;</td>
</tr>
<tr>
<td>INCRCST_ORD_INFO</td>
<td>THE_ORDER_CONTAINS_INCORRECT_INFORMATION_AND_WAS_NOT_PLACED;</td>
</tr>
<tr>
<td>INSUFFCNT_NUM_PRDS_FOR_REV_REC</td>
<td>NOT_ENOUGH_ACCOUNTING_PERIODS_IN_RANGE_SPECIFIED_FOR_REVENUE_RECOGNITION;</td>
</tr>
<tr>
<td>INSUFFCNT_OPEN_PRDS_FOR_REV_REC</td>
<td>NOT_ENOUGH_OPEN_ACCOUNTING_PERIODS_AVAILABLE_FOR_REVENU_E_RECOGNITION;</td>
</tr>
<tr>
<td>INCRCST_ORD_INFO</td>
<td>THE_ORDER_CONTAINS_INCORRECT_INFORMATION_AND_WAS_NOT_PLACED;</td>
</tr>
<tr>
<td>INSUFFICIENT_CHARS_IN_SEARCH</td>
<td>GLOBAL_SEARCHES_MUST_CONTAIN_AT_LEAST_THREE_CHARACTERS_TO_PREVENT_EXCESSIVE_MATCHES;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>YOU_MUST_HAVE__TRANSACTIONS__FULFILL_SALES_ORDERS__PERMISSION_TO_FULFILL_SALES_ORDERS;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>YOU_MUST_HAVE_EITHER_TRANSACTIONS__INVOICE_OR_TRANSACTIONS__CASH_SALE_PERMISSION_TO_BILL_SALES_ORDERS;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>YOU_NEED_EMPLOYEE_ACCESS_IN_ORDER_TO_DELETE_THIS_RECORD;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>FOR_SECURITY_REASONS_ONLY_AN_ADMINISTRATOR_IS_ALLOWED_TO_EDIT_AN_ADMINISTRATOR_RECORD;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>THIS_ORDER_HAS_BEEN_PARTIALLY_OR_FULLY_PROCESSED_AND_MAY_NOT_BE_EDITED_BY_A_USER_WITHOUT_PERMISSION_TO_APPROVE_SALES_ORDERS;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>YOU_DO_NOT_HAVE_ACCESS_TO_THE_ACTIVITY_HISTORY_FOR_THAT_RECORD;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>YOU_DO_NOT_HAVE_ACCESS_TO_THE_MEDIA_ITEM_YOU_SELECTED;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>YOU_DO_NOT_HAVE_PERMISSION_TO_EMAIL_TRANSACTIONS;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>YOU_MUST_HAVE_EITHER_TRANSACTIONS__INVOICE_OR_TRANSACTIONS__CASH_SALE_PERMISSION_TO_FULFILL_SALES_ORDERS;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>YOU_ARE_NOT_ALLOWED_TO_APPROVE_YOUR_OWN_TRANSACTIONS;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>_1_THE_RESTRICTIONS_ON_YOUR_ROLE_DENY_YOU_ACCESS_TO_THIS_RECORD;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>_1_THE_2_RESTRICTIONS_ON_YOUR_ROLE_DENY_YOU_ACCESS_TO_THIS_RECORD;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>_1_THE_2_RESTRICTIONS_ON_YOUR_ROLE_PREVENT_YOU_FROM_SEEING_THIS_RECORD;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>INSUFFICIENT_PRIVILEGES;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>YOUR_ROLE_DOES_NOT_HAVE_PERMISSION_TO_PROVISION_ACCOUNTS;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>USER_PERMISSION_LEVEL_Could_NOT_BE_ESTABLISHED;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>YOU_DO_NOT_HAVE_PERMISSION_TO_PERFORM_THIS_OPERATION;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>YOU_DO_NOT_HAVE_PERMISSION_TO_VIEW_THIS_PAGE;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>YOU_DO_NOT_HAVE_PRIVILEGES_TO_PERFORM_THIS_ACTION;</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>YOU DO NOT HAVE PRIVILEGES TO PERFORM THIS OPERATION;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>YOU DO NOT HAVE PRIVILEGES TO USE THIS PAGE;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>YOU DO NOT HAVE PRIVILEGES TO VIEW THIS PAGE;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td><em>1</em> THE CUSTOMER RESTRICTIONS ON YOUR PARTNER_ROLE PREVENT YOU FROM SEEING THIS RECORD;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>_1. YOU NEED _1. THE _2 PERMISSION TO ACCESS THIS_PAGE PLEASE CONTACT YOUR ACCOUNT_ADMINISTRATOR;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>PERMISSION_ERROR YOU MAY NOT EDIT THIS ROLE;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>PERMISSION VIOLATION PARTNERS DO NOT HAVE ACCESS TO THIS REPORT;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>PERMISSION VIOLATION PARTNERS MAY NOT DELETE SAVED REPORTS;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>PERMISSION VIOLATION YOU CANNOT DELETE SAVED REPORTS NOT CREATED BY YOURSELF;</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>WS NO PERMISSIONS TO SET VALUE</td>
</tr>
<tr>
<td>INTEGER_REQD_FOR_QTY</td>
<td>QUANTITY MUST BE AN INTEGER FOR NUMBERED_ITEMS;</td>
</tr>
<tr>
<td>INVALID_ACCT</td>
<td>INVALID_LOGIN_NO_SUCH_ACCOUNT;</td>
</tr>
<tr>
<td>INVALID_ACCT</td>
<td>WS_INVALID_ACCOUNT_NUMBER;</td>
</tr>
<tr>
<td>INVALID_ADJUSTMENT_ACCT</td>
<td>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;</td>
</tr>
<tr>
<td>INVALID_AUTHORIZATION</td>
<td>CORRECT AUTHORIZATION EXCEPTION;</td>
</tr>
<tr>
<td>INVALID_BALANCE_RANGE</td>
<td>YOUR BALANCE IS NOT WITHIN THE ALLOWED RANGE;</td>
</tr>
<tr>
<td>INVALID_BUG_NUM</td>
<td>BUG_NUMBER_SPECIFIED WAS INCORRECT _1 ISNT A NUMBER;</td>
</tr>
<tr>
<td>INVALID_CAMPAIGN_GROUP_SIZE</td>
<td>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;</td>
</tr>
<tr>
<td>INVALID_CATGRY_TAX_AGENCY_REQ</td>
<td>A VENDOR MUST BE CREATED_IN A CATEGORY WITH THE TAX AGENCY_CHECKBOX_CHECKED;</td>
</tr>
<tr>
<td>INVALID_CC_EMAIL_ADDRESS</td>
<td>THE_EMAIL_ADDRESS_TO_CC_STORE_ORDERS_TO_IS INVALID PLEASE _GO_BACK_AND_CORRECT_IT;</td>
</tr>
<tr>
<td>INVALID_CC_NUM</td>
<td>CREDIT_CARD_NUMBER IS NOT VALID PLEASE_CHECK_THAT_ALL_DIGITS_WERE_ENTERED_CORRECTLY;</td>
</tr>
<tr>
<td>INVALID_CC_NUM</td>
<td>CREDIT_CARD_NUMBER MUST Contain ONLY DIGITS;</td>
</tr>
<tr>
<td>INVALID_CC_NUM</td>
<td>CREDIT_CARD_NUMBERS MUST Contain BETWEEN 13 AND 16 DIGITS;</td>
</tr>
<tr>
<td>INVALID_CHARS_IN_EMAIL</td>
<td>EMAIL_ADDRESS_CONTAINS INVALID CHARACTERS;</td>
</tr>
<tr>
<td>INVALID_CHARS_IN_NAME</td>
<td>THE_FROM_NAME_FIELD_CANNOT_CONTAIN_APOSTROPHES_QUOTATION_MARKS_COMMAS_OR_GREATER_THAN_OR_LESS_THAN_SIGNS;</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>INVALID_CHARS_IN_NAME</td>
<td>YOU_CANNOT_USE_THE_COLON_CHARACTER_IN_THE_TOPIC_NAME__PLEASE_REMOVE_IT_;</td>
</tr>
<tr>
<td>INVALID_CHARS_IN_PARAM_FIELD</td>
<td>THE_ADDITIONAL_PARAMETERS_FIELD_CAN_NOT_CONTAIN_ANY_OF_THE_FOLLOWING_CHARACTERS___PLEASE_REMOVE_THEM_AND_TRY_AGAIN;</td>
</tr>
<tr>
<td>INVALID_CHARS_IN_URL</td>
<td>SPACES_ARE_NOT_ALLOWED_IN_THE_URL_EXAMPLES_OF_A_VALID_URL_ARE_BROWSABLE_HTTP_URLS_MYDOMAINCOMIMAGEGIFBSPNBSPORNBSNBSPNSHTTPONETWOORGUSERNAMETESTJPG;</td>
</tr>
<tr>
<td>INVALID_CHARS_IN_URL</td>
<td>THE_URL_COMPONENT_YOU_HAVE_CHOSEN_CONTAINS_A_SPACE_OR_ONE_OF_THE_FOLLOWING_PROHIBITED_CHARACTER___PLEASE_REMOVE_THEM_AND_TRY_AGAIN;</td>
</tr>
<tr>
<td>INVALID_COSTING_METHOD</td>
<td>SERIAL_AND_LOT_ARE_THE_ONLY_COSTING_METHODS_THAT_MAY_BE_PASSED_AS_PARAMETERS_TO_THIS_PAGE;</td>
</tr>
<tr>
<td>INVALID_CREDENTIALS</td>
<td>NCEL_AUTHORIZATION_EXCEPTION;</td>
</tr>
<tr>
<td>INVALID_CSTM_FIELD_REF</td>
<td>WS_INVALID_CUSTOM_FIELD_REF;</td>
</tr>
<tr>
<td>INVALID_CSTM_RCRD_TYPE_KEY</td>
<td>WS_INVALID_CUSTOM_RECORD_TYPE_KEY;</td>
</tr>
<tr>
<td>INVALID_DATA</td>
<td>Error in record number &lt;nskey&gt;: Invalid field value &lt;field&gt;. Please refer to the XSD for enumerated list of valid field values;</td>
</tr>
<tr>
<td>INVALID_DATE_RANGE</td>
<td>INVALID_DATE_RANGE THE_EVENT_1_START_TIME_MUST_BE_EARLIER_THAN_THE_END_TIME;</td>
</tr>
<tr>
<td>INVALID_DATE_RANGE</td>
<td>THE_DATE_RANGE_YOU_SPECIFIED_DOES_NOT_FALL_INSIDE_THAT_OF_THE_PARENT_PERIOD;</td>
</tr>
<tr>
<td>INVALID_EMAIL</td>
<td>EMAIL_ADDRESS_IS_NOT_VALID;</td>
</tr>
<tr>
<td>INVALID_EMAIL</td>
<td>YOU_HAVE_ENTERED_AN_INVALID_EMAIL_ADDRESS__PLEASE_TRY_AGAIN;</td>
</tr>
<tr>
<td>INVALID_EMAIL</td>
<td>YOUR_EMAIL OR_CODE IS INVALID__PLEASE_TRY_AGAIN;</td>
</tr>
<tr>
<td>INVALID_END_DATE</td>
<td>EVENT_1_RECURRENCE_END_DATE_IS_INVALID;</td>
</tr>
<tr>
<td>INVALID_END_TIME</td>
<td>INVALID_END_TIME;</td>
</tr>
<tr>
<td>INVALID_FILE</td>
<td>VERIFY_THAT_YOU_HAVE_A_VALID_FILE_TO_UPLOAD;</td>
</tr>
<tr>
<td>INVALID_FILE_TYPE</td>
<td>INVALID_FILE_TYPE_FILE_IS_NOT_A_COMPRESSED_ZIP_FILE;</td>
</tr>
<tr>
<td>INVALID_FILE_TYPE</td>
<td>INVALID_FILE_TYPE_FILE_IS_NOT_A_COMPRESSEDZIP_FILE;</td>
</tr>
<tr>
<td>INVALID_FORMAT_IN_PARAM_FIELD</td>
<td>THE_ADDITIONAL_PARAMETERS_FIELD_IS_NOT_FORMATTED_CORRECTLY__PLEASE_REFORMAT_AND_TRY_AGAIN;</td>
</tr>
<tr>
<td>INVALID_FROM_DATE</td>
<td>INVALID_FROM_DATE;</td>
</tr>
<tr>
<td>INVALID_FROM_TIME</td>
<td>INVALID_FROM_TIME;</td>
</tr>
<tr>
<td>INVALID_GROUP_SIZE</td>
<td>YOU_CANNOT_PERFORM_A_BULK_MERGE_OPERATION_WITH_A_GROUP_LARGER_THAN_500_RECORDS;</td>
</tr>
<tr>
<td>INVALID_GST_PST_AGENCIES</td>
<td>THE_GST_OR_PST_AGENCIES_ARE_NOT_VALIDPLEASE_REVIEW_YOUR_COMPANY_PREFERENCES;</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>INVALID_INVENTORY_NUM</td>
<td>INVALID_SET_OF_INVENTORY_NUMBERS_VALUES_MUST_BE_SEPARATED_BY_COMMAS_SPACES_TABS_OR_LINE_FEEDS;</td>
</tr>
<tr>
<td>INVALID_IP_ADDRESS_RULE</td>
<td>THE_FOLLOWING_IP_ADDRESS_RULE_IS_NOT_VALID_1;</td>
</tr>
<tr>
<td>INVALID_KEY_OR_REF</td>
<td>WS_INVALID_KEY;</td>
</tr>
<tr>
<td>INVALID_KEY_OR_REF</td>
<td>WS_INVALID_REFERENCE_KEY_1;</td>
</tr>
<tr>
<td>INVALID_KEY_OR_REF</td>
<td>WS_INVALID_REFERENCE_KEY_2;</td>
</tr>
<tr>
<td>INVALID_LOGIN</td>
<td>INVALID_LOGIN_ONLINE_FORM_ACCESS_IS_DISABLED;</td>
</tr>
<tr>
<td>INVALID_LOGIN</td>
<td>INVALID_LOGIN_SUPPLIER_ACCESS_IS_DISABLED;</td>
</tr>
<tr>
<td>INVALID_LOGIN_ATTEMPT</td>
<td>INVALID_LOGIN_ATTEMPT;</td>
</tr>
<tr>
<td>INVALID_LOGIN_CREDENTIALS</td>
<td>A_PROBLEM_OCCURRED_VERIFYING_THE_PRESENTED_EMAIL_ADDRESS_PASSWORD_ROLENAME_OR_ACCOUNT_NUMBERPLEASE_VERIFY_THOSE_PIECES_OF_INFORMATION_AND_TRY_AGAIN;</td>
</tr>
<tr>
<td>INVALID_LOGIN_CREDENTIALS</td>
<td>YOU_HAVE_ENTERED_AN_INVALID_EMAIL_ADDRESS_OR_ACCOUNT_NUMBERPLEASE_TRY_AGAIN;</td>
</tr>
<tr>
<td>INVALID_LOGIN_CREDENTIALS</td>
<td>YOU_HAVE_ENTERED_AN_INVALID_EMAIL_ADDRESS_OR_PASSWORD_PLEASE_TRY_AGAIN;</td>
</tr>
<tr>
<td>INVALID_LOGIN_CREDENTIALS</td>
<td>YOU_HAVE_ENTERED_AN_INVALID_LOGIN_PASSWORD_PLEASE_TRY_AGAIN;</td>
</tr>
<tr>
<td>INVALID_LOGIN_CREDENTIALS</td>
<td>INVALID_LOGIN. THE_EMAIL_ADDRESS_ACCOUNT_NUMBER_AND_ROLE_YOU_HAVE_SPECIFIED_MATCH_MORE_THAN_ONE_EMPLOYEE. PLEASE_CONTACT_YOUR_ADMINISTRATOR_TO_CORRECT_THE_PROBLEM.</td>
</tr>
<tr>
<td>INVALID_LOGIN_IP</td>
<td>INVALID_LOGIN_IP_ADDRESS_DOES_NOT_MATCH_ANY_OF_THE_IPADDRESS_RULESPECIFIED_FOR_THIS_ENTITY;</td>
</tr>
<tr>
<td>INVALID_LOT_NUM_FORMAT</td>
<td>LOT_NUMBERS_MUST_BE_ENTERED_USING_THIS_FORMAT;</td>
</tr>
<tr>
<td>INVALID_MARKUP_DISCOUNT</td>
<td>MARKUPDISCOUNT_MUST_BE_BETWEEN_999_AND_999;</td>
</tr>
<tr>
<td>INVALID_MEMRZD_TRANS</td>
<td>A_MEMORIZED_TRANSACTION_MAY_NOT_CONTAIN_ANY_SERIAL_OR_LOT_NUMBERS;</td>
</tr>
<tr>
<td>INVALID_NUMBER</td>
<td>INVALID_INTEGER_1;</td>
</tr>
<tr>
<td>INVALID_NUMBER</td>
<td>INVALID_NUMBER_1;</td>
</tr>
<tr>
<td>INVALID_ORD_STATUS</td>
<td>THIS_ORDER_HAS_BEEN_PARTIALLY_OR_FULLY_PROCESSED_AND_MAY_NOT_BE_RESET_TO_PENDING_APPROVAL;</td>
</tr>
<tr>
<td>INVALID_POST</td>
<td>INVALID_POST;</td>
</tr>
<tr>
<td>INVALID_PST_TAX_VALUE</td>
<td>PST_TAX_VALUE_IS_NOT_A_VALID_NUMBER_1;</td>
</tr>
<tr>
<td>INVALID_PSWD</td>
<td>YOUVE_USED_THAT_PASSWORD_BEFOREPLEASE_CHOOSE_A_NEWPASSWORD;</td>
</tr>
<tr>
<td>INVALID_PSWD</td>
<td>PASSWORD_MUST_BE_AT_LEAST_6_CHARACTERS_LONG;</td>
</tr>
<tr>
<td>INVALID_PSWD</td>
<td>PASSWORD_MUST_BE_AT_LEAST_6_CHARACTERS_LONG_AND_CONTAIN_AT_LEAST_ONE_NUMBER_OR_SPECIAL_CHARACTER;</td>
</tr>
<tr>
<td>INVALID_PSWD</td>
<td>PASSWORD_MUST_CONTAIN_AT_LEAST_ONE_LETTER_AZ;</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>INVALID_PSWD</td>
<td>PASSWORD_MUST_CONTAIN_AT_LEAST_ONE_NUMBER_OR_SPECIAL_CHARACTER;</td>
</tr>
<tr>
<td>INVALID_PSWD</td>
<td>PASSWORD_PROVIDED_DOES_NOT_MATCH_SYSTEM_PASSWORD;</td>
</tr>
<tr>
<td>INVALID_PSWD</td>
<td>YOUR_PASSWORD_MUST_BE_AT_LEAST_6_CHARACTERS;</td>
</tr>
<tr>
<td>INVALID_PSWD</td>
<td>THE_CURRENT_PASSWORD_YOU_SUPPLIED_IS_INCORRECT;</td>
</tr>
<tr>
<td>INVALID_PSWD</td>
<td>YOUR_NEW_PASSWORD_MUST_BE_AT_LEAST_6_CHARACTERS_CONTAIN_AT_LEAST_ONE_NONLETTER_AND_BE_SUBSTANTIALLY_DIFFERENT_FROM_THE_CURRENT_PASSWORD;</td>
</tr>
<tr>
<td>INVALID_PSWD_ILLEGAL_CHAR</td>
<td>PASSWORD_CONTAINS_AN_ILLEGAL_CHARACTER;</td>
</tr>
<tr>
<td>INVALID_QUANTITY</td>
<td>SERIAL_AND_LOT_NUMBER_QUANTITIES_MUST_BE_POSITIVE;</td>
</tr>
<tr>
<td>INVALID_RCRD_TYPE</td>
<td>WS_INVALID_RECORD_TYPE;</td>
</tr>
<tr>
<td>INVALID_REFFERER_EMAIL</td>
<td>THE_REFFERER_EMAIL_ADDRESS_YOU_HAVE_ENTERED_IS_NOT_VALID_PLEASE_TRYAGAIN;</td>
</tr>
<tr>
<td>INVALID_ROLE</td>
<td>WS_ROLE_IS_INVALID;</td>
</tr>
<tr>
<td>INVALID_ROLE</td>
<td>YOU_HAVE_ENTERED_AN_INVALID_ROLE_ID_PLEASE_TRYAGAIN;</td>
</tr>
<tr>
<td>INVALID_ROLE_FOR_EVENT</td>
<td>YOU_SEEM_TO_HAVE_BEEN_INVITED_TO_THIS_EVENT_IN_A_DIFFERENT_ROLE_PLEASE_CHANGE_YOUR_ROLE_TO_VIEW_THE_EVENT;</td>
</tr>
<tr>
<td>INVALID_RQST_CONTACTS_EXIST</td>
<td>IT_HAS_ASSOCIATED_PRIMARY_CONTACTS;</td>
</tr>
<tr>
<td>INVALID_RQST_PARENT_REQD</td>
<td>IT_HAS_ASSOCIATED_CONTACT_RECORDS_THAT_WOULD_BE_LEFT_WITH_NO_PARENT_COMPANY;</td>
</tr>
<tr>
<td>INVALID_RQST_SBCUST_JOBS_EXIST</td>
<td>IT_HAS_ASSOCIATED_SUBCUSTOMERS_OR_JOBS;</td>
</tr>
<tr>
<td>INVALID_SCHEDUL_FORMAT</td>
<td>TO_CREATE_A_VALID_SCHEDULE_PLEASE_ENTER_THE_BRACKET_VALUES_IN_ASCENDING.Orders_WITHOUT_GAPS;</td>
</tr>
<tr>
<td>INVALID_SEARCH</td>
<td>THAT_SEARCH_OR_MASS_UPDATE_DOES_NOT_EXIST;</td>
</tr>
<tr>
<td>INVALID_SEARCH_CRITERIA</td>
<td>GLOBAL_SEARCH_SUPPORTS_AT_MOST_THREE_KEYWORDS_AND_REQUIRES_AT_LEAST_ONE_KEYWORDS_ARE_COMPOSED_OF_ONLY_LETTERS_DIGITS_AND_DASHES;</td>
</tr>
<tr>
<td>INVALID_SEARCH_FIELD_OBJ</td>
<td>WS_ERROR_INVALID_SEARCH_CUSTOMFIELD_OBJECT;</td>
</tr>
<tr>
<td>INVALID_SEARCH_FIELD_OBJ</td>
<td>WS_INVALID_SEARCH_FIELD_OBJECT;</td>
</tr>
<tr>
<td>INVALID_SEARCH_MORE</td>
<td>WS_INVALID_SEARCHMORE_OPERATION;</td>
</tr>
<tr>
<td>INVALID_SEARCH_OPERATOR</td>
<td>WS_INVALID_SEARCH_OPERATOR;</td>
</tr>
<tr>
<td>INVALID_SEARCH_PAGE_INDEX</td>
<td>WS_INVALID_SEARCH_PAGE_INDEX;</td>
</tr>
<tr>
<td>INVALID_SEARCH_PAGE_SIZE</td>
<td>WS_INVALID_SEARCH_PAGE_SIZE;</td>
</tr>
<tr>
<td>INVALID_SEARCH_VALUE</td>
<td>WS_MISSING_SEARCH_VALUE;</td>
</tr>
<tr>
<td>INVALID_SEARCH_VALUE</td>
<td>WS_MISSING_SEARCH_VALUES;</td>
</tr>
<tr>
<td>INVALID_SERIAL_OR_LOT_NUMBER</td>
<td>SERIAL_AND_LOT_NUMBERS_MAY_NOT_CONTAIN_THE_1_CHARACTER;</td>
</tr>
<tr>
<td>INVALID_STATE</td>
<td>SIGNUP_PROSPECT_STATE_1 IS INVALID;</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>INVALID_SUPERVISOR</td>
<td>YOU CANT_INSERT_THIS_EMPLOYEE_RECORD_AS_IT_WOULD_CREATE_A_LOOP_IN_THE_SUPERVISOR_HIERARCHY;</td>
</tr>
<tr>
<td>INVALID_SUPERVISOR</td>
<td>EMPLOYEES_CAN_NOT_BE_THEIR_OWN_SUPERVISOR;</td>
</tr>
<tr>
<td>INVALID_TAX_VALUE</td>
<td>GST_AND_PST_AMOUNT_CANNOT_BE_NEGATIVE;</td>
</tr>
<tr>
<td>INVALID_TAX_VALUE</td>
<td>GST_TAX_VALUE_IS_NOT_A_VALID_NUMBER_1;</td>
</tr>
<tr>
<td>INVALID_TIME</td>
<td>_1 IS NOT A VALID_TIME</td>
</tr>
<tr>
<td>INVALID_TO_DATE</td>
<td>INVALID_TO_DATE;</td>
</tr>
<tr>
<td>INVALID_TRACKING_NUM</td>
<td>YOU HAVE_ENTERED_A_TRACKING_NUMBER_THAT_EXCEEDS_THE_MAXIMUM_SIZE_OF_1 CHARACTERS;</td>
</tr>
<tr>
<td>INVALID_TRANS</td>
<td>THIS_TRANSACTION_IS_NOT_VALID;</td>
</tr>
<tr>
<td>INVALID_TRANSACTION_DATE</td>
<td>THERE ARE NO ACCOUNTING PERIODS THAT COVER THIS TRANSACTION_DATE;</td>
</tr>
<tr>
<td>INVALID_TRANSACTION_DATE</td>
<td>TRANSACTION_DATE_1 IS NOT VALID;</td>
</tr>
<tr>
<td>INVALID_URL</td>
<td>PLEASE_BEGIN_THE_1_URL_WITH_BHTTPBNBSPNBSORNBSPNBSPBHTTBPSBPBPEXAMPLES_OF_A_VALID_1URL_AREBRHTTPWWWMYDOMAINCOMIMAGEGIFBNBSORNBSPNBSPBHTHTTPSONETWOORUSERNAMETESTJPEG;</td>
</tr>
<tr>
<td>INVALID_VAT_AMOUNT</td>
<td>VAT_AMOUNT_CANNOT_BE_NEGATIVE;</td>
</tr>
<tr>
<td>INVALID_WS_VERSION</td>
<td>MESSAGE_CONTAINS_A_WEB_SERVICES_VERSION_THAT_DOES_NOT_EXIST;</td>
</tr>
<tr>
<td>INVALID_YEAR_FORMAT</td>
<td>ILLEGAL_YEAR_FORMAT_OR_VALUE_EXAMPLES_1999_2000_2001_ETC;</td>
</tr>
<tr>
<td>INVALID_ZIP_FILE</td>
<td>INVALID_ARCHIVE_ZIP_FILE_MUST_CONTAIN_AT_LEAST_ONE_FILE;</td>
</tr>
<tr>
<td>INVENTORY_NUM_DISALLOWED</td>
<td>INVENTORY_NUMBERS_ARE_ONLY_ALLOWED_ON_ITEMS_WITH_SERIAL_NUMBERED_OR_LOT_NUMBERED_ITEMS;</td>
</tr>
<tr>
<td>ITEM_ACCT_REQD</td>
<td>ONE_OF_THE_ITEMS_ON_THIS_TRANSACTION_HAS_AN_AMOUNT_BUT_NO_ACCOUNTPLEASE_FIX_THE_ITEM_AND_RESUBMIT_THE_TRANSACTION;</td>
</tr>
<tr>
<td>ITEM_ACCT_REQD</td>
<td>ONE_OF_THE_ITEMS_ON_THIS_TRANSACTION_HAS_AN_AMOUNT_BUT_NO_ACCOUNTPLEASE_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;</td>
</tr>
<tr>
<td>ITEM_ACCT_REQD</td>
<td>YOU MUST_SPECIFY_ASSET_AND_COGS_ACCOUNTS_FOR_THIS_INVENTORY_ITEM;</td>
</tr>
<tr>
<td>ITEM_COUNT_MISMATCH</td>
<td>COGSCORRECTION_2_MEANS_OF_CALCULATING_THE_ITEM_COUNT_DO_NOT_MATCH_FOR_ITEM_1 VS 2;</td>
</tr>
<tr>
<td>ITEM_COUNT_MISMATCH</td>
<td>COGSCORRECTION_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;</td>
</tr>
<tr>
<td>ITEM_IS_UNAVAILABLE</td>
<td>ITEM IS UNAVAILABLE;</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>ITEM_NAME_MUST_BE_UNIQUE</td>
<td>AN ITEM WITH THAT NAME ALREADY EXISTS PLEASE CHOOSE ANOTHER NAME;</td>
</tr>
<tr>
<td>ITEM_PARAM_REQD_IN_URL</td>
<td>ERROR__ITEM_PARAMETER_IDNNN_WAS_NOT_PROVIDED_ON_THE_URL_;</td>
</tr>
<tr>
<td>LIST_KEY_REQD</td>
<td>WS_MISSING_LIST_KEY;</td>
</tr>
<tr>
<td>LOCATION_REQD</td>
<td>YOU MUST SPECIFY A LOCATION IN ORDER TO USE _1_NUMBERS WHEN_MULTILOCATION_INVENTORY_IS_ENABLED;</td>
</tr>
<tr>
<td>LOCATIONS_IN_USE</td>
<td>YOUR_CLASSES_CANNOT_BE_CONVERTED_TO_LOCATIONS_BECUSE_YOUR_EXISTING_LOCATION_RECORDS_ARE_REFERRED_TO_BY_TRANSACTIONS_OR_OTHER_RECORDS;</td>
</tr>
<tr>
<td>LOCATIONS_SETUP_REQD</td>
<td>YOU MUST FIRST_DEFINE_LOCATIONS_LISTSLOCATIONSNEW_BEFORE_YOU_CAN_DISTRIBUTE_INVENTORY;</td>
</tr>
<tr>
<td>LOCATIONS_SETUP_REQD</td>
<td>YOU MUST FIRST_DEFINE_LOCATIONS_LISTSLOCATIONSNEW_BEFORE_YOU_CAN_TRANSFER_INVENTORY;</td>
</tr>
<tr>
<td>LOGIN_DISABLED</td>
<td>LOGIN_ACCESS_HAS_BEEN_DISABLED_FOR_THIS_ROLE;</td>
</tr>
<tr>
<td>LOGIN_DISABLED</td>
<td>YOUR_ACCESS_TO_THIS_ACCOUNT_HAS_BEEN_REMOVED_OR_DISABLED_PLEASE_CONTACT_THE_ACCOUNT_ADMINISTRATOR;</td>
</tr>
<tr>
<td>LOGIN_DISABLED</td>
<td>LOGIN_NAME_AND_PSWD_REQD_PLEASE_ENTER_BOTH_A_USER_NAME_AND_A_PASSWORD;</td>
</tr>
<tr>
<td>LOGIN_DISABLED</td>
<td>INVALID_LOGIN_CUSTOMER_ACCESS_IS_DISABLED;</td>
</tr>
<tr>
<td>LOGIN_DISABLED</td>
<td>PARTNER_CENTER_DISABLED_LOGIN_ADVANCED_PARTNER_CENTER_ACCESS_HAS_BEEN_DISABLED_BY_THE_ACCOUNT_ADMINISTRATOR;</td>
</tr>
<tr>
<td>LOGIN_DISABLED</td>
<td>PARTNER_CENTER_DISABLED_LOGIN_STANDARD_PARTNER_CENTER_ACCESS_HAS_BEEN_DISABLED_BY_THE_ACCOUNT_ADMINISTRATOR;</td>
</tr>
<tr>
<td>LOGIN_EMAIL_REQD</td>
<td>INVALID_LOGIN YOU MUST PROVIDE_AN_EMAIL_ADDRESS;</td>
</tr>
<tr>
<td>MACHN_LIST_KEY_NAMES_REQD</td>
<td>WS_MISSING_MACHINE_LIST_KEY_NAMES;</td>
</tr>
<tr>
<td>MANDATORY_PRD_TYPE_REQD</td>
<td>PLEASE_SELECT_THE_MANDATORY_PERIOD_TYPE;</td>
</tr>
<tr>
<td>MATRIX_INFO_TEMP_LOST</td>
<td>MATRIX_ITEM_INFORMATION_WAS_LOST_THIS_WAS_PROBABLY_DUE_TO_A_TRANSIENT_CONDITION_LIKE_A_SERVER_REBOOT_PLEASE_TRY_AGAIN;</td>
</tr>
<tr>
<td>MERGE_OPERATION_DISALLWD</td>
<td>YOU_CANNOT_PERFORM_MERGE_OPERATIONS_ON_RECORDS_THAT_BELONG_TO_YOUR_GROUP;</td>
</tr>
<tr>
<td>EXCEEDED_MAX_FIELD_LENGTH</td>
<td>WS_EXCEEDED_MAX_RECORDS;</td>
</tr>
<tr>
<td>MAX_RCRDS_EXCEEDED</td>
<td>MEDIA_FILE_WAS_OF_TYPE_JAVASCRIPT_AND_WOULD_NOT_COMPILE_ERROR_ON_LINE;</td>
</tr>
<tr>
<td>MEDIA_NOT_FOUND</td>
<td>MEDIA_ITEM_NOT_FOUND_1;</td>
</tr>
<tr>
<td>MEDIA_NOT_INITIALIZED</td>
<td>MEDIA_ITEM_CANNOT_BE_INITIALIZED;</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MISMTCHED_SEARCH_PARENT PARENTESIS</td>
<td>SEARCH_ERROR_PARENTHESSES_ARE_UNBALANCED;</td>
</tr>
<tr>
<td>MISSING_ACCT_PRD</td>
<td>THERE_ARE_NOT_ENOUGH_PERIODS_SPECIFIED_TO_COMPLETELY_RECOGNIZE_THIS_TRANSACTION;</td>
</tr>
<tr>
<td>MISSNG_ACCT_PRD</td>
<td>UNABLE_TO_FIND_AN_ACCOUNTING_PERIOD_FOR_THE_ALLOCATION_DATE;</td>
</tr>
<tr>
<td>MISSNG_REVICREC_RCDRD</td>
<td>UNABLE_TO_LOCATE_REVENUE_RECOGNITION_RECORDS;</td>
</tr>
<tr>
<td>MISSNG_SO_REV_REC_PARAMS</td>
<td>UNABLE_TO_GET_REVENUE_RECOGNITION_PARAMETERS_FROM_ORIGINATING_SALES_ORDER;</td>
</tr>
<tr>
<td>MISSNG_SO_START_END_DATE S</td>
<td>UNABLE_TO_ACQUIRE_START_AND_END_DATE_FROM_SALES_ORDER;</td>
</tr>
<tr>
<td>MLTPLE_TAX_LINES_DISALLWD</td>
<td>MULTIPLE_TAX_LINES_FOR_LINE_ITEM_IN_TRANSACTION;</td>
</tr>
<tr>
<td>MORIZATION_JE_ALERT</td>
<td>MORIZATION_JOURNAL_ENTRY_ALERT;</td>
</tr>
<tr>
<td>MORIZED_TRAN_CANNOT_DELETE</td>
<td>MORIZED_TRAN_CANNOT_DELETE;</td>
</tr>
<tr>
<td>MORIZED_TRAN_FAILURE_REMEDY</td>
<td>MORIZED_TRAN_FAILURE_REMEDY;</td>
</tr>
<tr>
<td>MORIZED_TRAN_WITH_SAME_NAME</td>
<td>MORIZED_TRAN_WITH_SAME_NAME;</td>
</tr>
<tr>
<td>MSNG_FIELD_OWRTE_MUST_BE_TRUE</td>
<td>MISSINGFIELDOVERWRITE_ATTRIBUTE_REQUIRED_AS_TRUE;</td>
</tr>
<tr>
<td>MULTISELECT_TYPE_REQD</td>
<td>WS_NO_MULTISELECT_TYPE_IS_DEFINED;</td>
</tr>
<tr>
<td>NAME_ALREADY_IN_USE</td>
<td>A_SEARCH HAS_ALREADY_BEEN_SAVED_WITH_THAT_NAME_PLEASE_USE_A_DIFFERENT_NAME;</td>
</tr>
<tr>
<td>NAME_ALREADY_IN_USE</td>
<td>A_MASS_UPDATE HAS_ALREADY_BEEN_SAVED_WITH_THAT_NAME_PLEASE_USE_A_DIFFERENT_NAME;</td>
</tr>
<tr>
<td>NAME_NOT_AUTHORIZED</td>
<td>RT_NAME_NOT_AUTHORIZED_EXCEPTION;</td>
</tr>
<tr>
<td>NAME_TYPE_FLDR_FIELDS_REQD</td>
<td>MISSING_REQUIRED_FIELDS__NAME_TYPE_AND_FOLDER;</td>
</tr>
<tr>
<td>NEGATIVE_TAX_RATE_DISALLWD</td>
<td>A_TAX_RATE_CANNOT_BE_NEGATIVE;</td>
</tr>
<tr>
<td>NO_DATA_FOUND</td>
<td>NO_DATA_WAS_FOUND;</td>
</tr>
<tr>
<td>NO_EXPENSES_FOR_PRD</td>
<td>THE_ALLOCATION_SOURCES_OR_DESTINATIONS_DID_NOT_HAVE_ANY_EXPENSES_ASSOCIATED_WITH_THEM_FOR_THE_SELECTED_PERIOD;</td>
</tr>
<tr>
<td>NO_MTRX_ITEMS_TO_UPDATE</td>
<td>THERE_ARE_NO_MATRIX_SUBITEMS_TO_UPDATE;</td>
</tr>
<tr>
<td>NO_ORD_SHPMNT</td>
<td>THERE_IS_NO_SHIPMENT_ON_THAT_ORDER;</td>
</tr>
<tr>
<td>NO_RCRD_FOR_USER</td>
<td>THERE_IS_NO_RECORD_FOR_THIS_USER_IN_THE_COMPANYS_ENTITY_TABLE_EMAILLOGINESEMAIL1KENTITY2;</td>
</tr>
<tr>
<td>NO_SCHEDUL_APPLIED</td>
<td>THERE_WERE_NO_SCHEDULES_THAT_NEED_TO_APPLIED_TO_THE_GIVEN_PERIOD;</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NO_SCHDUL_APPLIED</td>
<td>THERE WERE NO SCHEDULES THAT NEED TO BE APPLIED TO THE INPUT ACCOUNTING PERIOD;</td>
</tr>
<tr>
<td>NONZERO_AMT_REQD</td>
<td>YOU DID NOT ENTER NONZERO AMOUNTS FOR ANY ACCOUNTS;</td>
</tr>
<tr>
<td>NOT_IN_INVT</td>
<td>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;</td>
</tr>
<tr>
<td>NONMATCHING_EMAILS</td>
<td>EMAIL ADDRESSES DON'T MATCH;</td>
</tr>
<tr>
<td>NULL_CHECK_NUMBER</td>
<td>NULL CHECK NUMBER;</td>
</tr>
<tr>
<td>NUM_ALREADY_USED</td>
<td>ECK NUMBER X IS ALREADY USED;</td>
</tr>
<tr>
<td>NUM_ITEMS_GRTR_THAN_QTY</td>
<td>NUMBER OF ITEM NUMBERS ENTERED 1 IS GREATER THAN THE ITEM QUANTITY 2;</td>
</tr>
<tr>
<td>NUM_ITEMS_NOT_EQUAL_TO_QTY</td>
<td>NUMBER OF ITEM NUMBERS ENTERED 1 IS NOT EQUAL TO THE ITEM QUANTITY 2;</td>
</tr>
<tr>
<td>NUM_REQD_FOR_FIRST_LABEL</td>
<td>NO NUMBER WAS SPECIFIED FOR THE FIRST LABEL;</td>
</tr>
<tr>
<td>ONE_ADMIN_REQD_PER_ACCT</td>
<td>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;</td>
</tr>
<tr>
<td>ONE_ADMIN_REQD_PER_ACCT</td>
<td>YOU CAN'T DELETE THIS EMPLOYEE NO ADMINISTRATORS FOR THIS ACCOUNT WOULD REMAIN;</td>
</tr>
<tr>
<td>ONE_ADMIN_REQD_PER_ACCT</td>
<td>YOU CAN'T INACTIVATE 1 THE ACCOUNT WOULD BE LEFT WITH NO ACTIVE ADMINISTRATORS;</td>
</tr>
<tr>
<td>ONE_ADMIN_REQD_PER_ACCT</td>
<td>YOU CAN'T REMOVE THE ADMINISTRATOR ROLE FROM THIS USER NO ADMINISTRATORS FOR THIS ACCOUNT WOULD REMAIN;</td>
</tr>
<tr>
<td>ONE_EMPL_REQD</td>
<td>AT LEAST ONE EMPLOYEE IS REQUIRED TO PROCESS PAYROLL;</td>
</tr>
<tr>
<td>ONE_RCRD_REQD_FOR_MASS_UPDATE</td>
<td>PLEASE CREATE AT LEAST ONE BEFORE USING THIS MASS UPDATE;</td>
</tr>
<tr>
<td>ONE_ROLE_REQD</td>
<td>YOU CAN'T INACTIVATE ALL ROLES YOU WOULD NOT BE ABLE TO LOG IN;</td>
</tr>
<tr>
<td>ONLINE_BILL_PAY_SETUP_REQD</td>
<td>B1B IS NOT SET UP FOR ONLINE BILL PAY TO SET UP THIS PAYEE CLICK GO BACK;</td>
</tr>
<tr>
<td>ONLINE_FORM_ID_REQD</td>
<td>MISSING REQUIRED ONLINE FORM ID;</td>
</tr>
<tr>
<td>ONLINE_ORD_FEATURE_DISABLED</td>
<td>CANT_OPEN_STORE FOR 1 THIS COMPANY DOES NOT HAVE THE ONLINE_ORDERING FEATURE ENABLED THE FEATURE IS REQUIRED FOR CUSTOMERS TO MAKE ONLINE PURCHASES;</td>
</tr>
<tr>
<td>ONLY_ONE_CONTRIB_ITEM_REQD</td>
<td>ONLY ONE INSTANCE OF A COMPANY CONTRIBUTION ITEM IS ALLOWED ON AN EMPLOYEE RECORD;</td>
</tr>
<tr>
<td>ONLY_ONE_DEDCT_ITEM_REQD</td>
<td>ONLY ONE INSTANCE OF A DEDUCTION ITEM IS ALLOWED ON AN EMPLOYEE RECORD;</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ONLY_ONE_DISTRIBUTED_ALLWD</td>
<td>YOU MAY NOT DISTRIBUTE 1 NUMBERS MORE THAN ONCE. YOU ATTEMPTED TO DISTRIBUTE THE FOLLOWING 1 NUMBERS MORE THAN ONCE.</td>
</tr>
<tr>
<td>ONLY_ONE_EARNING_ITEM_REQD</td>
<td>ONLY ONE INSTANCE OF AN EARNING ITEM IS ALLOWED ON AN EMPLOYEE RECORD.</td>
</tr>
<tr>
<td>ONLY_ONE_UNIT_AS_BASE_UNIT</td>
<td>ONLY ONE UNIT MAY BE DESIGNATED AS THE BASE UNIT.</td>
</tr>
<tr>
<td>ONLY_ONE_UPLOAD_ALLWD</td>
<td>YOU CANNOT UPLOAD MORE THAN ONE FILE AT A TIME.</td>
</tr>
<tr>
<td>ONLY_ONE_WITHHOLD_ITEM_REQD</td>
<td>ONLY ONE INSTANCE OF A WITHHOLDING ITEM IS ALLOWED ON AN EMPLOYEE RECORD.</td>
</tr>
<tr>
<td>ORD_ALREADY_APPROVED</td>
<td>YOU CANNOT CANCEL THIS ORDER BECAUSE IT HAS ALREADY BEEN APPROVED.</td>
</tr>
<tr>
<td>ORDER_DOES_NOT_EXIST</td>
<td>THAT ORDER DOES NOT EXIST.</td>
</tr>
<tr>
<td>PACKAGE_WEIGHT_REQD</td>
<td>ATTEMPTED TO CREATE A PACKAGE WITHOUT SPECIFYING A NONZERO PACKAGE WEIGHT.</td>
</tr>
<tr>
<td>PARENT_CANT_BE_MEMBER</td>
<td>PARENT_ITEM_CAN_NOT_BE_A_MEMBER_OF_ITSELF.</td>
</tr>
<tr>
<td>PARTNER_ACCESS_DENIED</td>
<td>PARTNERS DO NOT HAVE ACCESS TO THIS ITEM.</td>
</tr>
<tr>
<td>PARTNER_CODE_ALREADY_USED</td>
<td>A PARTNER WITH THAT PARTNER CODE ALREADY EXISTS.</td>
</tr>
<tr>
<td>PAYCHECK_IN_USE</td>
<td>YOU CANNOT CLEAR THIS PAYCHECK BECAUSE IT IS LINKED TO BY ONE OR MORE LIABILITY PAYMENTS. YOU MUST DELETE OR VOID THOSE TRANSACTIONS FIRST.</td>
</tr>
<tr>
<td>PAYEE_REQD</td>
<td>YOU MUST SPECIFY A PAYEE.</td>
</tr>
<tr>
<td>PAYEE_REQD_FOR_PMT</td>
<td>YOUR PAYMENT HAS BEEN RECORDED BUT AN ONLINE BILL PAYMENT WILL NOT BE MADE BECAUSE NO PAYEE WAS SPECIFIED YOU SHOULD RETURN TO THE PAYMENT SCREEN IF YOU WISH TO PRINT THE CHECK.</td>
</tr>
<tr>
<td>PAYROLL_EXPENSE_ACCT_REQD</td>
<td>PLEASE SELECT A EXPENSE ACCOUNT FOR PAYROLL ITEM.</td>
</tr>
<tr>
<td>PAYROLL_EXPENSE_ACCT_REQD</td>
<td>PLEASE SELECT AN EXPENSE ACCOUNT FOR PAYROLL ITEM.</td>
</tr>
<tr>
<td>PAYROLL_FEATURE_DISABLED</td>
<td>YOU HAVE NOT ENABLED THE PAYROLL FEATURE.</td>
</tr>
<tr>
<td>PAYROLL_FEATURE_UNAVAILABLE</td>
<td>YOU ARE TRYING TO EDIT A PAY CHEQUE PAYROLL IS NOT AVAILABLE IN NETLEDGER CANADA.</td>
</tr>
<tr>
<td>PAYROLL_ITEM_DELETE_DISALLOWED</td>
<td>UNABLE TO REMOVE PAYROLL ITEM 1 THERE ARE EXISTING TRAN ACTIONS FOR THIS PAYROLL ITEM YOU MAY MARK IT INACTIVE INSTEAD.</td>
</tr>
<tr>
<td>PAYROLL_LIABILITY_ACCT_REQD</td>
<td>PLEASE SELECT A LIABILITY ACCOUNT FOR PAYROLL ITEM.</td>
</tr>
<tr>
<td>PERMISSION_VIOLATION</td>
<td>PERMISSION VIOLATION YOU MAY NO LONGER EDIT THIS RECORD.</td>
</tr>
<tr>
<td>PERMISSION_VIOLATION</td>
<td>PERMISSION VIOLATION YOU MAY NOT ACCESS THIS RECORD.</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PHONE_NUM_REQD</td>
<td>PLEASE_PROVIDE_A_PHONE_NUMBER;</td>
</tr>
<tr>
<td>PLAN_IN_USE</td>
<td>THIS_PLAN_HAS_ALREADY_BEEN_USED_TO_GENERATE_COMMISSION_CALCULATIONS_AND_CANT_BE_DELETED;</td>
</tr>
<tr>
<td>PMT_ALREADY_SBMTD</td>
<td>THIS_PAYMENT_HAS_ALREADY_SUBMITTED_FOR_ONLINE_BILL_PAY;</td>
</tr>
<tr>
<td>PMT_EDIT_DISALLWD</td>
<td>THIS LIABILITY_PAYMENT_CANNOT_BE_EDITED WHILE IT HAS AN AUTOMATEDCLEARING_HOUSE_TRANSITION_IN_PROCESS;</td>
</tr>
<tr>
<td>POSITIVE_QTY_REQD</td>
<td>ASSEMBLY_MEMBER_ITEMS_MUST_HAVE_POSITIVE QUANTITIES;</td>
</tr>
<tr>
<td>POSTING_PRD_SETUP_REQD</td>
<td>CREATION_OF_JOURNAL_ENTRIES_REQUIRE_A_SINGLE_ACCOUNTING_PERIOD_VALUE_ACROSS_ALL_REVENUE_RECOGNITION_EVENTS_PLEASE_SETUP_A_POSTING_PERIOD_FILTER;</td>
</tr>
<tr>
<td>PRD_SETUP_REQD</td>
<td>YOU MUST_CHANGE_YOUR_PERIOD_DEFINITIONS_TO_CONTAIN_FISCAL_YEARS_PLEASE_VISIT_SETUPMANAGE_ACCOUNTING_PERIODS_AND_CLICK_SET_UP_YEAR;</td>
</tr>
<tr>
<td>PRD_SETUP_REQD</td>
<td>YOU MUST_DEFINE_THE_PERIODS_OF_THE_PRIOR_FISCAL_YEAR_PLEAS_VISIT_SETUPMANAGE_ACCOUNTING_PERIODS_AND_CLICK_SET_UP_YEAR;</td>
</tr>
<tr>
<td>PREF_VENDOR_COST_REQD</td>
<td>DROP_SHIPSPECIAL_ORDER_ITEMS_MUST_HAVE_A_PREFERRED_VENDOR_AND_A_COST;</td>
</tr>
<tr>
<td>PREFERRED_TAX_AGENCY_REQD</td>
<td>A_PREFERRED_TAX_AGENCY_HAS BEEN_DELETED_PLEASE_CHOOSE_A_NEW_ONE;</td>
</tr>
<tr>
<td>PREFERRED_TAX_AGENCY_REQD</td>
<td>ERROR_NO_PREFERRED_TAX_AGENCIES_HAVE BEEN_SET_UP;</td>
</tr>
<tr>
<td>PRDS_DISALLWD_NAMES_NOT_UNIQUE</td>
<td>AFTER ADDING_NEW_PERIODS_NOT_ALL_NAMES WOULD BE UNIQUE;</td>
</tr>
<tr>
<td>PRIVATE_RCRD_ACCESS_DISALLWD</td>
<td>YOU_CANNOT_VIEW_OR_EDIT_THIS_RECORD_BEGUEASE_IT IS MARKED_PRIVATE;</td>
</tr>
<tr>
<td>PRIVATE_STATUS_CHNG_DISALLWD</td>
<td>YOU_CANNOT_MAKE_THIS_CONTACT_PRIVATE;</td>
</tr>
<tr>
<td>PSWD_REQD</td>
<td>YOU MUST PROVIDE_A_PASSWORD_TO_GIVE_THIS PERSON_ACCESS_TO_YOUR ACCOUNT;</td>
</tr>
<tr>
<td>PSWD_REQD</td>
<td>PASSWORD IS EMPTY;</td>
</tr>
<tr>
<td>PSWD_REQD</td>
<td>PLEASE_TYPE_YOUR_PASSWORD INTO BOTH_FIELDS;</td>
</tr>
<tr>
<td>PWSDS_DONT_MATCH</td>
<td>PASSWORDS_DONT_MATCH;</td>
</tr>
<tr>
<td>PWSDS_DONT_MATCH</td>
<td>THE PASSWORDS YOU ENTERED DO NOT MATCH PLEASE REENTER YOUR PASSWORDS;</td>
</tr>
<tr>
<td>PWSDS_DONT_MATCH</td>
<td>THE PASSWORDS YOU HAVE ENTERED DO NOT MATCH NEW_PASSWORDS_DONT_MATCH;</td>
</tr>
<tr>
<td>RCRD_DSNT_EXIST</td>
<td>THAT RECORD DOES NOT EXIST;</td>
</tr>
<tr>
<td>RCRD_PREVSLY_DELETED</td>
<td>THIS RECORD HAS ALREADY BEEN_DELETED;</td>
</tr>
<tr>
<td>RCRD_PREVSLY_DELETED</td>
<td>THIS RECORD HAS BEEN DELETED SINCE THE LIST WAS GENERATED;</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RCRD_TYPE_REQD</td>
<td>WS_MISSING_RECORD_TYPE;</td>
</tr>
<tr>
<td>RCRD_UNEDITABLE</td>
<td>THAT_RECORD_IS_NOT_EDITABLE;</td>
</tr>
<tr>
<td>READY_MEMORIZED_TRANS</td>
<td>READY_MEMORIZED_TRAN;</td>
</tr>
<tr>
<td>RECOGNIZED_METHOD</td>
<td>RECOGNIZED_METHOD;</td>
</tr>
<tr>
<td>RECUR_EVENT_DISALLWD</td>
<td>A_YEARLY_EVENT_CANNOT_BE_ON_THE_29TH_OF_FEBRUARY;</td>
</tr>
<tr>
<td>REG_CRRNCY_NOT_BASE_CRRNCY</td>
<td>THE_CURRENCY_YOU_ARE_REGISTERED_TO_USE_IS_DIFFERENT_FROM_THE_BASE_CURRENCY_OF_THIS_COMPANY;</td>
</tr>
<tr>
<td>REQUEST_PARAM_REQD</td>
<td>THIS_REQUEST_IS_MISSING_A_REQUIRED_PARAMETER;</td>
</tr>
<tr>
<td>REV_REC_DATE_REQD</td>
<td>NO_REVENUE_RECOGNITION_START_DATE_SPECIFIED;</td>
</tr>
<tr>
<td>REVERSAL_DATE_WARNING</td>
<td>REVERSAL_DATE_WARNING;</td>
</tr>
<tr>
<td>ROLE_REQUIRED</td>
<td>In order to login, a role is required unless a default has been previously set.</td>
</tr>
<tr>
<td>ROUNGING_DIFF_TOO_BIG</td>
<td>ROUNGING_DIFFERENCE_TOO_BIG__TAX1.1 TAX2.2;</td>
</tr>
<tr>
<td>ROUNGING_ERROR</td>
<td>ROUNGING_ERROR_1;</td>
</tr>
<tr>
<td>SCHEDUL_EDIT_DISALLWD</td>
<td>THIS_SCHEDULE_CANNOT_BE_EDITED_AS_IT_HAS_ALREADY_BEEN_USED_FOR_COMMISSION_CALCULATIONS_PLEASE_GO_BACK_AND_SELECT_SAVE_AS_NEW_INSTEAD;</td>
</tr>
<tr>
<td>SEARCH_INTEGER_REQD</td>
<td>PLEASE_ENTER_AN_INTEGER_NUMBER_TO_SEARCH_ON;</td>
</tr>
<tr>
<td>SECTIONS_UNEDITABLE</td>
<td>IILTIN_SECTIONS_CANNOT_BE_EDITED_EXCEPTION;</td>
</tr>
<tr>
<td>SECURE_TRANS_REQD_ON_CHECKOUT</td>
<td>STORE_SERVER_ERROR_AS_CONFIGURED_THIS_SERVER_DOES_NOT_PERMIT_SECURE_TRANSACTIONS_REQUIRED_BY_STORE_CHECKOUT;</td>
</tr>
<tr>
<td>SESSION_TERMD_2ND_LOGIN_DECTD</td>
<td>SOMEONE_HAS_LOGGED_IN_AS_THIS_USER_FROM_A_DIFFERENT_COMPUTER_OR_BROWSER_WINDOW_ONLY_ONE_PERSON_MAY_LOGIN_AS_A_GIVEN_USER_AT_A_TIME_AS_A_CONSEQUENCE_THIS_SESSION_HAS_BEEN_TERMINATED_PTO_HAVE_MULTIPLE_WINDOWS_LOGGED_IN_AS_THESAME_USER_USE_FILENEWWINDOW_FRO;</td>
</tr>
<tr>
<td>SESSION_TERMD_2ND_LOGIN_DECTD</td>
<td>YOU_CAN_HAVE_A_MAXIMUM_OF_1_ACTIVE_USERS_AT_A_TIME_IN_2;</td>
</tr>
<tr>
<td>SESSION_TERMD_2ND_LOGIN_DECTD</td>
<td>YOU_CAN_HAVE_A_MAXIMUM_OF_1_ACTIVE_USERS_AT_A_TIME_THATARE_ENABLED_FOR_OFFLINE SALES_CLIENT;</td>
</tr>
<tr>
<td>SESSION_TIMED_OUT</td>
<td>WS_SESSION_TIMED_OUT;</td>
</tr>
<tr>
<td>SESSION_TIMED_OUT</td>
<td>YOUR_SESSION_HAS_TIMED_OUT_PLEASE_REENTER_YOUR_INFORMATION_AND_TRY_AGAIN;</td>
</tr>
<tr>
<td>SESSION_TIMED_OUT</td>
<td>YOUR_CONNECTION_HAS_TIMED_OUT__PLEASE_A_HREFPAGESLOGINJS_P_TARGETSELFLOG_INA_AGAIN;</td>
</tr>
<tr>
<td>SO_HAS_CHILD_TRANS</td>
<td>SALEOrderId_HAS_CHILD_TRANSACTIONS;</td>
</tr>
<tr>
<td>START_DATE_AFTER_END_DATE</td>
<td>THE_START_DATE_MUST_PRECEED_THE_END_DATE;</td>
</tr>
<tr>
<td>STORAGE_LIMIT_EXCEEDED</td>
<td>YOU_ENTERED_A_VALUE_THAT_WILL_EXCEED_THE_INTERNAL_STORAGE_LIMIT_OF_1_PLEASE_REDUCE_THE_NUMBER;</td>
</tr>
<tr>
<td>STORE_ALIAS_UNAVAILABLE</td>
<td>THE_STORE_ALIAS_YOU_CHOSE_1_IS_ALREADY_TAKEN__PLEASE_GO_BACK_AND_CHOOSE_ANOTHER;</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>STORE_DOMAIN_UNAVAILABLE</td>
<td>THE_STORE_DOMAIN_NAME_YOU_CHOSE_0_IS_ALREADY_TAKEN_PLEASE_GO_BACK_AND_CHOOSE_ANOTHER;</td>
</tr>
<tr>
<td>STORE_SERVER_ERROR</td>
<td>STORE_SERVER_ERROR_AS_CONFIGURED_THIS_SERVER_DOES_NOT_PERMIT_SHOPPING;</td>
</tr>
<tr>
<td>SUBITEM_REQD</td>
<td>YOU_MUST_FIRST_SELECT_THE_NEW_SUBITEMS_ON_THE_MATRIX_TAB_YOU_WANT_TO_ADD;</td>
</tr>
<tr>
<td>SUCCESS_TRANS</td>
<td>THE_TRANSACTION_WAS_ENTERED_1_SUCCESSFULLY_2;</td>
</tr>
<tr>
<td>TAX_CODES_SETUP_PROBLEM</td>
<td>THE_COMPANY_IS_NOT_USABLE_ADMINISTRATOR_HASNT_SET_UP_THE_TAX_CODES;</td>
</tr>
<tr>
<td>TAX_CODES_SETUP_REQD</td>
<td>CANT_OPEN_STORE_FOR_1_THIS_COMPANY_DOES_NOT_HAVE_TAX_CODES_FULLY_SET_UP_THIS_IS_REQUIRED_TO_PROPERLY_CALCULATE_TAXES_ON INTERNATIONAL OTHERPROVINCE AND SAMEPROVINCE_ORDERS;</td>
</tr>
<tr>
<td>TICKET_NOT_LOCATED</td>
<td>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;</td>
</tr>
<tr>
<td>TRANS_AMTS_UNBALNCD</td>
<td>TRANSACTION_IS_NOT_IN_BALANCE__AMOUNTSTAXESSHIPPING_1_TO_TAL_AMOUNT_2;</td>
</tr>
<tr>
<td>TRANS_APPLIED_AMTS_UNBALNCD</td>
<td>TRANSACTION_IS_NOT_IN_BALANCE__TOTAL_TO_APPLY_OF_1_DOES_NOT_EQUAL_SUM_OF_APPLIED_2_AND_UNAPPLIED_3;</td>
</tr>
<tr>
<td>TRANS_CLASS_UNBALNCD</td>
<td>TRANSACTION_WAS_NOT_IN_BALANCE_FOREIGN_CURRENCY_TOTAL__1;</td>
</tr>
<tr>
<td>TRANS_DEPT_UNBALNCD</td>
<td>TRANSACTION_OUT_OF_BALANCE_FOR_DEPARTMENT_1_TOTAL__2;</td>
</tr>
<tr>
<td>TRANS_DOES_NOT_EXIST</td>
<td>NO_TRANSACTION_EXISTS_FOR_THAT_ENTITY;</td>
</tr>
<tr>
<td>TRANS_EDIT_DISALLOWED</td>
<td>YOU_CANNOT_EDIT_THIS_TRANSACTION_1_DOES_NOT_SUPPORT_THE_IMPORTED_TRANSACTION;</td>
</tr>
<tr>
<td>TRANS_EDIT_DISALLOWED</td>
<td>THIS_TRANSACTION_IS_IN_A_PERIOD_THAT_HAS_BEEN_CLOSED_YOU_MAY_NOT_EDIT_IT;</td>
</tr>
<tr>
<td>TRANS_FORGN_CUR_UNBALNCD</td>
<td>TRANSACTION_WAS_NOT_IN_BALANCE_FOREIGN_CURRENCY_POSTING__TOTAL__1;</td>
</tr>
<tr>
<td>TRANS_IN_USE</td>
<td>THIS_TRANSACTION_CANNOT_BE_DELETED_Because_IT_IS_LINKED_TO_BY_ONE_OR_MORE_TRANSACTIONS;</td>
</tr>
<tr>
<td>TRANS_LINE_AND_PMT_UNBALNCD</td>
<td>TRANSACTION_IS_NOT_IN_BALANCE__LINE_ITEM_SUM_OF_1_DOES_NOT_EQUAL_TO_PAYMENT_AMOUNT_2;</td>
</tr>
<tr>
<td>TRANS_LINES_UNBALNCD</td>
<td>TRANSACTION_IS_NOT_IN_BALANCE__LINE_ITEM_SUM_OF_1_DOES_NOT_EQUAL_AMOUNT_OF_2;</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TRANS_LINES_UNBALNCD</td>
<td>TRANSACTION_IS_NOT_IN_BALANCE__LINE_ITEM_SUM_OF_1_DOES_NOT_EQUAL_APPLIED_AMOUNT_OF_2;</td>
</tr>
<tr>
<td>TRANS_LOC_UNBALNCD</td>
<td>TRANSACTION_OUT_OF_BALANCE_FOR_LOCATION_1_TOTAL_2;</td>
</tr>
<tr>
<td>TRANS_NOT_CLEANED</td>
<td>TRANSACTION_NOT_CLEANED_UP;</td>
</tr>
<tr>
<td>TRANS_NOT_COMPLETED</td>
<td>TRANSACTION_WAS_NOT_COMPLETE;</td>
</tr>
<tr>
<td>TRANS_UNBALNCD</td>
<td>TRANSACTION_IS_NOT_IN_BALANCE_1;</td>
</tr>
<tr>
<td>TRANS_UNBALNCD</td>
<td>TRANSACTION_IS_NOT_IN_BALANCE_12__OTHERCOUNT__3;</td>
</tr>
<tr>
<td>TRANS_UNBALNCD</td>
<td>TRANSACTION_WAS_NOT_IN_BALANCE_POSTING_TOTAL__1;</td>
</tr>
<tr>
<td>TRANS_UNBALNCD</td>
<td>TRANSACTION_WAS_NOT_IN_BALANCE_TOTAL__1;</td>
</tr>
<tr>
<td>TRANSACTION_DELETED</td>
<td>THE_TRANSACTION_YOU_ARE_ATTEMPTING_TO_ACCESS_HAS BEEN_DELETED;</td>
</tr>
<tr>
<td>TYPE_NOT_RECGNZD</td>
<td>DIA_TYPE_NOT_RECOGNIZED;</td>
</tr>
<tr>
<td>UNDEFINED_ACCTNG_PRD</td>
<td>THE_ACCOUNTING_PERIOD_RANGE_1_HAS_NOT_BEEN_DEFINED_PLEASE_VISIT_A_HREFAPPSETUPPERIODFISCALPERIODSNSNSETUP__ACCOUNTING__MANAGE_ACCOUNTING_PERIODS_DA_TO_DEFINE_THIS_PERIOD_OR_SET_UP_YOUR_YEAR;</td>
</tr>
<tr>
<td>UNDEFINED_ACCTNG_PRD</td>
<td>THE_COMPARISON_ACCOUNTING_PERIOD_RANGE_1_HAS_NOT_BEEN_DEFINED_PLEASE_VISIT_A_HREFAPPSETUPPERIODFISCALPERIODSNSNSETUP__ACCOUNTING__MANAGE_ACCOUNTING_PERIODS_DA_TO_DEFINE_THIS_PERIOD_OR_SET_UP_YOUR_YEAR;</td>
</tr>
<tr>
<td>UNDEFINED_ACCTNG_PRD</td>
<td>THE_DEFAULT_ACCOUNTING_PERIOD_FOR_THIS_REPORT_HAS_NOT_BEEN_DEFINED_PLEASE_VISIT_A_HREFAPPSETUPPERIODFISCALPERIODSNSNSETUP__ACCOUNTING__MANAGE_ACCOUNTING_PERIODS_DA_TO_DEFINE_THIS_PERIOD_OR_SET_UP_YOUR_YEAR;</td>
</tr>
<tr>
<td>UNDEFINED_TAX_PRD</td>
<td>THE_DEFAULT_TAX_PERIOD FOR THIS REPORT HAS NOT BEEN DEFINED_PLEASE_VISIT_A_HREFAPPSETUPPERIODFISCALPERIODSNSNSETUP__ACCOUNTING__MANAGE_TAX_PERIODS_DA_TO_DEFINE_THIS_PERIOD_OR_SET_UP_YOUR_YEAR;</td>
</tr>
<tr>
<td>UNEXPECTED_ERROR</td>
<td>AN_UNEXPECTED_ERROR_HAS_OCCURRED__TECHNICAL_SUPPORT_HAS_BEEN_ALERTED_TO_THIS_PROBLEM;</td>
</tr>
<tr>
<td>UNEXPECTED_ERROR</td>
<td>WS_AN_UNEXPECTED_ERROR_OCCURRED;</td>
</tr>
<tr>
<td>UNEXPECTED_ERROR</td>
<td>WS_DTO_JAVA_CLASS_IS_NOT_DEFINED;</td>
</tr>
<tr>
<td>UNEXPECTED_ERROR</td>
<td>WS_FAILED_TO_INSTANTIATE_CLASS;</td>
</tr>
<tr>
<td>UNEXPECTED_ERROR</td>
<td>WS_NO_DTO_CLASS IS DEFINED;</td>
</tr>
<tr>
<td>UNEXPECTED_ERROR</td>
<td>WS_SCHEMA_CREATION_FAILED_1 DOES NOT HAVE A MATCHING_XML_SCHEMA_TYPE;</td>
</tr>
<tr>
<td>UNEXPECTED_ERROR</td>
<td>WS_SERVER_ERROR_MISSING_DATABASE_ENTRIES_IN_WSRECORDELEMENT_AND_WSNAMESPACE_TABLES;</td>
</tr>
<tr>
<td>UNEXPECTED_ERROR</td>
<td>WS_SERVER_ERROR_NO_FORM_REQUEST_CLASS_IS_DEFINED;</td>
</tr>
<tr>
<td>UNEXPECTED_ERROR</td>
<td>AN_ERROR_OCCURRED WHILE PROCESSING ITEM OPTIONS;</td>
</tr>
<tr>
<td>UNEXPECTED_ERROR</td>
<td>ACTION_EXCEPTION;</td>
</tr>
<tr>
<td>Error Code Returned</td>
<td>Long Description or Message</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UNEXPECTED_ERROR</td>
<td>ERROR;</td>
</tr>
<tr>
<td>UNEXPECTED_ERROR</td>
<td>ERROR_1;</td>
</tr>
<tr>
<td>UNEXPECTED_ERROR</td>
<td>PROBLEM_DURING_COMMISSION_CALCULATION;</td>
</tr>
<tr>
<td>UNITS_TYP_IN_USE</td>
<td>THIS_UNITS_TYPE_IS_USED_BY_1_2__YOU_MUST_DELETE_THE_2_AND_ALL_ASSOCIATED_TRANSACTIONS_IN_ORDER_TO_DELETE_THIS_UNITS_TYPE;</td>
</tr>
<tr>
<td>UNKNWN_ALLOCTN_FREQ_TYP</td>
<td>UNABLE_TO_DETERMINE_ALLOCATION_FREQUENCY_TYPE;</td>
</tr>
<tr>
<td>UNKNWN_ALLOCTN_SCHEDUL_FREQ_TYP</td>
<td>UNABLE_TO_DETERMINE_ALLOCATION_SCHEDULE_FREQUENCY_TYPE;</td>
</tr>
<tr>
<td>UNRECOGNIZED_METHOD</td>
<td>UNRECOGNIZED_METHOD_1;</td>
</tr>
<tr>
<td>UNSUPPORTED_DOC_TYP</td>
<td>UPLOAD_UNSUPPORTED_DOC_TYPE_ERROR;</td>
</tr>
<tr>
<td>UNSUPPORTED_WS_VERSION</td>
<td>MESSAGE_CONTAINS_AN_UNSUPPORTED_WEB_SERVICES_VERSION</td>
</tr>
<tr>
<td>UPDATE_DISALLWD</td>
<td>WS_UPDATE_IS_NOT_ALLOWED;</td>
</tr>
<tr>
<td>UPDATE_PRICE_AMT_REQD</td>
<td>PLEASE_SPECIFY_AN_AMOUNT_TO_UPDATE_PRICES;</td>
</tr>
<tr>
<td>URL_ID_PARAM_REQD</td>
<td>URL IS MISSING THE ID_PARAMETER THE_FILE COULD NOT BE RETRIEVED;</td>
</tr>
<tr>
<td>URL_REQD</td>
<td>YOU MUST ENTER A URL FOR THIS MEDIA_ITEM;</td>
</tr>
<tr>
<td>USER_DISABLED</td>
<td>USER_DISABLED;</td>
</tr>
<tr>
<td>VALID_PRD_REQD</td>
<td>INSERT_TRANSACTION_FAILURE_NO_VALID_OPEN_1_PERIOD_FOR_DATE_2;</td>
</tr>
<tr>
<td>VALID_PRD_REQD</td>
<td>UPDATE_TRANSACTION_FAILURE_NO_VALID_OPEN_1_PERIOD_FOR_DATE_2;</td>
</tr>
<tr>
<td>VENDOR_TYPE_REQD</td>
<td>NO_VENDOR_TYPE_WAS_SPECIFIED_IF_CREATING_A_TAX_AGENCY_PLEASE_ENSURE_THAT_THE_VENDOR_TYPE_IS_ACTIVE_AND_MARKED_AS_A_TAX_AGENCY;</td>
</tr>
<tr>
<td>VOIDING_REVERSAL_DISALLWD</td>
<td>YOU_MAY_NOT_CREATE_A_VOIDING_REVERSAL_FOR_TRANSACTIONS_WITH_INVENTORY_IMPACT;</td>
</tr>
<tr>
<td>WS_CONCUR_SESSION_DISALLWD</td>
<td>WSINVALID_CONCURRENT_SESSION;</td>
</tr>
<tr>
<td>WS_FEATURE_REQD</td>
<td>WS_NO_WEB_SERVICES_FEATURE;</td>
</tr>
<tr>
<td>WS_LOG_IN_REQD</td>
<td>WS_LOG_IN_FIRST;</td>
</tr>
</tbody>
</table>

**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 “Handling Errors” on page 48.
## 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, refer to “Web Services to UI Single Login” on page 46.

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Page Label in NetSuite</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDIT_ACCOUNT</td>
<td>New Accounts</td>
<td>/app/accounting/account/account.nl</td>
</tr>
<tr>
<td>EDIT_ACTIVITY</td>
<td>New Activity</td>
<td>/app/crm/calendar/activity.nl</td>
</tr>
<tr>
<td>EDIT_ALLOCATION</td>
<td>Create Allocation Schedules</td>
<td>/app/accounting/transactions/allocation.nl</td>
</tr>
<tr>
<td>EDIT_AMENDW4</td>
<td>Form W-4</td>
<td>/app/common/entity/amendw4.nl</td>
</tr>
<tr>
<td>EDIT_BILLINGSCHEDULE</td>
<td>New Billing Schedule</td>
<td>/app/accounting/otherlists/billingschedule.nl</td>
</tr>
<tr>
<td>EDIT_CALL</td>
<td>New Phone Calls</td>
<td>/app/crm/calendar/call.nl</td>
</tr>
<tr>
<td>EDIT_CAMPAIGN</td>
<td>New Campaigns</td>
<td>/app/crm/marketing/campaign.nl</td>
</tr>
<tr>
<td>EDIT_CAMPAIGNEMAIL</td>
<td>Campaign E-mail Address</td>
<td>/app/crm/marketing/campaignemail.nl</td>
</tr>
<tr>
<td>EDIT_CASEFIELDRULE</td>
<td>New Set Up Case Rules</td>
<td>/app/crm/support/casefieldrule.nl</td>
</tr>
<tr>
<td>EDIT_CASEFORM</td>
<td>New Online Case Forms</td>
<td>/app/crm/support/caseform.nl</td>
</tr>
<tr>
<td>EDIT_CASEISSUE</td>
<td>New Case Issue List</td>
<td>/app/crm/support/caseissue.nl</td>
</tr>
<tr>
<td>EDIT_CASEORIGIN</td>
<td>New Case Origin Types</td>
<td>/app/crm/support/caseorigin.nl</td>
</tr>
<tr>
<td>EDIT_CASEPRIORITY</td>
<td>New Case Priority</td>
<td>/app/crm/support/casepriority.nl</td>
</tr>
<tr>
<td>EDIT_CASESTATUS</td>
<td>New Case Status</td>
<td>/app/crm/support/casestatus.nl</td>
</tr>
<tr>
<td>EDIT_CASETERRITORY</td>
<td>New Manage Case Territories</td>
<td>/app/crm/support/supportterritory.nl</td>
</tr>
<tr>
<td>EDIT_CASETYPE</td>
<td>New Case Types</td>
<td>/app/crm/support/casetype.nl</td>
</tr>
<tr>
<td>EDIT_CLASS</td>
<td>New Classes</td>
<td>/app/common/otherlists/classtype.nl</td>
</tr>
<tr>
<td>EDIT_COLORTHEME</td>
<td>New Color Themes</td>
<td>/app/setup/look/colortheme.nl</td>
</tr>
<tr>
<td>EDIT_COMMISSIONSCHEDULE</td>
<td>New Commission Schedules</td>
<td>/app/crm/sales/commissions/commissionschedule.nl</td>
</tr>
<tr>
<td>EDIT_COMPANY</td>
<td>New Company</td>
<td>/app/common/entity/company.nl</td>
</tr>
<tr>
<td>EDIT_COMPETITOR</td>
<td>Competitor</td>
<td>/app/crm/sales/competitor.nl</td>
</tr>
<tr>
<td>EDIT_CONTACT</td>
<td>New Contacts</td>
<td>/app/common/entity/contact.nl</td>
</tr>
<tr>
<td>EDIT_CRMGROUP</td>
<td>New Groups</td>
<td>/app/crm/common/crmgroup.nl</td>
</tr>
<tr>
<td>EDIT_CRMMESSAGE</td>
<td>New E-mail</td>
<td>/app/crm/common/crmmessage.nl</td>
</tr>
<tr>
<td>EDIT_CRMTEMPLATE</td>
<td>New Marketing Templates</td>
<td>/app/crm/common/merge/marketingtemplate.nl</td>
</tr>
<tr>
<td>EDIT_CURRENCY</td>
<td>New Currencies</td>
<td>/app/common/multicurrency/currency.nl</td>
</tr>
<tr>
<td>EDIT_CUST</td>
<td>Custom Record Entry</td>
<td>/app/common/custom/customrecordentry.nl</td>
</tr>
<tr>
<td>EDIT_CUSTADDRESSFORM</td>
<td>Address Form</td>
<td>/app/common/custom/customadressform.nl?e=T</td>
</tr>
<tr>
<td>Task ID</td>
<td>Page Label in NetSuite</td>
<td>URL</td>
</tr>
<tr>
<td>-------------------------</td>
<td>----------------------------------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>EDIT_CUSTBODYFIELD</td>
<td>New Transaction Body Fields</td>
<td>/app/common/custom/bodycustfield.nl</td>
</tr>
<tr>
<td>EDIT_CUSTCENTER</td>
<td>New Center</td>
<td>/app/common/custom/custcenter.nl</td>
</tr>
<tr>
<td>EDIT_CUSTCOLUMNFIELD</td>
<td>New Transaction Column Fields</td>
<td>/app/common/custom/columncustfield.nl</td>
</tr>
<tr>
<td>EDIT_CUSTEMAILAYOUT</td>
<td>New Transaction Form HTML Layouts</td>
<td>/app/common/custom/custemailayout.nl</td>
</tr>
<tr>
<td>EDIT_CUSTENTITYFIELD</td>
<td>New Entity Fields</td>
<td>/app/common/custom/entitycustfield.nl</td>
</tr>
<tr>
<td>EDIT_CUSTENTRYFORM</td>
<td>New Entry Forms</td>
<td>/app/common/custom/custentryform.nl</td>
</tr>
<tr>
<td>EDIT_CUSTEVENTFIELD</td>
<td>New CRM Fields</td>
<td>/app/common/custom/eventcustfield.nl</td>
</tr>
<tr>
<td>EDIT_CUSTFORM</td>
<td>New Transaction Forms</td>
<td>/app/common/custom/custform.nl</td>
</tr>
<tr>
<td>EDIT_CUSTITEMFIELD</td>
<td>New Item Fields</td>
<td>/app/common/custom/itemcustfield.nl</td>
</tr>
<tr>
<td>EDIT_CUSTJOB</td>
<td>New Customers</td>
<td>/app/common/entity/custjob.nl</td>
</tr>
<tr>
<td>EDIT_CUSTLAYOUT</td>
<td>New Transaction Form PDF Layouts</td>
<td>/app/common/custom/custlayout.nl</td>
</tr>
<tr>
<td>EDIT_CUSTLIST</td>
<td>New Lists</td>
<td>/app/common/custom/custlist.nl</td>
</tr>
<tr>
<td>EDIT_CUSTOMERFIELDRULE</td>
<td>New Set Up Sales Rules</td>
<td>/app/crm/sales/customerfieldrule.nl</td>
</tr>
<tr>
<td>EDIT_CUSTOMERFORM</td>
<td>New Online Customer Forms</td>
<td>/app/crm/sales/leadform.nl</td>
</tr>
<tr>
<td>EDIT_CUSTOMERSTATUS</td>
<td>New Customer Statuses</td>
<td>/app/crm/sales/customerstatus.nl</td>
</tr>
<tr>
<td>EDIT_CUSTOMSCRIPT</td>
<td>New Server Script</td>
<td>/app/common/scripting/customscript.nl</td>
</tr>
<tr>
<td>EDIT_CUSTOMSCRIPTRECORD</td>
<td>New Server Script Record</td>
<td>/app/common/scripting/customscriptrecord.nl</td>
</tr>
<tr>
<td>EDIT_CUSTOTHERFIELD</td>
<td>New Other Field</td>
<td>/app/common/entity/custotherfield.nl</td>
</tr>
<tr>
<td>EDIT_CUSTPROFILE</td>
<td>Customer Profile</td>
<td>/app/common/entity/custprofile.nl?category=billing&amp;sc=4</td>
</tr>
<tr>
<td>EDIT_CUSTRECORD</td>
<td>New Record Types</td>
<td>/app/common/custom/custrecord.nl</td>
</tr>
<tr>
<td>EDIT_CUSTRECORDFIELD</td>
<td>Custom Record Field</td>
<td>/app/common/custom/custreccustfield.nl</td>
</tr>
<tr>
<td>EDIT_CUSTRECORDFORM</td>
<td>New Custom Record Form</td>
<td>/app/common/custom/custrecordform.nl</td>
</tr>
<tr>
<td>EDIT_CUSTSECTION</td>
<td>New Center Tab</td>
<td>/app/common/custom/custsection.nl</td>
</tr>
<tr>
<td>EDIT_CUSTTASKS</td>
<td>Center Links</td>
<td>/app/common/custom/custtasks.nl</td>
</tr>
<tr>
<td>EDIT_DEPARTMENT</td>
<td>New Departments</td>
<td>/app/common/otherlists/departmenttype.nl</td>
</tr>
<tr>
<td>EDIT_EDMITPROFILE</td>
<td>Employee Profile</td>
<td>/app/crm/common/empeditprofile.nl</td>
</tr>
<tr>
<td>EDIT_EMPLCATEGORY</td>
<td>New E-mail Templates</td>
<td>/app/site/setup/emplcategory.nl</td>
</tr>
<tr>
<td>EDIT_EMPLOYEE</td>
<td>New Employees</td>
<td>/app/common/entity/employee.nl</td>
</tr>
<tr>
<td>EDIT_EMPLOYEESFA</td>
<td>New Assign Support Reps</td>
<td>/app/common/entity/employeesfa.nl</td>
</tr>
<tr>
<td>EDIT_ESCALATIONRULE</td>
<td>New Escalation Rules</td>
<td>/app/crm/support/escalationfieldrule.nl</td>
</tr>
<tr>
<td>EDIT_ESCALATIONTERRITORY</td>
<td>New Manage Escalation Assignment</td>
<td>/app/crm/support/escalationterritory.nl</td>
</tr>
<tr>
<td>EDIT_EVENT</td>
<td>New Event</td>
<td>/app/crm/calendar/event.nl</td>
</tr>
<tr>
<td>Task ID</td>
<td>Page Label in NetSuite</td>
<td>URL</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>EDIT_EXPSCATEGORY</td>
<td>New Expense Categories</td>
<td>/app/accounting/otherlists/expscategory.nl</td>
</tr>
<tr>
<td>EDIT_FAXMESSAGE</td>
<td>New Fax Message</td>
<td>/app/crm/common/merge/faxmessage.nl</td>
</tr>
<tr>
<td>EDIT_FAXTEMPLATE</td>
<td>New Fax Templates</td>
<td>/app/crm/common/merge/faxtemplate.nl</td>
</tr>
<tr>
<td>EDIT_FISCALPERIOD</td>
<td>Setup Fiscal Period</td>
<td>/app/setup/period/fiscalperiod.nl</td>
</tr>
<tr>
<td>EDIT_INFOITEM</td>
<td>New Information Items</td>
<td>/app/site/setup/infoitem.nl</td>
</tr>
<tr>
<td>EDIT_ISSUE</td>
<td>New Issue</td>
<td>/app/crm/support/issuedb/issue.nl</td>
</tr>
<tr>
<td>EDIT_ISSUETAG</td>
<td>New Issue Tag</td>
<td>/app/crm/support/issuedb/issuetag.nl</td>
</tr>
<tr>
<td>EDIT_ITEM</td>
<td>New Items</td>
<td>/app/common/item/item.nl</td>
</tr>
<tr>
<td>EDIT_ITEMOPTION</td>
<td>New Transaction Item Options</td>
<td>/app/common/custom/itemoption.nl</td>
</tr>
<tr>
<td>EDIT_KBCATEGORY</td>
<td>New Knowledge Base</td>
<td>/app/site/setup/kbcategory.nl</td>
</tr>
<tr>
<td>EDIT_KPIREPORT</td>
<td>New KPI Report</td>
<td>/app/center/enhanced/kpiexpressreportsetup.nl</td>
</tr>
<tr>
<td>EDIT_LEAD</td>
<td>New Leads</td>
<td>/app/common/entity/custjob.nl?stage=lead</td>
</tr>
<tr>
<td>EDIT_LOCATION</td>
<td>New Locations</td>
<td>/app/common/otherlists/locationtype.nl</td>
</tr>
<tr>
<td>EDIT_MAILMERGE</td>
<td>Bulk Merge</td>
<td>/app/crm/common/merge/mailmerge.nl</td>
</tr>
<tr>
<td>EDIT_MAILMESSAGE</td>
<td>New Word Message</td>
<td>/app/crm/common/merge/mailmessage.nl</td>
</tr>
<tr>
<td>EDIT_MAILTEMPLATE</td>
<td>New Letter Templates</td>
<td>/app/crm/common/merge/mailtemplate.nl</td>
</tr>
<tr>
<td>EDIT_MEDIAITEM</td>
<td>New Media Item</td>
<td>/app/common/media/mediaitem.nl</td>
</tr>
<tr>
<td>EDIT_MEDIAITEMFOLDER</td>
<td>New File Cabinet</td>
<td>/app/common/media/mediaitemfolder.nl</td>
</tr>
<tr>
<td>EDIT_MEMDOC</td>
<td>New Memorized Transactions</td>
<td>/app/accounting/otherlists/memdoc.nl</td>
</tr>
<tr>
<td>EDIT_OTHERLIST</td>
<td>New Other Lists</td>
<td>/app/common/otherlists/otherlist.nl</td>
</tr>
<tr>
<td>EDIT_OTHERNAME</td>
<td>New Other Names</td>
<td>/app/common/entity/othername.nl</td>
</tr>
<tr>
<td>EDIT_PARTNER</td>
<td>New Partners</td>
<td>/app/common/entity/partner.nl</td>
</tr>
<tr>
<td>EDIT_PAYROLLITEM</td>
<td>New Payroll Items</td>
<td>/app/common/item/payrollitem.nl</td>
</tr>
<tr>
<td>EDIT_PDFMESSAGE</td>
<td>New PDF Message</td>
<td>/app/crm/common/merge/pdfmessage.nl</td>
</tr>
<tr>
<td>EDIT_PDFTEMPLATE</td>
<td>New PDF Templates</td>
<td>/app/crm/common/merge/pdftemplate.nl</td>
</tr>
<tr>
<td>EDIT_PERIOD</td>
<td>New Manage Accounting Periods</td>
<td>/app/setup/period/fiscalperiod.nl</td>
</tr>
<tr>
<td>EDIT_PLANASSIGN</td>
<td>New Commission Plan</td>
<td>/app/crm/sales/commissions/planassign.nl</td>
</tr>
<tr>
<td>EDIT_PRESCATEGORY</td>
<td>New Categories</td>
<td>/app/site/setup/prescategory.nl</td>
</tr>
<tr>
<td>EDIT_PROSPECT</td>
<td>New Prospects</td>
<td>/app/common/entity/custjob.nl?stage=prospect</td>
</tr>
<tr>
<td>EDIT_REFERRALCODE</td>
<td>New Promotion Codes</td>
<td>/app/crm/sales/referralcode.nl</td>
</tr>
<tr>
<td>EDIT_RELATEDITEM</td>
<td>New Related Items Group</td>
<td>/app/site/setup/relateditem.nl</td>
</tr>
<tr>
<td>EDIT_RESOLVECONFLICTS</td>
<td>Resolve Conflicts</td>
<td>/app/common/entity/conflictesresolution.nl</td>
</tr>
<tr>
<td>EDIT_RESOURCE</td>
<td>Resource</td>
<td>/app/crm/calendar/resource.nl</td>
</tr>
<tr>
<td>EDIT_REVRECSCHEDULE</td>
<td>New Revenue Recognition Template</td>
<td>/app/accounting/otherlists/revrecschedule.nl</td>
</tr>
<tr>
<td>Task ID</td>
<td>Page Label in NetSuite</td>
<td>URL</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>EDIT_ROLE</td>
<td>New Role</td>
<td>/app/setup/role.nl</td>
</tr>
<tr>
<td>EDIT_RPTGROUP</td>
<td>New Financial Statement Section</td>
<td>/app/reporting/reportgroup.nl</td>
</tr>
<tr>
<td>EDIT_RPTLAYOUT</td>
<td>New Financial Statement Layout</td>
<td>/app/reporting/reportsectionlayout.nl</td>
</tr>
<tr>
<td>EDIT_RSSFEED</td>
<td>New RSS Feed</td>
<td>/app/site/hosting/rssfeed.nl</td>
</tr>
<tr>
<td>EDIT_SALESTERRITORY</td>
<td>New Manage Sales Territories</td>
<td>/app/crm/sales/salesterritory.nl</td>
</tr>
<tr>
<td>EDIT_SAVEDSEARCH</td>
<td>New Saved Search</td>
<td>/app/common/search/search.nl?cu=T&amp;e=F</td>
</tr>
<tr>
<td>EDIT_SEARCH</td>
<td>New Search</td>
<td>/app/common/search/search.nl</td>
</tr>
<tr>
<td>EDIT_SHIPITEM</td>
<td>New Shipping Items</td>
<td>/app/common/item/shipitem.nl</td>
</tr>
<tr>
<td>EDIT_SITEITEMTEMPLAT</td>
<td>Item Template</td>
<td>/app/site/setup/siteitemtemplate.nl</td>
</tr>
<tr>
<td>EDIT_SITEMEDIA</td>
<td>Site Media</td>
<td>/app/site/media/sitemedia.nl</td>
</tr>
<tr>
<td>EDIT_SOLUTION</td>
<td>New Solutions</td>
<td>/app/crm/support/kb/solution.nl</td>
</tr>
<tr>
<td>EDIT_SSCATEGORY</td>
<td>New Publish Saved Search</td>
<td>/app/site/setup/sscategory.nl</td>
</tr>
<tr>
<td>EDITSTOREITEMLISTLAYOUT</td>
<td>New Layouts</td>
<td>/app/site/setup/storeitemlistlayout.nl</td>
</tr>
<tr>
<td>EDIT_STORETAB</td>
<td>New Tabs</td>
<td>/app/site/setup/storetab.nl</td>
</tr>
<tr>
<td>EDIT_SUPPORTCASE</td>
<td>New Cases</td>
<td>/app/crm/support/supportcase.nl</td>
</tr>
<tr>
<td>EDIT_SYSALERT</td>
<td>System Alert</td>
<td>/app/common/otherlists/systemalert.nl</td>
</tr>
<tr>
<td>EDIT_TASK</td>
<td>New Tasks</td>
<td>/app/crm/calendar/task.nl</td>
</tr>
<tr>
<td>EDIT_TAXACCT</td>
<td>Tax Control Account</td>
<td>/app/setup/taxacct.nl</td>
</tr>
<tr>
<td>EDIT_TAXCODE</td>
<td>New Tax Codes</td>
<td>/app/common/item/taxitem.nl?CA=T</td>
</tr>
<tr>
<td>EDIT_TAXITEM</td>
<td>New Sales Tax Items</td>
<td>/app/common/item/taxitem.nl</td>
</tr>
<tr>
<td>EDIT_TAXITEMVAT</td>
<td>New Tax Codes</td>
<td>/app/common/item/taxitem.nl?vat=T</td>
</tr>
<tr>
<td>EDIT_TAXPERIOD</td>
<td>New Manage Tax Periods</td>
<td>/app/setup/period/taxperiod.nl</td>
</tr>
<tr>
<td>EDIT_TAXTYPE</td>
<td>Tax Type</td>
<td>/app/setup/taxtype.nl</td>
</tr>
<tr>
<td>EDIT_TOPIC</td>
<td>New Topics</td>
<td>/app/crm/support/kb/topic.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_BUILD</td>
<td>Build Assemblies</td>
<td>/app/accounting/transactions/build.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_CARDCHRG</td>
<td>Use Credit Card</td>
<td>/app/accounting/transactions/cardchrg.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_CASHRFND</td>
<td>Refund Cash Sales</td>
<td>/app/accounting/transactions/cashrfnd.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_CASHSALE</td>
<td>Enter Cash Sales</td>
<td>/app/accounting/transactions/cashsale.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_CHECK</td>
<td>Write Checks</td>
<td>/app/accounting/transactions/check.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_COMMISNN</td>
<td>Individual Commission</td>
<td>/app/accounting/transactions/commissn.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_CUSTCHRG</td>
<td>Create Statement Charges</td>
<td>/app/accounting/transactions/custchrg.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_CUSTCRED</td>
<td>Issue Credit Memos</td>
<td>/app/accounting/transactions/custcred.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_CUSTDEP</td>
<td>Record Customer Deposits</td>
<td>/app/accounting/transactions/custdep.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_CUSTINVC</td>
<td>Create Invoices</td>
<td>/app/accounting/transactions/custinvc.nl</td>
</tr>
<tr>
<td>Task ID</td>
<td>Page Label in NetSuite</td>
<td>URL</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>EDIT_TRAN_CUSTPYMT</td>
<td>Accept Customer Payments</td>
<td>/app/accounting/transactions/custpymt.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_CUSTRFND</td>
<td>Issue Customer Refund</td>
<td>/app/accounting/transactions/custrfnd.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_DEPAPPL</td>
<td>Apply Customer Deposits</td>
<td>/app/accounting/transactions/depappl.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_DEPOSIT</td>
<td>Make Deposits</td>
<td>/app/accounting/transactions/deposit.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_ESTIMATE</td>
<td>Prepare Estimates</td>
<td>/app/accounting/transactions/estimate.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_EXPRPT</td>
<td>Enter Expense Reports</td>
<td>/app/accounting/transactions/exprept.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_FXREVAL</td>
<td>Revalue Open Currency Balances</td>
<td>/app/accounting/transactions/fxrevalsetup.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_INVADJST</td>
<td>Adjust Inventory</td>
<td>/app/accounting/transactions/invadjst.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_INVDISTR</td>
<td>Distribute Inventory</td>
<td>/app/accounting/transactions/invdistr.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_INVTRNFLR</td>
<td>Transfer Inventory</td>
<td>/app/accounting/transactions/invtrnfr.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_INWKSHT</td>
<td>Adjust Inventory Worksheet</td>
<td>/app/accounting/transactions/inwwksht.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_ITEMRCPCT</td>
<td>New Item Receipt</td>
<td>/app/accounting/transactions/itemrcpt.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_ITEMSHIP</td>
<td>New Item Fulfillment</td>
<td>/app/accounting/transactions/itemship.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_JOURNAL</td>
<td>Make Journal Entries</td>
<td>/app/accounting/transactions/journal.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_JOURNALIMPORT</td>
<td>Import Journal Entries</td>
<td>/app/accounting/transactions/journal.nl?imp=T</td>
</tr>
<tr>
<td>EDIT_TRAN_LIADJST</td>
<td>Liability Adjustment</td>
<td>/app/accounting/transactions/laadjsyt.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_LIAPYMT</td>
<td>Pay Payroll Liabilities</td>
<td>/app/accounting/transactions/liapymt.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_OPPRTNTY</td>
<td>Create Opportunities</td>
<td>/app/accounting/transactions/opprntnty.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_PAYCHECK</td>
<td>Individual Paychecks</td>
<td>/app/accounting/transactions/paycheck.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_PURCHORD</td>
<td>Enter Purchase Orders</td>
<td>/app/accounting/transactions/purchord.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_REPLENISHLOC</td>
<td>Replenish Location</td>
<td>/app/accounting/transactions/replenishloc.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_RTNAUTH</td>
<td>Issue Return Authorizations</td>
<td>/app/accounting/transactions/rtnauth.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_SALESORD</td>
<td>Enter Sales Orders</td>
<td>/app/accounting/transactions/salesord.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_TAXLIAB</td>
<td>Write GST Liability</td>
<td>/app/accounting/transactions/vatliab.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_TAXPYMT</td>
<td>Pay Sales Tax</td>
<td>/app/accounting/transactions/taxpymt.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_TRANSFER</td>
<td>Transfer Funds</td>
<td>/app/accounting/transactions/transfer.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_UNBUILD</td>
<td>Unbuild Assemblies</td>
<td>/app/accounting/transactions/unbuild.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_VATLIABAU</td>
<td>Pay Tax Liability</td>
<td>/app/accounting/transactions/vatliab.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_VENDBILL</td>
<td>Enter Bills</td>
<td>/app/accounting/transactions/vendbill.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_VENDCRED</td>
<td>Enter Vendor Credits</td>
<td>/app/accounting/transactions/vendcred.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_VENDPYMT</td>
<td>Pay Single Vendor</td>
<td>/app/accounting/transactions/vendpymt.nl</td>
</tr>
<tr>
<td>EDIT_TRAN_YTDADJST</td>
<td>Enter Payroll YTD Amounts</td>
<td>/app/accounting/transactions/ytdadjst.nl</td>
</tr>
<tr>
<td>EDIT_TRANSACTIONLIST</td>
<td>Paycheck History</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Paycheck</td>
</tr>
<tr>
<td>EDIT_UNITSTYPE</td>
<td>New Units Of Measure</td>
<td>/app/common/units/unitstype.nl</td>
</tr>
<tr>
<td>Task ID</td>
<td>Page Label in NetSuite</td>
<td>URL</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>EDIT_URLALIAS</td>
<td>New Promotional URL</td>
<td>/app/setup/urlalias.nl</td>
</tr>
<tr>
<td>EDIT_VENDOR</td>
<td>New Vendors</td>
<td>/app/common/entity/vendor.nl</td>
</tr>
<tr>
<td>INTL_SECTION</td>
<td>New Tabs</td>
<td>/internal/admin/section.nl</td>
</tr>
<tr>
<td>INTL_SECTIONS</td>
<td>Tabs</td>
<td>/internal/admin/sections.nl</td>
</tr>
<tr>
<td>INTL_TASKLINKS</td>
<td>New Tasklinks</td>
<td>/internal/admin/tasklink.nl</td>
</tr>
<tr>
<td>INTL_TASKLINKS</td>
<td>Tasklinks</td>
<td>/internal/admin/tasklinks.nl</td>
</tr>
<tr>
<td>LIST_ACCOUNT</td>
<td>Accounts</td>
<td>/app/accounting/account/accounts.nl</td>
</tr>
<tr>
<td>LIST_ACTIVITY</td>
<td>Activity List</td>
<td>/app/crm/calendar/activitylist.nl</td>
</tr>
<tr>
<td>LIST_ALLOCATION</td>
<td>Allocation Schedules</td>
<td>/app/accounting/transactions/allocscheduleslist.nl</td>
</tr>
<tr>
<td>LIST_APPROVEACH</td>
<td>Approve Direct Deposits</td>
<td>/app/accounting/transactions/approveach.nl</td>
</tr>
<tr>
<td>LIST_APPROVEEFT</td>
<td>Approve Electronic Funds Transfers</td>
<td>/app/accounting/transactions/approveeft.nl</td>
</tr>
<tr>
<td>LIST_APPROVEVP</td>
<td>Approve Vendor Payment Transfers</td>
<td>/app/accounting/transactions/approvevp.nl</td>
</tr>
<tr>
<td>LIST_BILLINGSCHEDULE</td>
<td>Billing Schedules</td>
<td>/app/accounting/otherlists/billingschedules.nl</td>
</tr>
<tr>
<td>LIST_BUDGET</td>
<td>Budgets</td>
<td>/app/accounting/transactions/budgetlist.nl</td>
</tr>
<tr>
<td>LIST_BULKOP</td>
<td>Mass Updates</td>
<td>/app/common/bulk/bulkops.nl</td>
</tr>
<tr>
<td>LISTCALENDAR</td>
<td>Calendar/Event</td>
<td>/app/crm/calendar/calendar.nl</td>
</tr>
<tr>
<td>LIST_CALL</td>
<td>Phone Calls</td>
<td>/app/crm/calendar/calllist.nl</td>
</tr>
<tr>
<td>LIST_CAMPAIGN</td>
<td>Campaigns</td>
<td>/app/crm/marketing/campaignlist.nl</td>
</tr>
<tr>
<td>LIST_CAMPAIGNCALENDAR</td>
<td>Campaign Calendar</td>
<td>/app/crm/marketing/campaigncalendar.nl</td>
</tr>
<tr>
<td>LIST_CAMPAIGNEMAIL</td>
<td>Campaign E-mail Addresses</td>
<td>/app/crm/marketing/campaignemails.nl</td>
</tr>
<tr>
<td>LIST_CASEFIELDRULE</td>
<td>Set Up Case Rules</td>
<td>/app/crm/support/casefieldrules.nl</td>
</tr>
<tr>
<td>LIST_CASEFORM</td>
<td>Online Case Forms</td>
<td>/app/crm/support/caseforms.nl</td>
</tr>
<tr>
<td>LIST_CASEISSUE</td>
<td>Case Issue List</td>
<td>/app/crm/support/caseissuelist.nl</td>
</tr>
<tr>
<td>LIST_CASEORIGIN</td>
<td>Case Origin Types</td>
<td>/app/crm/support/caseoriginlist.nl</td>
</tr>
<tr>
<td>LIST_CASEPRIORITY</td>
<td>Case Priorities</td>
<td>/app/crm/support/caseprioritylist.nl</td>
</tr>
<tr>
<td>LIST_CASESTATUS</td>
<td>Case Statuses</td>
<td>/app/crm/support/casestatuslist.nl</td>
</tr>
<tr>
<td>LIST_CASETERRITORY</td>
<td>Manage Case Territories</td>
<td>/app/crm/common/automation/territorymanager.nl?case=T</td>
</tr>
<tr>
<td>LIST_CASETYPE</td>
<td>Case Types</td>
<td>/app/crm/support/casetyplist.nl</td>
</tr>
<tr>
<td>LIST_CCTRAN</td>
<td>View Credit Card Transactions</td>
<td>/app/accounting/transactions/cardtrans.nl</td>
</tr>
<tr>
<td>LIST_CHART_ACCOUNT</td>
<td>Chart of Accounts</td>
<td>/app/accounting/account/accounts.nl?report=T&amp;code=COA</td>
</tr>
<tr>
<td>LIST_CLASS</td>
<td>Classes</td>
<td>/app/common/otherlists/classlist.nl</td>
</tr>
<tr>
<td>LIST_COLORETHME</td>
<td>Color Themes</td>
<td>/app/setup/look/colorthemes.nl</td>
</tr>
<tr>
<td>LIST_COMMISSIONSCHEDULE</td>
<td>Commission Schedule</td>
<td>/app/crm/sales/commissions/commissionsscheds.nl</td>
</tr>
<tr>
<td>Task ID</td>
<td>Page Label in NetSuite</td>
<td>URL</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>LIST_COMPANY</td>
<td>All Companies</td>
<td>/app/common/entity/companylist.nl</td>
</tr>
<tr>
<td>LIST_COMPETITOR</td>
<td>Competitors</td>
<td>/app/crm/sales/competitorlist.nl</td>
</tr>
<tr>
<td>LIST_CONTACT</td>
<td>Contacts</td>
<td>/app/common/entity/contactlist.nl</td>
</tr>
<tr>
<td>LIST_CRMGROUP</td>
<td>Groups</td>
<td>/app/crm/common/crmgrouplist.nl</td>
</tr>
<tr>
<td>LIST_CRMTEMPLATE</td>
<td>Marketing Templates</td>
<td>/app/crm/common/merge/marketingtemplates.nl</td>
</tr>
<tr>
<td>LIST_CURRENCY</td>
<td>Currencies</td>
<td>/app/common/multicurrency/currencylsit.nl</td>
</tr>
<tr>
<td>LIST_CUSTBODYFIELD</td>
<td>Transaction Body Fields</td>
<td>/app/common/custom/bodycustfields.nl</td>
</tr>
<tr>
<td>LIST_CUSTCENTER</td>
<td>Centers</td>
<td>/app/common/custom/custcenters.nl</td>
</tr>
<tr>
<td>LIST_CUSTCOLUMNFIELD</td>
<td>Transaction Column Fields</td>
<td>/app/common/custom/columncustfields.nl</td>
</tr>
<tr>
<td>LIST_CUSTMAILLAYOUT</td>
<td>Transaction Form HTML</td>
<td>/app/common/custom/custmaillayouts.nl</td>
</tr>
<tr>
<td>LIST_CUSTENTITYFIELD</td>
<td>Entity Fields</td>
<td>/app/common/custom/entitycustfields.nl</td>
</tr>
<tr>
<td>LIST_CUSTENTRYFORM</td>
<td>Entry Forms</td>
<td>/app/common/custom/custentryforms.nl</td>
</tr>
<tr>
<td>LIST_CUSTEVENTFIELD</td>
<td>CRM Fields</td>
<td>/app/common/custom/eventcustfields.nl</td>
</tr>
<tr>
<td>LIST_CUSTFIELDTAB</td>
<td>Subtabs</td>
<td>/app/common/custom/custfieldtabs.nl</td>
</tr>
<tr>
<td>LIST_CUSTFORM</td>
<td>Transaction Forms</td>
<td>/app/common/custom/custforms.nl</td>
</tr>
<tr>
<td>LIST_CUSTITEMFIELD</td>
<td>Item Fields</td>
<td>/app/common/custom/itemcustfields.nl</td>
</tr>
<tr>
<td>LIST_CUSTJOB</td>
<td>Customers</td>
<td>/app/common/entity/custjoblist.nl?Customer_STAGE=CUSTOMER</td>
</tr>
<tr>
<td>LIST_CUSTLAYOUT</td>
<td>Transaction Form PDF</td>
<td>/app/common/custom/custlayouts.nl</td>
</tr>
<tr>
<td>LIST_CUSTLIST</td>
<td>Lists</td>
<td>/app/common/custom/custlists.nl</td>
</tr>
<tr>
<td>LIST_CUSTOMCODEFILES</td>
<td>Custom Code Scripts</td>
<td>/app/common/media/mediaitemfolders.nl?folder=-15</td>
</tr>
<tr>
<td>LIST_CUSTOMERFIELDRULE</td>
<td>Set Up Sales Rules</td>
<td>/app/crm/sales/customerfieldrules.nl</td>
</tr>
<tr>
<td>LIST_CUSTOMERFORM</td>
<td>Online Customer Forms</td>
<td>/app/crm/sales/leadforms.nl</td>
</tr>
<tr>
<td>LIST_CUSTOMERSTATUS</td>
<td>Customer Statuses</td>
<td>/app/crm/sales/customerstatuslist.nl</td>
</tr>
<tr>
<td>LIST_CUSTOMSCRIPT</td>
<td>Server Scripts</td>
<td>/app/common/scripting/customscriptlist.nl</td>
</tr>
<tr>
<td>LIST_CUSTOMSCRIPTCALENDAR</td>
<td>Script Schedule</td>
<td>/app/common/scripting/customscriptcalendar.nl</td>
</tr>
<tr>
<td>LIST_CUSTOMSCRIPTRECORD</td>
<td>Server Script Records</td>
<td>/app/common/scripting/customscriptrecordlist.nl</td>
</tr>
<tr>
<td>LIST_CUTOFFIELD</td>
<td>Other Custom Fields</td>
<td>/app/common/custom/othercustfields.nl</td>
</tr>
<tr>
<td>LIST_CUSTRECORD</td>
<td>Record Types</td>
<td>/app/common/custom/custrecords.nl</td>
</tr>
<tr>
<td>LIST_CUSTRECORDFORM</td>
<td>Online Custom Record Forms</td>
<td>/app/common/custom/custrecordforms.nl</td>
</tr>
<tr>
<td>LIST_CUSTSECTION</td>
<td>Center Tabs</td>
<td>/app/common/custom/custsections.nl</td>
</tr>
<tr>
<td>LIST_DEPARTMENT</td>
<td>Departments</td>
<td>/app/common/otherlists/departmentlist.nl</td>
</tr>
<tr>
<td>LIST_DUPLICATES</td>
<td>Mass Duplicate Record Merge</td>
<td>/app/common/entity/manageduplicates.nl</td>
</tr>
<tr>
<td>LIST_EMAILTEMPLATE</td>
<td>E-mail Templates</td>
<td>/app/crm/common/merge/emailtemplates.nl</td>
</tr>
<tr>
<td>Task ID</td>
<td>Page Label in NetSuite</td>
<td>URL</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------</td>
<td>---------------------------------------------------------------------</td>
</tr>
<tr>
<td>LIST_EMPLCATEGORY</td>
<td>Employee Directory</td>
<td>/app/site/setup/emplcategories.nl</td>
</tr>
<tr>
<td>LIST_EMPLOYEE</td>
<td>Employees</td>
<td>/app/common/entity/employeelist.nl</td>
</tr>
<tr>
<td>LIST_ENTITY</td>
<td>All Entities</td>
<td>/app/common/entity/entitylist.nl</td>
</tr>
<tr>
<td>LIST_ESCALATIONRULE</td>
<td>Set Up Escalation Rules</td>
<td>/app/crm/support/escalationfieldrules.nl</td>
</tr>
<tr>
<td>LIST_ESCALATIONTERRITORY</td>
<td>Manage Escalation Assignment</td>
<td>/app/crm/common/automation/territorymanager.nl?escalation=T</td>
</tr>
<tr>
<td>LIST_EVENT</td>
<td>Event List</td>
<td>/app/crm/calendar/eventlist.nl</td>
</tr>
<tr>
<td>LIST_FAXTEMPLATE</td>
<td>Fax Templates</td>
<td>/app/crm/common/merge/faxtemplates.nl</td>
</tr>
<tr>
<td>LIST_FCSITEFOLDER</td>
<td>Web Site Hosting Files</td>
<td>/app/common/media/mediaitemfolders.nl?folder=-100</td>
</tr>
<tr>
<td>LIST_FINCHRG</td>
<td>Assess Finance Charges</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustInvvc&amp;Transaction_FINCHRG=T</td>
</tr>
<tr>
<td>LIST_FORECAST</td>
<td>List Sales Rep Forecast</td>
<td>/app/crm/sales/forecastlist.nl?Forecast_ISTEAM=F</td>
</tr>
<tr>
<td>LIST_IMAGE</td>
<td>Images</td>
<td>/app/common/media/mediaitemfolders.nl?folder=-4</td>
</tr>
<tr>
<td>LIST_INFOITEM</td>
<td>Information Items</td>
<td>/app/site/setup/infoitemlist.nl</td>
</tr>
<tr>
<td>LIST_ISSUE</td>
<td>Issues</td>
<td>/app/crm/support/issuelist.nl</td>
</tr>
<tr>
<td>LIST_ITEM</td>
<td>Items</td>
<td>/app/common/item/itemlist.nl</td>
</tr>
<tr>
<td>LIST_ITEMOPTION</td>
<td>Transaction Item Options</td>
<td>/app/common/custom/itemoptions.nl</td>
</tr>
<tr>
<td>LIST_KBCATEGORY</td>
<td>Knowledge Base</td>
<td>/app/site/setup/kbcategories.nl</td>
</tr>
<tr>
<td>LIST_KPIREPORT</td>
<td>KPI Reports</td>
<td>/app/center/enhanced/kpireports.nl</td>
</tr>
<tr>
<td>LIST_LEAD</td>
<td>Leads</td>
<td>/app/common/entity/customerlist.nl?Customer_STAGE=LEAD</td>
</tr>
<tr>
<td>LIST_LOCATION</td>
<td>Locations</td>
<td>/app/common/otherlists/locationlist.nl</td>
</tr>
<tr>
<td>LIST_MAILMERGE</td>
<td>Merge History</td>
<td>/app/crm/common/merge/mailmergehistory.nl</td>
</tr>
<tr>
<td>LIST_MAILTEMPLATE</td>
<td>Letter Templates</td>
<td>/app/crm/common/merge/mailtemplates.nl</td>
</tr>
<tr>
<td>LIST_MEDIAITEM</td>
<td>Media Items</td>
<td>/app/common/media/mediaitems.nl</td>
</tr>
<tr>
<td>LIST_MEDIAITEMFOLDER</td>
<td>File Cabinet</td>
<td>/app/common/media/mediaitemfolders.nl</td>
</tr>
<tr>
<td>LIST_MEDIAITEMFOLDER_LOG</td>
<td>Job Status</td>
<td>/app/external/xml/upload/uploadlog.nl?displayType=FILECABINET</td>
</tr>
<tr>
<td>LIST_MEMDOC</td>
<td>Memorized Transactions</td>
<td>/app/accounting/otherlists/memdoclist.nl</td>
</tr>
<tr>
<td>LIST_MGRFORECAST</td>
<td>List Sales Manager Forecast</td>
<td>/app/crm/sales/forecastlist.nl?Forecast_ISTEAM=T</td>
</tr>
<tr>
<td>LIST_MYROLES</td>
<td>My Roles</td>
<td>/app/center/myroles.nl</td>
</tr>
<tr>
<td>LIST_OTHERLIST</td>
<td>Other Lists</td>
<td>/app/common/otherlists/otherlists.nl</td>
</tr>
<tr>
<td>LIST_OTHERNAME</td>
<td>Other Names</td>
<td>/app/common/entity/othernames.nl</td>
</tr>
<tr>
<td>LIST_PARTNER</td>
<td>Partners</td>
<td>/app/common/entity/partnerlist.nl</td>
</tr>
<tr>
<td>LIST_PAYROLLBATCH</td>
<td>Payroll Batches</td>
<td>/app/payroll/payrollbatchlist.nl</td>
</tr>
<tr>
<td>Task ID</td>
<td>Page Label in NetSuite</td>
<td>URL</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>LIST_PAYROLLITEM</td>
<td>Payroll Items</td>
<td>/app/common/item/payrollitems.nl</td>
</tr>
<tr>
<td>LIST_PDFTEMPLATE</td>
<td>PDF Templates</td>
<td>/app/crm/common/merge/pdftemplates.nl</td>
</tr>
<tr>
<td>LIST_PERIOD</td>
<td>Manage Accounting Periods</td>
<td>/app/setup/period/fiscalperiods.nl</td>
</tr>
<tr>
<td>LIST_PLANASSIGN</td>
<td>Commission Plan</td>
<td>/app/crm/sales/commissions/commissionplans.nl</td>
</tr>
<tr>
<td>LIST_PLANASSIGNS</td>
<td>Plan Assignments</td>
<td>/app/crm/sales/commissions/planassigns.nl</td>
</tr>
<tr>
<td>LIST_PRESCATEGORY</td>
<td>Categories</td>
<td>/app/site/setup/prescategories.nl?type=PRES</td>
</tr>
<tr>
<td>LIST_PROSPECT</td>
<td>Prospects</td>
<td>/app/common/entity/custjoblist.nl?Customer_STAGE=PROSPECT</td>
</tr>
<tr>
<td>LIST_QUOTA</td>
<td>Quotas</td>
<td>/app/crm/sales/quotalist.nl</td>
</tr>
<tr>
<td>LIST_RECENTRECORDS</td>
<td>Recent Records</td>
<td>/app/common/otherlists/recentrecords.nl</td>
</tr>
<tr>
<td>LIST_RECVDFILES</td>
<td>Attachments Received</td>
<td>/app/common/media/mediaitemfolders.nl?folder=-10</td>
</tr>
<tr>
<td>LIST_REFERRALCODE</td>
<td>Promotion Codes</td>
<td>/app/crm/sales/referralcodes.nl</td>
</tr>
<tr>
<td>LIST_RELATEDITEM</td>
<td>Related Items Group</td>
<td>/app/site/setup/relateditems.nl</td>
</tr>
<tr>
<td>LISTRESOURCE</td>
<td>Resources</td>
<td>/app/accounting/otherlists/revrectemplates.nl</td>
</tr>
<tr>
<td>LIST_REVRECSCHEDULE</td>
<td>Revenue Recognition Templates</td>
<td>/app/accounting/otherlists/revrectemplates.nl</td>
</tr>
<tr>
<td>LIST_ROLE</td>
<td>Manage Roles</td>
<td>/app/setup/allroles.nl</td>
</tr>
<tr>
<td>LIST_RSSFEED</td>
<td>RSS Feeds</td>
<td>/app/site/hosting/rssfeeds.nl</td>
</tr>
<tr>
<td>LIST_SALESTERRITORIES</td>
<td>Sales Territory List</td>
<td>/app/crm/sales/salesterritorylist.nl</td>
</tr>
<tr>
<td>LIST_SALESTERRITORY</td>
<td>Manage Sales Territories</td>
<td>/app/crm/common/automation/territorymanager.nl?sales=T</td>
</tr>
<tr>
<td>LIST_SAVEDASHBOARD</td>
<td>Published Dashboards</td>
<td>/app/center/setup/savedashboards.nl</td>
</tr>
<tr>
<td>LIST_SAVEDBULKOP</td>
<td>Saved Mass Updates</td>
<td>/app/common/bulk/savedbulkops.nl</td>
</tr>
<tr>
<td>LIST_SAVEDSEARCH</td>
<td>Saved Searches</td>
<td>/app/common/search/savedsearches.nl</td>
</tr>
<tr>
<td>LIST_SEARCHRESULTS</td>
<td>Search Results</td>
<td>/app/common/search/searchresults.nl</td>
</tr>
<tr>
<td>LIST_SENTFILES</td>
<td>Attachments Sent</td>
<td>/app/common/media/mediaitemfolders.nl?folder=-14</td>
</tr>
<tr>
<td>LIST_SHIPITEM</td>
<td>Shipping Items</td>
<td>/app/common/item/shipitems.nl</td>
</tr>
<tr>
<td>LIST_SHIPPINGMANIFEST</td>
<td>Shipping Manifest</td>
<td>/app/common/shipping/fedex/shippingmanifest.nl</td>
</tr>
<tr>
<td>LIST_SITEITEMTEMPLAT</td>
<td>Item Templates</td>
<td>/app/site/setup/siteitemtemplates.nl</td>
</tr>
<tr>
<td>LIST_SITETHemes</td>
<td>Site Templates</td>
<td>/app/site/setup/sitetheme.nl</td>
</tr>
<tr>
<td>LIST_SMBIMPORT</td>
<td>SMB Import</td>
<td>/app/external/xml/upload/upload.nl</td>
</tr>
<tr>
<td>LIST_SOLUTION</td>
<td>Solutions</td>
<td>/app/crm/support/kb/solutions.nl</td>
</tr>
<tr>
<td>LIST_SSCATEGORY</td>
<td>Publish Saved Search</td>
<td>/app/site/setup/sscategories.nl</td>
</tr>
<tr>
<td>LIST_STATUSACH</td>
<td>View Direct Deposit Status</td>
<td>/app/accounting/transactions/statusach.nl</td>
</tr>
<tr>
<td>LIST_STATUSIFT</td>
<td>View Electronic Funds Transfer Status</td>
<td>/app/accounting/transactions/statuseft.nl</td>
</tr>
<tr>
<td>LIST_STATUSVP</td>
<td>View Vendor Payment Status</td>
<td>/app/accounting/transactions/statusvp.nl</td>
</tr>
<tr>
<td>Task ID</td>
<td>Page Label in NetSuite</td>
<td>URL</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>LIST_STOREITEMLISTLAYOUT</td>
<td>Layouts</td>
<td>/app/site/setup/storeitemlistlayouts.nl</td>
</tr>
<tr>
<td>LIST_STORETAB</td>
<td>Tabs</td>
<td>/app/site/setup/storetabs.nl</td>
</tr>
<tr>
<td>LIST_SUPPORTCASE</td>
<td>Cases</td>
<td>/app/crm/support/caselist.nl</td>
</tr>
<tr>
<td>LIST_SYSALER T</td>
<td>System Alerts</td>
<td>/app/common/otherlists/systemalerts.nl</td>
</tr>
<tr>
<td>LIST_SYSTEMEMAILTEMPLATE</td>
<td>Setup System E-mail Templates</td>
<td>/app/crm/common/merge/systememailtemplates.nl</td>
</tr>
<tr>
<td>LIST_TASK</td>
<td>Tasks</td>
<td>/app/crm/calendar/tasklist.nl</td>
</tr>
<tr>
<td>LIST_TAXCODE</td>
<td>Tax Codes</td>
<td>/app/common/item/taxitems.nl?CA=T</td>
</tr>
<tr>
<td>LIST_TAXITEM</td>
<td>Sales Tax Items</td>
<td>/app/common/item/taxitems.nl</td>
</tr>
<tr>
<td>LIST_TAXITEMVAT</td>
<td>Sales Tax Items</td>
<td>/app/common/item/taxitems.nl?vat=T</td>
</tr>
<tr>
<td>LIST_TAXPERIOD</td>
<td>Manage Tax Periods</td>
<td>/app/setup/period/taxperiods.nl</td>
</tr>
<tr>
<td>LIST_TEMPLATEFILES</td>
<td>Template Files</td>
<td>/app/common/media/mediaitemfolders.nl?folder=-9</td>
</tr>
<tr>
<td>LIST_TOPIC</td>
<td>Topics</td>
<td>/app/crm/support_kb/topics.nl</td>
</tr>
<tr>
<td>LIST_TRAN_BUILD</td>
<td>Assembly Builds</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Build</td>
</tr>
<tr>
<td>LIST_TRAN_CARDCHRG</td>
<td>Credit Card Charges</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CardChrg</td>
</tr>
<tr>
<td>LIST_TRAN_CASHRFND</td>
<td>Cash Refunds</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CashRfnd</td>
</tr>
<tr>
<td>LIST_TRAN_CASHSALE</td>
<td>Cash Sales</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CashSale</td>
</tr>
<tr>
<td>LIST_TRAN_CHECK</td>
<td>Checks</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Check</td>
</tr>
<tr>
<td>LIST_TRAN_COMMISSN</td>
<td>Individual Commission</td>
<td>/app/accounting/transactions/commissns.nl</td>
</tr>
<tr>
<td>LIST_TRAN_CUSTCHRG</td>
<td>Statement Charges</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustChrg</td>
</tr>
<tr>
<td>LIST_TRAN_CUSTCRED</td>
<td>Credit Memos</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustCred</td>
</tr>
<tr>
<td>LIST_TRAN_CUSTDEP</td>
<td>Customer Deposits</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustDep</td>
</tr>
<tr>
<td>LIST_TRAN_CUSTINVC</td>
<td>Invoices</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustInvc</td>
</tr>
<tr>
<td>LIST_TRAN_CUSTPYMT</td>
<td>Customer Payments</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustPymt</td>
</tr>
<tr>
<td>LIST_TRAN_CUSTRFND</td>
<td>Customer Refunds</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=CustRfnd</td>
</tr>
<tr>
<td>LIST_TRAN_DEPAPPL</td>
<td>Deposit Applications</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=DepAppl</td>
</tr>
<tr>
<td>LIST_TRAN_DEPOSIT</td>
<td>Deposits</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Deposit</td>
</tr>
<tr>
<td>Task ID</td>
<td>Page Label in NetSuite</td>
<td>URL</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>LIST_TRAN_ESTIMATE</td>
<td>Estimates</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Estimate</td>
</tr>
<tr>
<td>LIST_TRAN_EXPREPT</td>
<td>Expense Reports</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=ExpRept</td>
</tr>
<tr>
<td>LIST_TRAN_FXREVAL</td>
<td>Currency Revaluations</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=FxReval</td>
</tr>
<tr>
<td>LIST_TRAN_INVADJUST</td>
<td>Inventory Adjustments</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=InvAdjst</td>
</tr>
<tr>
<td>LIST_TRAN_INVDISTR</td>
<td>Inventory Distributions</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=InvDistr</td>
</tr>
<tr>
<td>LIST_TRAN_INVTRNFR</td>
<td>Inventory Transfers</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=InvTrnfr</td>
</tr>
<tr>
<td>LIST_TRAN_INVWKSHT</td>
<td>Inventory Worksheets</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=InvWksht</td>
</tr>
<tr>
<td>LIST_TRAN_ITEMRCPT</td>
<td>Item Receipts</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=ItemRcpt</td>
</tr>
<tr>
<td>LIST_TRAN_ITEMSHIP</td>
<td>Item Fulfillments</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=ItemShip</td>
</tr>
<tr>
<td>LIST_TRAN_LIABPYMT</td>
<td>Liability Payments</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=LiabPymt</td>
</tr>
<tr>
<td>LIST_TRAN_OPPRTNTY</td>
<td>Opportunities</td>
<td>/app/accounting/transactions/opprtntylist.nl</td>
</tr>
<tr>
<td>LIST_TRAN_PAYCHECK</td>
<td>Paychecks</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Paycheck</td>
</tr>
<tr>
<td>LIST_TRAN_PURCHORD</td>
<td>Purchase Orders</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=PurchOrd</td>
</tr>
<tr>
<td>LIST_TRAN_REORDER</td>
<td>Items Ordered</td>
<td>/app/common/item/itemsordered.nl</td>
</tr>
<tr>
<td>LIST_TRAN_RTNAUTH</td>
<td>Return Authorizations</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=RtnAuth</td>
</tr>
<tr>
<td>LIST_TRAN_SALESORD</td>
<td>Sales Orders</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=SalesOrd</td>
</tr>
<tr>
<td>LIST_TRAN_TAXLIAB</td>
<td>Tax Liabilities</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TaxLiab</td>
</tr>
<tr>
<td>LIST_TRAN_TAXPYMT</td>
<td>Tax Payments</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=TaxPymt</td>
</tr>
<tr>
<td>LIST_TRAN_TRANSFER</td>
<td>Bank Transfers</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Transfer</td>
</tr>
<tr>
<td>LIST_TRAN_UNBUILD</td>
<td>Assembly Unbuilds</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=Unbuild</td>
</tr>
<tr>
<td>LIST_TRAN_VATLIABAU</td>
<td>GST Liabilities</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=VatLiab</td>
</tr>
<tr>
<td>Task ID</td>
<td>Page Label in NetSuite</td>
<td>URL</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>LIST.TRAN.VATLIABUK</td>
<td>VAT Liabilities</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=VatLiab</td>
</tr>
<tr>
<td>LIST.TRAN.VENDBILL</td>
<td>Bills</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=VendBill</td>
</tr>
<tr>
<td>LIST.TRAN.VENDCRED</td>
<td>Bill Credits</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=VendCred</td>
</tr>
<tr>
<td>LIST.TRAN.VENDPYMNT</td>
<td>Bill Payments</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=VendPymt</td>
</tr>
<tr>
<td>LIST.TRAN.YTDADJST</td>
<td>YTD Adjustments</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=YtdAdjst</td>
</tr>
<tr>
<td>LIST.TRANSACTION</td>
<td>Transactions</td>
<td>/app/accounting/transactions/transactionlist.nl?Transaction_TYPE=@ALL@</td>
</tr>
<tr>
<td>LIST.UNCATSITEITEM</td>
<td>View Uncategorized Items</td>
<td>/app/setup/uncatsiteitems.nl</td>
</tr>
<tr>
<td>LIST.UNITSTYPE</td>
<td>Units Of Measure</td>
<td>/app/common/units/unitstypelist.nl</td>
</tr>
<tr>
<td>LIST.UPSELLWIZARD</td>
<td>Upsell Manager</td>
<td>/app/crm/sales/upsell/upsellmanager.nl</td>
</tr>
<tr>
<td>LIST.URLALIASES</td>
<td>Promotional URLs</td>
<td>/app/setup/urlaliases.nl</td>
</tr>
<tr>
<td>LIST.USER</td>
<td>Manage Users</td>
<td>/app/setup/listusers.nl</td>
</tr>
<tr>
<td>LIST_VENDOR</td>
<td>Vendors</td>
<td>/app/common/entity/vendorlist.nl</td>
</tr>
<tr>
<td>SRCH.ACTIVITY</td>
<td>Search Activities</td>
<td>/app/common/search/search.nl?searchtype=Activity</td>
</tr>
<tr>
<td>SRCH.BUDGET</td>
<td>Search Set Up Budget</td>
<td>/app/common/search/search.nl?searchtype=Budget</td>
</tr>
<tr>
<td>SRCH.CALENDAR</td>
<td>Search Calendar/Event</td>
<td>/app/common/search/search.nl?searchtype=Calendar</td>
</tr>
<tr>
<td>SRCH.CALL</td>
<td>Search Calls</td>
<td>/app/common/search/search.nl?searchtype=Call</td>
</tr>
<tr>
<td>SRCH.CAMPAIGN</td>
<td>Search Campaign</td>
<td>/app/common/search/search.nl?searchtype=Campaign</td>
</tr>
<tr>
<td>SRCH.CASE</td>
<td>Search Case</td>
<td>/app/common/search/search.nl?searchtype=Case</td>
</tr>
<tr>
<td>SRCH.CLASS</td>
<td>Search Class</td>
<td>/app/common/search/search.nl?searchtype=Class</td>
</tr>
<tr>
<td>SRCH.COMPANY</td>
<td>Search All Companies</td>
<td>/app/common/search/search.nl?searchtype=Company</td>
</tr>
<tr>
<td>SRCH.COMPETITOR</td>
<td>Search Competitors</td>
<td>/app/common/search/search.nl?searchtype=Competitor</td>
</tr>
<tr>
<td>SRCHCONTACT</td>
<td>Search Contact</td>
<td>/app/common/search/search.nl?searchtype=Contact</td>
</tr>
<tr>
<td>SRCH.CUSTOMER</td>
<td>Search Customer</td>
<td>/app/common/search/search.nl?searchtype=Customer</td>
</tr>
<tr>
<td>SRCH.DEPARTMENT</td>
<td>Search Department</td>
<td>/app/common/search/search.nl?searchtype=Department</td>
</tr>
<tr>
<td>SRCH.DOCUMENT</td>
<td>Search File Cabinet</td>
<td>/app/common/search/search.nl?searchtype=Document</td>
</tr>
<tr>
<td>SRCH.EMPLOYEE</td>
<td>Search Employee</td>
<td>/app/common/search/search.nl?searchtype=Employee</td>
</tr>
<tr>
<td>SRCH.FIRSTVISIT</td>
<td>Search First Site Visit</td>
<td>/app/common/search/search.nl?searchtype=FirstVisit</td>
</tr>
<tr>
<td>SRCH.INFOITEM</td>
<td>Search Information Items</td>
<td>/app/common/search/search.nl?searchtype=InfoItem</td>
</tr>
<tr>
<td>SRCH.ISSUE</td>
<td>Search Issue</td>
<td>/app/common/search/search.nl?searchtype=Issue</td>
</tr>
<tr>
<td>SRCH.ITEM</td>
<td>Search Item</td>
<td>/app/common/search/search.nl?searchtype=Item</td>
</tr>
<tr>
<td>SRCH.JOB</td>
<td>Search Job</td>
<td>/app/common/search/search.nl?searchtype=Job</td>
</tr>
<tr>
<td>Task ID</td>
<td>Page Label in NetSuite</td>
<td>URL</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>SRCH_LEAD</td>
<td>Search Leads</td>
<td>/app/common/search/search.nl?searchtype=Customer&amp;Customer_STAGE=LEAD</td>
</tr>
<tr>
<td>SRCH_LOCATION</td>
<td>Search Location</td>
<td>/app/common/search/search.nl?searchtype=Location</td>
</tr>
<tr>
<td>SRCH_MEDIAITEM</td>
<td>Search Media Items</td>
<td>/app/common/search/search.nl?searchtype=Document</td>
</tr>
<tr>
<td>SRCH_MEMDOC</td>
<td>Search Memorized Transactions</td>
<td>/app/common/search/search.nl?searchtype=MemDoc</td>
</tr>
<tr>
<td>SRCH_PARTNER</td>
<td>Search Partner</td>
<td>/app/common/search/search.nl?searchtype=Partner</td>
</tr>
<tr>
<td>SRCH_PRESCATEGORY</td>
<td>Search Categories</td>
<td>/app/common/search/search.nl?searchtype=PresCategory</td>
</tr>
<tr>
<td>SRCH_PROMOTION</td>
<td>Search Promotion Code</td>
<td>/app/common/search/search.nl?searchtype=Promotion</td>
</tr>
<tr>
<td>SRCH_PROSPECT</td>
<td>Search Prospects</td>
<td>/app/common/search/search.nl?searchtype=Customer&amp;Customer_STAGE=PROSPECT</td>
</tr>
<tr>
<td>SRCH_QUOTA</td>
<td>Search Establish Quota</td>
<td>/app/common/search/search.nl?searchtype=Quota</td>
</tr>
<tr>
<td>SRCH_REVREC</td>
<td>Search Revenue Recognition Schedules</td>
<td>/app/common/search/search.nl?searchtype=RevRec</td>
</tr>
<tr>
<td>SRCH_SALESTERRITORIES</td>
<td>Sales Territory Search</td>
<td>/app/common/search/search.nl?searchtype=SalesTerritory</td>
</tr>
<tr>
<td>SRCH_SHOPPINGCART</td>
<td>Search Shopping Cart</td>
<td>/app/common/search/search.nl?searchtype=ShoppingCart</td>
</tr>
<tr>
<td>SRCH_SOLUTION</td>
<td>Search Solution</td>
<td>/app/common/search/search.nl?searchtype=Solution</td>
</tr>
<tr>
<td>SRCH_TASK</td>
<td>Search Tasks</td>
<td>/app/common/search/search.nl?searchtype=Task</td>
</tr>
<tr>
<td>SRCH_TIME</td>
<td>Search Track Time</td>
<td>/app/common/search/search.nl?searchtype=Time</td>
</tr>
<tr>
<td>SRCH_TOPIC</td>
<td>Search Topic</td>
<td>/app/common/search/search.nl?searchtype=Topic</td>
</tr>
<tr>
<td>SRCH_TRAN_OPPRTNTY</td>
<td>Search Opportunities</td>
<td>/app/common/search/search.nl?searchtype=Opptnty</td>
</tr>
<tr>
<td>SRCH_TRANSACTION</td>
<td>Search Transactions</td>
<td>/app/common/search/search.nl?searchtype=Transaction</td>
</tr>
<tr>
<td>SRCH_VENDOR</td>
<td>Search Vendor</td>
<td>/app/common/search/search.nl?searchtype=Vendor</td>
</tr>
<tr>
<td>SUPT_CENTER_ROLE</td>
<td>NetSuite Support Center</td>
<td>/app/login/dashboard.nl?id=</td>
</tr>
<tr>
<td>TRAN_ADDCONTENT</td>
<td>Add Content</td>
<td>/app/center/setup/addcontent.nl</td>
</tr>
<tr>
<td>TRAN_ADDSHORTCUT</td>
<td>Short Cuts</td>
<td>/core/pages/addShortcut.nl</td>
</tr>
<tr>
<td>TRAN_APPROVAL_EXPIREPT</td>
<td>Approve Expense Reports</td>
<td>/app/accounting/transactions/approval.nl?type=exprept&amp;label=Expense%20Report</td>
</tr>
<tr>
<td>TRAN_APPROVAL_PURCHORD</td>
<td>Approve Purchase Requests</td>
<td>/app/accounting/transactions/approval.nl?type=purchord&amp;label=Purchase%20Request</td>
</tr>
<tr>
<td>TRAN_APPROVALS</td>
<td>Approvals</td>
<td>/app/accounting/transactions/approvals.nl</td>
</tr>
<tr>
<td>TRAN_APPROVECOMMISSN</td>
<td>Approve Commissions</td>
<td>/app/accounting/transactions/approvecommissn.nl</td>
</tr>
<tr>
<td>TRAN_AUDIT</td>
<td>View Audit Trail</td>
<td>/app/accounting/transactions/audit.nl</td>
</tr>
<tr>
<td>TRAN_BANKRECON</td>
<td>Find Matching Transactions</td>
<td>/app/accounting/transactions/bankrecon.nl</td>
</tr>
<tr>
<td>TRAN_BANKVIEW</td>
<td>Online Bank Statement</td>
<td>/app/accounting/transactions/bankview.nl</td>
</tr>
<tr>
<td>TRAN_BAS</td>
<td>Business Activity Statement</td>
<td>/app/accounting/reports/intl/bas.nl</td>
</tr>
<tr>
<td>TRAN_BILLINGS</td>
<td>Invoice Schedule</td>
<td>/app/accounting/transactions/billings.nl</td>
</tr>
<tr>
<td>Task ID</td>
<td>Page Label in NetSuite</td>
<td>URL</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>TRAN_BILLPAY_LOG</td>
<td>Job Status</td>
<td><code>/app/external/xml/upload/uploadlog.nl?displayType=BILLPAY</code></td>
</tr>
<tr>
<td>TRAN_BUDGET</td>
<td>Set Up Budgets</td>
<td><code>/app/accounting/transactions/budgets.nl</code></td>
</tr>
<tr>
<td>TRAN_BULKBILL_LOG</td>
<td>Job Status</td>
<td><code>/app/external/xml/upload/uploadlog.nl?displayType=BULKBILL</code></td>
</tr>
<tr>
<td>TRAN_BULKFULFILL_LOG</td>
<td>Job Status</td>
<td><code>/app/external/xml/upload/uploadlog.nl?displayType=BULKFULFILL</code></td>
</tr>
<tr>
<td>TRANCALENDARPREference</td>
<td>Calendar Preferences</td>
<td><code>/app/cm/calendar/calendarpreference.nl</code></td>
</tr>
<tr>
<td>TRAN_CAMPAIGNSETUP</td>
<td>Set Up Marketing</td>
<td><code>/app/setup/campaignsetup.nl</code></td>
</tr>
<tr>
<td>TRAN_CONFIRM</td>
<td>Transaction Confirmation</td>
<td><code>/core/pages/confirm.nl</code></td>
</tr>
<tr>
<td>TRAN_COPY_BUDGET</td>
<td>Copy Budgets</td>
<td><code>/app/accounting/transactions/copybudget.nl</code></td>
</tr>
<tr>
<td>TRAN_DOMAINS</td>
<td>Set Up Domains</td>
<td><code>/app/setup/domains.nl</code></td>
</tr>
<tr>
<td>TRAN_EMAILPWD</td>
<td>Change E-mail Password</td>
<td><code>/app/center/emailpwd.nl</code></td>
</tr>
<tr>
<td>TRAN_EMPLOYEESFA</td>
<td>Assign Reps</td>
<td><code>/app/common/entity/employeestfa.nl</code></td>
</tr>
<tr>
<td>TRAN_FINCHRG</td>
<td>Assess Finance Charges</td>
<td><code>/app/accounting/transactions/finchrg.nl</code></td>
</tr>
<tr>
<td>TRAN_FORECAST</td>
<td>Edit Sales Rep Forecast</td>
<td><code>/app/cm/sales/forecast.nl</code></td>
</tr>
<tr>
<td>TRANGENERATEFISCALPERIODS</td>
<td>Generate Fiscal Periods</td>
<td><code>/app/setup/period/generatefiscalperiods.nl</code></td>
</tr>
<tr>
<td>TRANGENERATETAXPERIODS</td>
<td>Generate Tax Periods</td>
<td><code>/app/setup/period/generatetaxperiods.nl</code></td>
</tr>
<tr>
<td>TRAN_GSTREFUND</td>
<td>Process GST Refund</td>
<td><code>/app/accounting/transactions/gstrefund.nl</code></td>
</tr>
<tr>
<td>TRAN_IMPACT</td>
<td>GL Impact</td>
<td><code>/app/accounting/transactions/impact.nl</code></td>
</tr>
<tr>
<td>TRAN_INVOICECUSTOMERS</td>
<td>Invoice Billable Customers</td>
<td><code>/app/accounting/transactions/invoicecustomers.nl</code></td>
</tr>
<tr>
<td>TRAN_ITEMSHIPPACK</td>
<td>Mark Orders Packed</td>
<td><code>/app/accounting/transactions/itemshipmanager.nl?type=pack</code></td>
</tr>
<tr>
<td>TRAN_ITEMSHIPSHP</td>
<td>Mark Orders Shipped</td>
<td><code>/app/accounting/transactions/itemshipmanager.nl?type=ship</code></td>
</tr>
<tr>
<td>TRAN_JOURNALAPPROVAL</td>
<td>Approve Journal Entries</td>
<td><code>/app/accounting/transactions/journalapproval.nl</code></td>
</tr>
<tr>
<td>TRAN_LOGINAUDIT</td>
<td>Login Audit</td>
<td><code>/app/setup/loginAudit.nl</code></td>
</tr>
<tr>
<td>TRAN_MANAGEPAYROLL</td>
<td>Manage Payroll</td>
<td><code>/app/payroll/managepayroll.nl</code></td>
</tr>
<tr>
<td>TRAN_MGRFORECAST</td>
<td>Edit Sales Manager Forecast</td>
<td><code>/app/cm/sales/mgrforecast.nl</code></td>
</tr>
<tr>
<td>TRAN_NLVENDOR</td>
<td>Vendor Center</td>
<td><code>/app/center/nlvendor.nl.nl</code></td>
</tr>
<tr>
<td>TRAN_OPENBAL</td>
<td>Enter Opening Balances</td>
<td><code>/app/accounting/transactions/openbal.nl</code></td>
</tr>
<tr>
<td>TRAN_ORDERITEMS</td>
<td>Order Items</td>
<td><code>/app/accounting/transactions/orderitems.nl</code></td>
</tr>
<tr>
<td>TRAN_PAYMENTS</td>
<td>Payments</td>
<td><code>/app/accounting/transactions/payments.nl</code></td>
</tr>
<tr>
<td>TRAN_PAYROLLRUN</td>
<td>Process Payroll</td>
<td><code>/app/accounting/transactions/payroll/payrollrun.nl</code></td>
</tr>
<tr>
<td>TRAN_PDF_F940</td>
<td>Annual Federal Unemployment (940)</td>
<td><code>/app/accounting/transactions/payroll/pdf/pdfframe.nl?type=f940</code></td>
</tr>
<tr>
<td>Task ID</td>
<td>Page Label in NetSuite</td>
<td>URL</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>TRAN_PDF_F941</td>
<td>Quarterly Federal Tax Return (941)</td>
<td>/app/accounting/transactions/payroll/pdf/pdfframe.nl?type=f941</td>
</tr>
<tr>
<td>TRAN_PREVIEWW2</td>
<td>Form W-2 Preview</td>
<td>/app/accounting/reports/w2preview.nl</td>
</tr>
<tr>
<td>TRAN_PRINT</td>
<td>Print Checks and Forms</td>
<td>/app/accounting/print/print.nl</td>
</tr>
<tr>
<td>TRAN_PRINT_CASHSALE</td>
<td>Print Receipts</td>
<td>/app/accounting/print/printframe.nl?trantype=cashsale&amp;printtype=transaction&amp;method=print</td>
</tr>
<tr>
<td>TRAN_PRINT_CHECK</td>
<td>Print Checks</td>
<td>/app/accounting/print/printframe.nl?trantype=check&amp;printtype=transaction&amp;method=print</td>
</tr>
<tr>
<td>TRAN_PRINT_COMMERCIALINVOICE</td>
<td>Print Commercial Invoice</td>
<td>/app/accounting/print/commericalinvoice.nl</td>
</tr>
<tr>
<td>TRAN_PRINT_CUSTCRED</td>
<td>Print Credits Memos</td>
<td>/app/accounting/print/printframe.nl?trantype=custcred&amp;printtype=transaction&amp;method=print</td>
</tr>
<tr>
<td>TRAN_PRINT_CUSTINVC</td>
<td>Print Invoices</td>
<td>/app/accounting/print/printframe.nl?trantype=custinvc&amp;printtype=transaction&amp;method=print</td>
</tr>
<tr>
<td>TRAN_PRINT_ESTIMATE</td>
<td>Print Estimates</td>
<td>/app/accounting/print/printframe.nl?trantype=estimate&amp;printtype=transaction&amp;method=print</td>
</tr>
<tr>
<td>TRAN_PRINT_INTEGRATEDSHIPPINGLABEL</td>
<td>Print Integrated Shipping Labels</td>
<td>/app/accounting/print/printlabels.nl?printtype=integratedshippinglabel&amp;method=print&amp;title=Integrated Shipping Labels</td>
</tr>
<tr>
<td>TRAN_PRINT_PACKINGSLIP</td>
<td>Print Packing Slips</td>
<td>/app/accounting/print/printframe.nl?trantype=&amp;printtype=packingslip&amp;method=print</td>
</tr>
<tr>
<td>TRAN_PRINT_PAYCHECK</td>
<td>Print Checks</td>
<td>/app/accounting/print/printframe.nl?trantype=paycheck&amp;printtype=transaction&amp;method=print</td>
</tr>
<tr>
<td>TRAN_PRINT_PICKINGTICKET</td>
<td>Print Picking Tickets</td>
<td>/app/accounting/print/printframe.nl?trantype=salesord&amp;printtype=pickingticket&amp;method=print</td>
</tr>
<tr>
<td>TRAN_PRINT_PURCHORD</td>
<td>Print Purchase Orders</td>
<td>/app/accounting/print/printframe.nl?trantype=purchord&amp;printtype=transaction&amp;method=print</td>
</tr>
<tr>
<td>TRAN_PRINT_RTNAUTH</td>
<td>Return Authorizations</td>
<td>/app/accounting/print/printform.nl?printtype=transaction&amp;trantype=rtnauth&amp;method=print&amp;title=Return%20Authorizations</td>
</tr>
<tr>
<td>TRAN_PRINT_SALESORD</td>
<td>Print Sales Orders</td>
<td>/app/accounting/print/printframe.nl?trantype=salesord&amp;printtype=transaction&amp;method=print</td>
</tr>
<tr>
<td>TRAN_PRINT_SHIPPINGLABEL</td>
<td>Print Shipping Labels</td>
<td>/app/accounting/print/printframe.nl?trantype=&amp;printtype=shippinglabel&amp;method=print</td>
</tr>
<tr>
<td>Task ID</td>
<td>Page Label in NetSuite</td>
<td>URL</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>TRAN_PRINT_STATEMENT</td>
<td>Generate Statements</td>
<td>/app/accounting/print/printframe.nl?trantype=&amp;printtype=statement&amp;method=print</td>
</tr>
<tr>
<td>TRAN_PRINT1099</td>
<td>Form 1099-MISC</td>
<td>/app/accounting/reports/NLPrint1099s.nl?mode=frame</td>
</tr>
<tr>
<td>TRAN_PRINTBARCODES</td>
<td>Generate Barcodes</td>
<td>/app/accounting/print/printbarcodes.nl?printtype=null&amp;method=print</td>
</tr>
<tr>
<td>TRAN_PRINTSTATEMENT</td>
<td>Individual Statement</td>
<td>/app/accounting/print/printstatement.nl</td>
</tr>
<tr>
<td>TRAN_PRINTW2</td>
<td>Form W-2</td>
<td>/app/accounting/reports/NLPrintW2s.nl?mode=frame</td>
</tr>
<tr>
<td>TRAN_PRINTW3</td>
<td>Form W-3</td>
<td>/app/accounting/reports/NLPrintW3s.nl?mode=frame</td>
</tr>
<tr>
<td>TRAN_PROCESSCOMMISSN</td>
<td>Authorize Commissions</td>
<td>/app/accounting/transactions/processcommissn.nl</td>
</tr>
<tr>
<td>TRAN_PROCESSORDER</td>
<td>Process Individual Order</td>
<td>/app/accounting/transactions/processorder.nl</td>
</tr>
<tr>
<td>TRAN_PURCHORDPROC</td>
<td>Bill Purchase Orders</td>
<td>/app/accounting/transactions/purchordermanager.nl?type=proc</td>
</tr>
<tr>
<td>TRAN_PURCHORDRECEIVE</td>
<td>Receive Purchase Order</td>
<td>/app/accounting/transactions/purchordermanager.nl?type=receive</td>
</tr>
<tr>
<td>TRAN_QUOTA</td>
<td>Establish Quotas</td>
<td>/app/crm/sales/quota.nl</td>
</tr>
<tr>
<td>TRAN_REALLOCITEMS</td>
<td>Reallocate Items</td>
<td>/app/accounting/transactions/reallocitems.nl</td>
</tr>
<tr>
<td>TRAN_RECONCILE</td>
<td>Reconcile Bank Statement</td>
<td>/app/accounting/transactions/reconcile.nl</td>
</tr>
<tr>
<td>TRAN_RECONCILE_CC</td>
<td>Reconcile Credit Card Statement</td>
<td>/app/accounting/transactions/reconcile.nl?page_type=cc</td>
</tr>
<tr>
<td>TRAN_REIMBURSEMENTS</td>
<td>Reimbursements</td>
<td>/app/accounting/transactions/reimbursements.nl</td>
</tr>
<tr>
<td>TRAN_REMINDERS</td>
<td>Setup Reminders</td>
<td>/app/center/setup/reminders.nl</td>
</tr>
<tr>
<td>TRAN_REVRECCREATEJE</td>
<td>Revenue Recognition Schedules</td>
<td>/app/accounting/transactions/revreccreateje.nl</td>
</tr>
<tr>
<td>TRAN_RTNAUTHAPPRV</td>
<td>Approve Return Authorizations</td>
<td>/app/accounting/transactions/returnauthmanager.nl?type=apprv</td>
</tr>
<tr>
<td>TRAN_RTNAUTHCREDIT</td>
<td>Refund Returns</td>
<td>/app/accounting/transactions/returnauthmanager.nl?type=credit</td>
</tr>
<tr>
<td>TRAN_RTNAUTHRECEIVE</td>
<td>Receive Returned Order</td>
<td>/app/accounting/transactions/returnauthmanager.nl?type=receive</td>
</tr>
<tr>
<td>TRAN_SALESORDAPPRV</td>
<td>Approve Sales Orders</td>
<td>/app/accounting/transactions/salesordermanager.nl?type=apprv</td>
</tr>
<tr>
<td>TRAN_SALESORDFULFILL</td>
<td>Fulfill Sales Orders</td>
<td>/app/accounting/transactions/salesordermanager.nl?type=fulfill</td>
</tr>
<tr>
<td>TRAN_SALESORDPROC</td>
<td>Bill Sales Orders</td>
<td>/app/accounting/transactions/salesordermanager.nl?type=proc</td>
</tr>
<tr>
<td>TRAN_SEARCH</td>
<td>Search</td>
<td>/app/common/search/search.nl</td>
</tr>
<tr>
<td>TRAN_SHORTCUTS</td>
<td>Add Shortcuts</td>
<td>/app/center/shortcuts.nl</td>
</tr>
<tr>
<td>TRAN_SNAPSHOTCOMPOSER</td>
<td>Custom Snapshot Report</td>
<td>/app/reporting/snapshotcomposer.nl</td>
</tr>
<tr>
<td>TRAN_SNAPSHOTS</td>
<td>Setup Snapshots</td>
<td>/app/center/setup/snapshots.nl</td>
</tr>
<tr>
<td>Task ID</td>
<td>Page Label in NetSuite</td>
<td>URL</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>TRAN_TAXPERIODS</td>
<td>Generate Tax Reporting Periods</td>
<td>/app/setup/period/generatetaxperiods.nl</td>
</tr>
<tr>
<td>TRAN_TIMEAPPROVAL</td>
<td>Approve Time</td>
<td>/app/accounting/transactions/timeapproval.nl</td>
</tr>
<tr>
<td>TRAN_TIMEBILL</td>
<td>Track Time</td>
<td>/app/accounting/transactions/timebill.nl</td>
</tr>
<tr>
<td>TRAN_TIMEBILL_WEEKLY</td>
<td>Weekly Time Sheet</td>
<td>/app/accounting/transactions/timebill.nl?weekly=T</td>
</tr>
<tr>
<td>TRAN_TIMECALC</td>
<td>Calculate Time</td>
<td>/core/pages/timecalc.nl</td>
</tr>
<tr>
<td>TRAN_TIMER</td>
<td>Timer</td>
<td>/core/pages/timer.nl</td>
</tr>
<tr>
<td>TRAN_USERPREFS</td>
<td>Set Preferences</td>
<td>/app/center/userprefs.nl</td>
</tr>
<tr>
<td>TRAN_VAT100</td>
<td>VAT 100</td>
<td>/app/accounting/reports/intl/vat100.nl</td>
</tr>
<tr>
<td>TRAN_VENDBILLPURCHORD</td>
<td>New Purchase Order</td>
<td>/app/accounting/transactions/vendbillpurchord.nl</td>
</tr>
<tr>
<td>TRAN_VENDPYMTS</td>
<td>Pay Bills</td>
<td>/app/accounting/transactions/vendpymts.nl</td>
</tr>
</tbody>
</table>
FAQs

- **Where is the NetSuite WSDL located?**
  
  https://webservices.netsuite.com/wsdl/v2_5_0/netsuite.wsdl
  
  (where v2_5_0 is the version of the wsdl)

- **What programming languages and platforms are supported in this release?**
  
  This release should work with any platform or tool that supports SOAP 1.1. NetSuite has verified that the Java Apache Axis and Microsoft .NET platforms work correctly, and sample programs for these environments are available in the Resources section of the Web Services Beta portal.

- **What is an internal ID (or nsKey)? What is a RecordRef? How do I get the values for internal IDs?**
  
  An internal ID is a unique key used to reference a record within NetSuite. In Web Services, the RecordRef type is used to abstract the internal ID. For example, a RecordRef is required in order to uniquely identify the records for update, delete, or get operations. Also, a typical record exposed through Web services will have references to other NetSuite records of different types through the RecordRef type.
  
  The internal ID for a record can be obtained as follows:
  
  - If a record type has been exposed through Web services, a search operation will return the internal ID.
  - Alternatively, you can also obtain the internal ID for that record in the UI on the list page for that record type. In order to see the internal ID, you will first need to set the "Web Services: Show internal IDs" checkbox in Home > Set Preferences > General.

- **What are the maximum number of records allowed for write operations such as addList, updateList, or deleteList?**
  
  This varies depending on whether records are submitted during peak or off-peak hours. Contact your account representative for more information.

- **Why can't I maintain a valid session using Apache Axis?**
  
  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. A patched version of the axis.jar can be downloaded from NetSuite Web Services Developer’s page to fix this problem. Please note that the patch you download depends on the version of Axis that you are using. Replace the existing axis.jar file in the lib directory of your project with the NetSuite version.
**Why can’t I maintain a valid session using Microsoft .NET?**

The NetSuite Web service implementation requires the client application to support multiple cookies. Please see the .NET sample application for an example of how to enable support for multiple cookies.

**I have populated some fields in a record when adding or updating, but why are these fields not being populated in the SOAP document?**

This is a .NET anomaly. For every field that’s not of type string, the generated .NET object contains an additional attribute whose name is the field name concatenated with the word "Specified". For example, there is a Boolean field name called inactive in the Customer record type. When .NET generates this class, it adds an additional Boolean field called inactiveSpecified. This field needs to be set to true for the inactive field to be included in the SOAP document.

**Can multiple users logged into the same account run Web services simultaneously?**

Yes. Web services are associated to the user, NOT the account. Therefore, multiple users can access the same account via Web services. Also, a single user can have a concurrent browser and Web services session, and additional Web services sessions for a given user can exist for Web services Partner applications.

**Is it possible to log directly into an account via Web services without having to login as a specific NetSuite user?**

In other words is there such a thing as a "web services only user" with zero seat charge, the cost of service being tied to usage or calls rather than seat.

No. Currently, a Web service must be tied to an account and a specific user within that account. A *Web services only user*, with the cost of the service associated to usage rather than a specific seat, is not possible.

**Is it possible to run Web services and XML independently and simultaneously within an account without mutually exclusive restrictions?**

There are currently no restrictions. However, to ensure adequate load distribution, the use of all NetSuite integration technologies will be analyzed and restrictions implemented as appropriate.

**How do I query for all records of a given type?**

Use the search operation with a search record that has no fields populated.

**I'm getting an INVALID_CREDENTIALS fault even though all of my login information is correct. What more do I need to do?**

To login to NetSuite Web services, you must add a cookie to the http header of your request. The cookie should be ACCOUNT=<your account>, and must match the account submitted in the login SOAP request.

In Java this code should be executed before you log in.
**Note:** You must use a Vector object because multiple cookies are used. This is different from the standard Axis code which only accepts one cookie and takes a String type.

**Sample Code: Java**

```java
Vector cookies = new Vector();
String acct = "ACCT999999"; // This string is unique for your account
cookies.add("ACCOUNT=\"+acct);
((javax.xml.rpc.Stub) port)._setProperty(HTTPConstants.HEADER_COOKIE, cookies);
```

**Sample Code: .NET**

```csharp
Uri myUri = new Uri("https://webservices.netsuite.com");
_service.CookieContainer = new CookieContainer();
// accountId is the string value of the NetSuite account being logged into
_service.CookieContainer.SetCookies(myUri, "ACCOUNT=\" + accountId);
```

**Can you recommend a utility for capturing and analyzing SOAP request and response messages associated with an operation call?**

Mindreef's SOAPScope is an excellent utility for this. NetSuite has partnered with Mindreef to offer a special NetSuite branded version of SOAPscope. This NetSuite branded version of SOAPscope is now available from Mindreef as a free evaluation download or can be purchased as a subscription for the low price of $99 per year.

Download SOAPscope

For detailed instructions on how to install and get started with SOAPScope, see “Using SOAPScope” on page 19.

**I get a soap fault and a message which states “The Operation has timed-out” when using my .Net client and I get no soap response back. What could be causing this?**

If you have NOT explicitly set the timeout on your .Net application, this may be related to the preset timeout of 100 seconds on the SoapHttpClientProtocol class. You must change this timeout to be equal to or greater than the Netsuite timeout (15 minutes). Otherwise, your client will abort processing prematurely.

**Note:** If you receive the same fault, which includes an HTTP error code (e.g. 50x), this is likely a NetSuite related timeout.

**Sample Code: .Net**

```csharp
public NSClient()
{
    _isAuthenticated = false;
    _pageSize = 40;
    // Instantiate the NetSuite web service
    _service = new NetSuiteService();
    // Set client timeout to 100 hours
    _service.timeout = 100*60*60*1000;
```

**Sample Code: Java**

```java
// Locate the NetSuite web service.
NetSuiteServiceLocator service = new NetSuiteServiceLocator();
// Enable client cookie management. This is required.
service.setMaintainSession(true);
// Get the service port
_port = service.getNetSuitePort();
```
I am getting an unexpected server side protocol timeout error (502) and no soap response is sent back. What could be causing this?

Check the timeout settings on any intermediaries that you are using — such as proxy servers or port sniffer SOAPScope.

To edit the timeout setting for SOAPScope, you need to edit the httpd.conf file to the desired settings and then restart the SOAPScope service. The httpd.conf file is located in the conf subdirectory of the SOAPScope data directory.

Note: There is a link to the SOAPScope data directory in the Start menu.

In the following code the default Timeout value of 300 has been changed to 901.

```plaintext
PidFile nul:
Timeout 901
KeepAlive On
MaxKeepAliveRequests 0
KeepAliveTimeout 15
```

Why are attachments uploaded with Messages getting corrupted?

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

Sample Code:

- Correct

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

- Incorrect

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

How can I determine internal ID values for individual items in a custom list?

Generally, a given list displays the internal IDs as one of the columns. For example, List > Relationships > Customers displays the internal IDs (nsKeys) as the second column (when the preference is turned on -- see “Displaying Internal IDs” on page 25).

When referencing items in a custom list, you must first reference the SelectCustomFieldRef by its internal ID, which is displayed in the respective custom field list (for example, for a custom customer field it would be displayed at Setup > Customization > Entity Fields).
Next, reference the internal ID to the Custom List or Record. Internal IDs to custom lists are NOT displayed in the UI. You must refer to the HTML source for the internal ID. When in the HTML source, search for the name of the value and note the integer which follows. For example, if referring to the Items of Interest custom list (provided in most test accounts) and need to select Monitors as one of the values, view the HTML source in the browser and find that string:


Note: The display order is in the order the values appear, but the internal IDs may be different. Monitors in this case has a internal ID of 2.

If you are referring to a Record, you simply need to locate the internal ID displayed in the column of the appropriate list.

When working with custom records, why do I need a separate CustomRecordRef class, when even for creating a customRecord I have to instantiate the RecordRef type?

In general RecordRef is for get() and Record is for add(). We sometimes use RecordRef to store name/value pairs.

RecordRef and CustomRecordRef both descend from BaseRef. They are most prominently used in get(), and that is why they have a type attribute. CustomRecordRef has a typeId to indicate which kind of CustomRecord it is. Therefore, you can NOT get a CustomRecord by setting the RecordRef:type to customRecord on add — you must use CustomRecordRef to specify the kind of CustomRecord.

CustomRecord descends from Record, just like Customer or Opportunity. This is what you submit to add() and it is what is returned through search(). CustomRecord uses a RecordRef() to store it's type information.

Java Example

To add a CustomRecord

```java
CustomRecord myCR = new CustomRecord();
RecordRef rt = new RecordRef();
rt.setName("Authors");
rt.setId("3"); // This indicates a typeId of 3 and corresponds to the Authors CustomRecord type
myCR.setRecType(rt);
myCR.setName("Ernest Hemmingway"); // This is what will show up in List->CustomRecords->Authors under Name
myCR.setCustomFieldList(myCFL); // An already filled out CustomField List
WriteResponse wr = port.add(myCR);
```

Note that setTypeId is not submitted since this is the internal ID returned by the response, just like for normal records.

To get a CustomRecord:

```java
CustomRecordRef myCRR = new CustomRecordRef();
myCRR.setKey("1200"); // The key we got back
myCRR.setTypeId("3"); // For Authors
myCRR.setType(RecordType.customRecord);
ReadResponse rr = port.get(myCRR);
```

.Net Example
To add a CustomRecord

```java
CustomRecord myCR = new CustomRecord();
RecordRef rt = new RecordRef();
rt.name = "Authors";
// This indicates a typeId of 3 and corresponds to the Authors CustomRecord type
rt.internalId = "3";
customRecord.recType = rt;
// This is what will show up in List->CustomRecords->Authors under Name
customRecord.name = "Ernest Hemingway";
customRecord.customFieldList = myCFL; // An already filled out CustomField List
WriteResponse wr = port.add(customRecord);
```

To get a CustomRecord:

```java
CustomRecordRef myCRR = new CustomRecordRef();
myCRR.internalId = "1200"; // The key we got back
myCRR.typeId = "3"; // For Authors
ReadResponse rr = _service.get(myCRR);
```

Why do I not get data for all elements on a search for a given record type?

When performing searches on records, by default only the body fields are returned.

For example, if a Get or GetList operation against a customer record is submitted, the addressbooklist element contains data but if a Search operation against a customer record is submitted, the addressbooklist element is un-defined.

To return field values for all lists on a record, you must set the bodyFieldsOnly element of the searchPreferences type to False. It is recommended that the default settings are used wherever list values are not necessary since this significantly improves performance.

```xml
<complexType name="SearchPreferences">
  <sequence>
    <element name="bodyFieldsOnly" type="xsd:boolean" default="true" />
  </sequence>
</complexType>
```

Note: You can perform quick and full searches as the same time. For example, to show a list of email folders, but only query for messages in the default folder, do a quick search to get folder names/keys, and a full search on all messages.

Why are some classes from the NetSuite WSDL not being generated?

If you are using Java, in order to generate unreferenced types you should set the all parameter in your axis-wsdl2java ant task to true. For example:

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